

Mark R. Tuttle

8 Melanie Lane, Arlington, MA 02474, tuttle@acm.org, 781-648-1106, www.markrtuttle.com

Summary

May 27, 2016

- I am a mathematically-trained computer scientist specializing in algorithm design and testing, especially highly-concurrent, scalable, fault-tolerant algorithms as they appear in hardware, BIOS, operating systems, database systems, networks, and applications.
- I most recently led a team to build a software tool using symbolic execution and fuzzing to find functional and security vulnerabilities in BIOS as part of a production software flow.
- I love algorithms, I am fascinated by scale, and I want to work in an environment deploying algorithms and infrastructure on a modern, massive scale.

Employment

- Intel, Principal Engineer, 2006-present
- Hewlett Packard, Master Technologist, 2002-2005
- Compaq, promoted to Principal Member of Technical Staff, 1998-2002
- Digital Equipment Corporation, promoted to Consulting Engineer, 1989-1998

Major software projects (more details at www.markrtuttle.com)

- **Intel: Security bug hunting for BIOS.** Led team to develop software tool using symbolic execution to find functional and security vulnerabilities in BIOS, developed complete tool flow from code repository to bug reports complete with test cases to debug reported bugs, generated test cases achieving 2x code coverage improvement over fuzzing,
- **Intel: Data coherence over high-speed communication fabrics.** Developed a software tool to extract validation collateral directly from architectural design documents, build an architectural model, and formally prove architectural correctness with model checking. Demonstrated on Intel's first 14nm system-on-a-chip fabric..
- **HP: Electronic commerce and web search.** Developed algorithms for collaboration with untrusted parties, algorithms patented for recommendation systems, algorithms patented and deployed within the company for improved corporate web search.
- **DEC: Redo recovery after system crashes in database systems.** Developed a theory of redo recovery explaining all major database redo recovery techniques, leading to two patents for logical logging extending redo recovery into new domains (eventually to application recovery at Microsoft).

Recognition

- Division Recognition Award from Intel's most profitable division for validation pathfinding, 2015
- Best Paper Award for validation innovation, Formal Methods in Computer Aided Design, 2008
- Wrote 2 of the 100 most-cited papers published in ACM PODC distributed computing conference
- First place team, ACM International Programming Competition, 1983

Skill sets

- Languages: Developed in C, C++, Java, Ocaml, Perl, PHP, bash and cmd scripting, SQL, HTML, CCS. Written Scheme, CLU, LUA, IBM and x86 assembly, Pascal, APL, Fortran.
- Development environments: Linux, Windows, MySQL, Apache, Eclipse, CVS, SVN, GIT, gcc, gdb, valgrind. Familiar with continuous integration (Team City), unit testing (JUnit), fuzzing (Klocwork).
- Formal modeling and validation: I/O Automata, Murphi, Spin, HOL, ACL2, SAT and SMT solvers.

Education

- Massachusetts Institute of Technology, PhD in distributed computing, 5.0/5.0, 1989
- University of Nebraska-Lincoln, BS in math and computer science, 4.0/4.0, 1984