

Resume

William M. McKeeman Senior Engineering Fellow, Computer Science, Tufts University
Adjunct Faculty, Computer Science Department, Dartmouth College
mckeeman@cs.dartmouth.edu
(prepared January 26, 2011)

EDUCATION

Ph.D. Computer Science, Stanford University, 1966
M.A. Mathematics, George Washington University, 1961
A.B. Mathematics, University of California, Berkeley, 1956

EMPLOYMENT

1999-2010 Language of Technical Computing, MathWorks, Fellow
1988-1999 Core Technology Group, Digital/Compaq, Senior Consulting Engineer
1987-1988 Aiken Computation Laboratory, Harvard University, Visiting Research Professor
1981-1987 Wang Institute of Graduate Studies, Professor, Chair of the Faculty 1981-84
1979-81 Xerox Palo Alto Research Center, Visiting Scientist
1968-79 University of California, Santa Cruz, Associate Professor, Professor, Chair of Information Sciences
1973-76
1971-72 Computer Systems Research Group, University of Toronto, Visiting Associate Professor
1966-68 Computer Science Department, Stanford University, Assistant Professor
1959-61 Physics Department, U.S. Naval Academy, Instructor
1956-61 U.S. Navy, Lt., pilot

PROFESSIONAL ACTIVITIES

2011-present Senior Engineering Fellow, Computer Science, Tufts University
1999-present Adjunct Professor of Computer Science, Dartmouth University
1984-1987 Editorial Board, IEEE Software
1979-present IFIP Working Group 2.3 on Programming Methodology (emeritus)
1966-1988 Consulting with Avnet, Burroughs, MathWorks, TRW, Wang, IBM, Zilog and others.
1961-1996 ΣΞ

PUBLIC PRESENTATIONS

Writing Compilers in MATLAB, Chitkara University, June 2010
How do JITs Work?, Worcester Polytechnic, April 22, 2010
Can a tiny compiler compiler grow into something useful?, Stanford EE, March 2009
A New Look at LR(k), Worcester Polytechnic, February 20, 2004
Solving Space Fillings, Colby College, April 12, 1999
A Modern Software Development Process, 25th Anniversary of the German ACM, November 12, 1993

Technical Means of Software Protection, Wang Institute Distinguished Lecturer series, October 1, 1984

Keynote Speaker, The Computer in Society, Stetson University, April 15, 1982

Advanced Programming Methodology, *API*, Lisbon, Portugal, October 1981

Advanced Compiler Construction, Tsinghua University and Peking University, Peking, PRC, April 27, 1979

Compiler Construction: an advanced course, The Mathematical Institute of The Technical University of Munich, March 1974.

Compiler Organization, Design and Implementation Techniques, 10 lectures, International Seminar in Computer Science, Hebrew University, Jerusalem, July 1970

ACM National Lecturer 1969

PUBLICATIONS.

Bill McKeeman. The computation of pi by archimedes.

<http://www.mathworks.com/matlabcentral/fileexchange/29504>, November 2010.

Bill McKeeman. Polytopes. <http://www.mathworks.com/matlabcentral/fileexchange/18523>, January 2008.

Bill McKeeman. Matlab performance measurement.

<http://www.mathworks.com/matlabcentral/fileexchange/18510>, January 2008.

William M. McKeeman and Loren Shure. An adventure of sorts. *MATLAB News and Notes*, December 2004.

W. M. McKeeman. Solving space fillings.

<http://www.cs.dartmouth.edu/~mckeeman/references/soma/index.html>, April 1999.

William M. McKeeman. Differential testing for software. *Digital Technical Journal*, 10(1):100–107, December 1998.

Bill McKeeman. A lexer for Java in C++. *C/C++ User's Journal*, 16(11):23–30, January 1998.

William M. McKeeman. Compiler construction. In Anthony Ralston and Edwin D. Reilly, editors, *Encyclopedia of Computer Science and Engineering*, pages 207–210. Van Nostrand, 1993. Third edition.

W. M. McKeeman. Avoiding arithmetic exceptions in constant expression evaluation. *Journal of C Language Translation*, 4(4):291–295, June 1993.

W. M. McKeeman. A brief introduction to C. In *Proceedings of the Second ACM SIGPLAN History of Programming Languages Conference*, pages 347–348. ACM, 1993. In support of the Ritchie History of C paper.

W. M. McKeeman and Shota Aki. Reusable incremental C parsing. *Journal of C Language Translation*, 4(3):191–202, March 1993.

W. M. McKeeman and Shota Aki. Reusable incremental C preprocessing, part II. *Journal of C Language Translation*, 4(2):117–125, December 1992.

W. M. McKeeman and Shota Aki. Reusable incremental C preprocessing, part I. *Journal of C Language Translation*, 4(1):53–69, September 1992.

W. M. McKeeman. Cb: a low-level subset of C. *Journal of C Language Translation*, 3(3):214–226, December 1991.

W. M. McKeeman and Shota Aki. Reusable incremental scanning. *Journal of C Language Translation*, 3(2):101–120, September 1991.

W. M. McKeeman. Resolving typedefs in a multipass C compiler. *Journal of C Language Translation*, 2(4):259–266, March 1991.

W. M. McKeeman, Shota Aki, and Scot Aurenz. Parser-independent compilers. *Journal of C Language Translation*, 2(3):177–184, December 1990.

- W. M. McKeeman. Graduation talk at Wang Institute. *IEEE Computer*, pages 78–80, May 1989.
- W. M. McKeeman. Experience with a software engineering project course. *IEEE Transactions on Software Engineering*, SE-13(11), November 1987. Special Issue in Software Engineering Education. See also *Software Engineering Education: The Educational Needs of the Software Community*, Norman Gibbs and Richard Fairley (editors), Springer-Verlag, New York, 1987, pages 234-262 and Technical Report TR-86-01, Wang Institute of Graduate Studies.
- W. M. McKeeman, Nancy Martin, Tim Gill, Jack Gehling, and Susan Trager. Expression side effects in C. *C Journal*, pages 48–52, April 1986. See also Technical Report TR-85-04, Wang Institute of Graduate Studies for more complete treatment.
- W. M. McKeeman. Communicating programmers – the glue of software engineering. *Notizen zu Interaktiven Systemen*, 11:3–12, Juli 1983.
- Keith Clark, W. M. McKeeman, and Sharon Sickel. Logic programming specification of numerical integration. In *Logic Programming*, volume 16 of *APIC Studies in Data Processing*, pages 123–140. Academic Press, 1982. See also *Logic programming applied to numerical integration*, same authors, Technical Report 78-0-004, University of California at Santa Cruz August 1978, and review in *SigSoft* July 1983, page 66.
- W. M. McKeeman. Addressless instructions. In Anthony Ralston, editor, *Encyclopedia of Computer Science*, page 21. Van Nostrand, 1980. Second edition.
- W. M. McKeeman. The Burroughs B5000 series of computers. *ibid* pages 189-191.
- W. M. McKeeman. A formal model for space-filling puzzles. In *Machine Intelligence*, volume 8, pages 86–93. John Wiley and Sons, 1977.
- W. M. McKeeman. A bachelor of arts in computer programming. In *Proceedings of COMPCOM 76*, pages 212–214, San Francisco, February 1976. IEEE Computer Society. See also *Software Engineering Education*, P. Freeman and A. Wasserman (editor), Springer-Verlag, 1976, pages 123-126.
- W. M. McKeeman. Mechanizing bankers’ morality. *Journal of Computer Languages*, 1:73–82, 1975.
- W. M. McKeeman. On preventing programming languages from interfering with programming. *IEEE Transactions on Software Engineering*, 1(1):19–25, March 1975. See also The Eighth Hawaii International Conference on Computer Sciences, January 1975 and Keynote at COMPCON 75, San Francisco, February 26, 1975.
- W. M. McKeeman and F.L. DeRemer. Feedback-free modularization of compilers. In *Conference on Programming Languages*, volume 7 of *Lecture Notes in Computer Science*, pages 78–88. Springer-Verlag, Kiel, Germany, March 1974.
- W. M. McKeeman. Compiler structure. In *Proceedings of the First USA-Japan Computer Conference*, Tokyo, October 1972. AFIPS. See also Technical Report CSRG-23, University of Toronto, January 1973.
- P. S. Abrams and W. M. McKeeman. Computer display of the derived polytopes. *CEGOS*, June 1970.
- W. M. McKeeman, J. J. Horning, E. C. Nelson, and D. B. Wortman. The XPL compiler generator system. In *Proceedings of the Fall Joint Computer Conference*, pages 617–635. AFIPS, Spartan Books, December 1968.
- W. M. McKeeman. Analysis of binary formats for floating point arithmetic. *IEEE Transactions on Electronic Computers*, EC-16:682, October 1967.
- W. M. McKeeman. Language directed computer design. In *Proceedings of the Fall Joint Computer Conference*, pages 413–417. AFIPS, 1967.
- W. M. McKeeman. Peephole optimization. *Comm ACM*, 8:443, July 1965. See also ACM collection: *Compiler Techniques*, B. Pollack, editor Auerbach, pages 225-228, 1972.

BOOKS and CHAPTERS.

- Snook, Bass, Roberts, Nahapetian, Fay, McKeeman, Meyer, Carper, and Lane. *Report on the Language PLZ/SYS*. Springer-Verlag, 1978. Second edition 1983. (First edition listed first five authors only.).

- W. M. McKeeman. Case studies in reliable computing. In *Computer Science and Scientific Computing*, pages 163–189. Academic Press, 1976.
- W. M. McKeeman. Stack computers. In Harold Stone, editor, *Introduction to Computer Architecture*, chapter 7. Science Research Associates, July 1975. Second edition 1980.
- W. M. McKeeman, J. J. Horning, and D. B. Wortman. *A Compiler Generator*. Prentice-Hall, 1970. Main selection for Library of Computer and Information Sciences Book Club, November 1971, also in Russian, 1980.
- W. M. McKeeman. Compiler construction. In *Advanced Course on Compiler Construction*, Lecture Notes in Economics and Mathematical Systems, pages 1–36. Springer-Verlag, October 1974. Second edition 1976.
- W. M. McKeeman. Symbol table access. *ibid.* pages 253-301. See also Technical Report CEP 2,1, University of California at Santa Cruz, April 1970.
- W. M. McKeeman. Programming language design. *ibid.* pages 514-524.

ALGORITHMS and SHORT NOTES.

- Bill McKeeman. Solutions to the soma cube.
<http://www.mathworks.com/matlabcentral/fileexchange/26346>, January 2010.
- Bill McKeeman. Can a tiny compiler-compiler grow into something useful?
<http://www.mathworks.com/matlabcentral/fileexchange/23255>, March 2009.
- Bill McKeeman. World time zones. <http://www.mathworks.com/matlabcentral/fileexchange/22223>, November 2008.
- Bill McKeeman. Summer olympic medals 1896-present.
<http://www.mathworks.com/matlabcentral/fileexchange/21117>, August 2008.
- Bill McKeeman. Wind fern. <http://www.mathworks.com/matlabcentral/fileexchange/19141>, March 2008.
- Bill McKeeman. Tumbling bucky ball. <http://www.mathworks.com/matlabcentral/fileexchange/10329>, March 2006.
- Bill McKeeman. perms. <http://www.mathworks.com/matlabcentral/fileexchange/10237>, March 2006.
- W. M. McKeeman. Alpha magnetic spectrometer animation.
<http://www.cs.dartmouth.edu/~mckeeman/programs/ams/ams.html>, November 1997.
- W. M. McKeeman. Solving the soma cube (java).
<http://www.cs.dartmouth.edu/~mckeeman/programs/soma/somadisplay.html>, September 1996.
- W. M. McKeeman. Summary and analysis of student retrospectives for the Master of Software Engineering degree at Wang Institute. *Wang Institute Software Engineering Review*, 1(2):66–77, 1986.
- W. M. McKeeman. Professional software engineering. *IEEE Software*, page 112, October 1984. In SoftNews section.
- W. M. McKeeman. Writing correct programs. In Programming Pearls section, *Communications of the ACM*, pages 631-632., July 1984.
- Nancy Martin and W. M. McKeeman. Educating the software engineer. In *The Software Explosion: Issues in Software Technology, Measurement and Evaluation through Human Engineering and Management Perspectives*, page 22. American Association for the Advancement of Science, January 1982.
- Frank DeRemer, Tom Pennello, and W. M. McKeeman. Ada syntax chart. *Sigplan Notices*, 8(12):48–59, September 1981.
- W. M. McKeeman. The role of software engineering in the microcomputer revolution: an overview. In *Proceedings of the Fourth International Conference on Software Engineering*, page 340. IEEE, 1979. Extended abstract.
- W. M. McKeeman. A primitive report generator. *Dr. Dobbs*, 4(1):38–39, January 1979. See also Application Note 2, Zilog Software Department, May 1978 and Technical Report 78-5-001, University of California at Santa Cruz, June 1978.

- W. M. McKeeman. Programming language translation techniques. *Dr. Dobbs*, 3(5):4–16, May 1978. See also Technical Report CSRG-17, Computer Systems Research Group, University of Toronto, July 1972, 51 pages.
- W. M. McKeeman. Free field read. *Comm ACM*, 7(8):481, August 1964. Algorithm 239.
- W. M. McKeeman. Algol 60 reference language editor. *Comm ACM*, 8(11), November 1965. Algorithm 268.
- W. M. McKeeman. Adaptive integration and multiple integration. *Comm ACM*, 6(8):443, August 1963. Algorithm 198.
- W. M. McKeeman and Larry Tesler. Nonrecursive adaptive integration. *Comm ACM*, 6(6):315, June 1963. Algorithm 182.
- W. M. McKeeman. Multiple integration. *Comm ACM*, 5(12):604, December 1962. Algorithm 146.
- W. M. McKeeman. Adaptive numerical integration by Simpson’s Rule. *Comm ACM*, 5(12):604, December 1962. See also Certification of Algorithm 145, *Comm ACM* 6(4):167, April 1963.
- W. M. McKeeman. Crout with equilibration and iteration. *Comm ACM*, 5(11):553, November 1962. See also Certification of Algorithm 135, *Comm ACM* 5(11):557, November 1962 and An Accurate Algorithm for the Solution of Simultaneous Linear Algebraic Equations, Technical Report 26, Applied Mathematics and Statistics Laboratories, Stanford, January 18, 1963.

REPORTS.

- W. M. McKeeman, Susan Trager, Joshua Cohen, and Ting Yang. C grammars. Technical Report TR-87-02, Wang Institute of Graduate Studies, Tyngsboro, MA 01879, February 1987. 68 pages.
- Ardis, Bernstein, Claybrook, Fairley, Goodenough, Lomet, McKeeman, Raghavan, Perlman, Velasco, and Weiss. Core courses in the master of software engineering at Wang Institute 1985-86. Technical Report TR-86-11, Wang Institute of Graduate Studies, Tyngsboro, MA 01879, September 1986. W. M. McKeeman, editor.
- W. M. McKeeman. Static analysis for minimal basic. Technical Report TR-83-02, Wang Institute of Graduate Studies, September 1982. 10 pages.
- Dan Ligett, Glen McCluskey, and W. M. McKeeman. Parallel LR parsing. Technical Report TR-82-03, Wang Institute of Graduate Studies, 1982. 113 pages.
- Sharon Sickel and W. M. McKeeman. Hoare’s program FIND revisited. Logic Programming Workshop, 1980. 6 pages.
- Sharon Sickel and W. M. McKeeman. Axiomatic specification of syntax-directed translation. Technical Report 78-8-002, University of California at Santa Cruz, August 1978. 16 pages.
- W. B. Tyler and W. M. McKeeman. Genesis: A language for describing the development of programs. Technical Report 78-2-002, University of California at Santa Cruz, January 1977. 23 pages.
- W. M. McKeeman, M. J. Fay, and T. J. Pennello. Efficient solution of space-filling puzzles. Technical Report 78-2-001, University of California at Santa Cruz, 1978. 9 pages.
- W. M. McKeeman. Stack computers. In B. Shaw, editor, *Proceedings of the Joint IBM University of Newcastle upon Tyne Seminar*, pages 163–171. University of Newcastle upon Tyne, 1978.
- W. M. McKeeman. Application programming systems so friendly that even a salesman can use them. In B. Shaw, editor, *Proceedings of the Joint IBM University of Newcastle upon Tyne Seminar*, pages 157–163. University of Newcastle upon Tyne, 1978.
- Daniel Ross and W. M. McKeeman. Highlights of LO11. Technical Report 77-8-005, University of California at Santa Cruz, August 1977. 18 pages.
- W. M. McKeeman. Respecifying the telegram problem. Technical Report 77-2-001, University of California at Santa Cruz, February 1977. 14 pages.
- W. M. McKeeman. Corporate integration through systems software. *Sigplan Notices*, April 1976. 5 pages.

- W. M. McKeeman. Computer design evaluation using programming language primitives. *Computer Architecture News*, 3(1):7–18, March 1974. See also Technical Report CEP 5,4, University of California at Santa Cruz, December 1973.
- W. M. McKeeman. The weight of information. Technical Report CEP 5,2, University of California at Santa Cruz, November 1973. 11 pages.
- W. M. McKeeman. A simple computer. Technical Report CEP 4,1, University of California at Santa Cruz, September 1972. 32 pages Revised September 1973. See also Proceedings of the Joint IBM-University of Newcastle upon Tyne Seminar, September 1977, pages 171-180.
- W. M. McKeeman. On using paging. Technical Report CEP 2,5, University of California at Santa Cruz, June 1970. 6 pages.
- W. M. McKeeman. Data protection by self-aware computing systems. Technical Report CEP 2,6, University of California at Santa Cruz, June 1970. 7 pages.
- W. M. McKeeman. Computer evolution project definition. Technical Report CEP Report 1,1, University of California at Santa Cruz, November 1968.
- W. M. McKeeman and H. Van Zoeren. An experimental PL/I translator. Technical Report 33-78-01, Computation Center, Stanford University, 1966. 49 pages.
- W. M. McKeeman and Niklaus Wirth. Gogol. Stanford Time-sharing Project Memo No. 24, November 1964. 21 pages.
- W. M. McKeeman. Language comparisons. *Datamation*, August 1964. Letter to the Editor, page 12.
- Bill Rouseau and W. M. McKeeman. Map: A program to generate mappings in the complex plane. Memorandum, Stanford Artificial Intelligence Group, September 1963.

PATENTS.

- W. M. McKeeman and Steve Johnson. Reading encrypted information without ever representing it in cleartext in memory. filed, November 2006.
- John Micco et al. Function call translation 7454746. filed, November 2006.
- W. M. McKeeman et al. Improved program interpreter 6973644 7725883. issued, April 2002.
- W. M. McKeeman. Compiler as runtime assistant. filed, July 1998.
- W. M. McKeeman, August Reinig, and Andy Payne. Method and apparatus for software testing using a differential testing technique to test compilers. U. S. Patent 5,754,860, May 1998.
- W. M. McKeeman and August Reinig. Standalone test and development jig generator. U. S. Patent 5,651,111, July 1997.
- W. M. McKeeman and Shota Aki. Line-skip compiler for source-code development system. U. S. Patent 5,201,050, April 1993. Filed June 30, 1989, application 375,402.
- W. M. McKeeman and Shota Aki. Re-execution of edit-compile-run cycles for changed lines of source code, with storage of associated data in buffers. U. S. Patent 5,313,387, May 1994. Filed June 30, 1989, application 375,401.
- W. M. McKeeman and Shota Aki. Virtual memory management for source-code development system. U. S. Patent 5,301,327, April 1994. Filed June 30, 1989, application 375,399.
- W. M. McKeeman and Shota Aki. Incremental linking in source-code development system. U. S. Patent 5,193,191, March 1993. Filed June 30, 1989, application 375,398.
- W. M. McKeeman and Shota Aki. Incremental compiler for source-code development system. U. S. Patent 5,182,806, January 1993. Filed June 30, 1989, application 375,397. Also Taiwan patent 44813.
- W. M. McKeeman and Shota Aki. Compiler using clean lines table with entries indicating unchanged text lines for incrementally compiling only changed source text lines. U. S. Patent 5,325,531, June 1994. Filed June 30, 1989, application 375,384.
- W. M. McKeeman and Shota Aki. Incremental-scanning compiler for source-code development system. U. S. Patent 5,170,465, December 1992. Filed June 30, 1989, application 375,383.

- W. C. Price C. B. Carlson, W. M. McKeeman. Address manipulation circuitry for a digital computer. U. S. Patent 3,699,528, October 1972. The copy bit for the Burroughs B6500 computer.
- R. S. Barton, B. A. Creech, B. A. Dent, E. A. Hauck, and W. M. McKeeman. Method and apparatus for establishing states in a data-processing system. U. S. Patent 3,611,312, October 1971.
- R. S. Barton, B. A. Creech, B. A. Dent, E. A. Hauck, and W. M. McKeeman. Procedure entry for a data processor employing a stack. U. S. Patent 3,548,384, December 1970.
- R. S. Barton, B. A. Creech, B. A. Dent, E. A. Hauck, and W. M. McKeeman. Data processing system having tree-structured stack implementation. U. S. Patent 3,546,677, December 1970.
- C. B. Carlson, B. A. Dent, and W. M. McKeeman. Address manipulation circuitry for a digital computer. U.S. Patent 3,510,847, May 1970. The index bit for the Burroughs B6500 computer.

VIDEO.

- W. M. McKeeman. Formal language: Practical parsing. In Formal Methods in Software Engineering, Lecture 23, Software Engineering Institute, Carnegie Mellon University, April 7, 1988. Video module.
- W. M. McKeeman and Randy Forgaard. Computer programming for science students. Listed as Natural Sciences 150SA, Instructional Services, University of California at Santa Cruz, 1979. 24 video modules. See also TV Teaching, W. M. McKeeman, *Teacher on the Hill*, Issue 14, December 1979.

THESES.

- W. M. McKeeman. *An Approach to Computer Language Design*. PhD thesis, Stanford University, April 1966. See also Technical Report CS48, Computer Science Department, Stanford University, August 31, 1966, 124 pages.
- W. M. McKeeman. The derived polytopes in Euclidian N-space. Master's thesis, George Washington University, June 1961. 108 pages.