

Mark Rogers Tuttle

Intel Corporation
77 Reed Road
Hudson, MA 01749
(978) 553-3305
tuttle@acm.org

8 Melanie Lane
Arlington, MA 02474-2619
(781) 648-1106
mrtuttle@alum.mit.edu
www.markrtuttle.com

Research Interests

Distributed computation (design, analysis, and verification of distributed algorithms; models and semantics of distributed computation; formal methods for specification and verification of algorithms), sequential and parallel algorithms, general theory of computation.

Education

MASSACHUSETTS INSTITUTE OF TECHNOLOGY Cambridge, MA
Ph.D. in Computer Science, September 1989. Thesis Title: Knowledge and Distributed Computation. Thesis Advisor: Nancy A. Lynch. First place on computer algorithms Ph.D. qualifying exam; letter of commendation for exceptional performance in Computer System Architecture under Professor Arvind; IBM Graduate Fellowship; GTE Graduate Fellowship; Mortar Board National Foundation Fellowship; NSF Graduate Fellowship Competition, honorable mention; member, Sigma Xi.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY Cambridge, MA
M.S. in Computer Science, May 1987. Thesis Title: Hierarchical Correctness Proofs for Distributed Algorithms. Thesis Advisor: Nancy A. Lynch.

UNIVERSITY OF NEBRASKA–LINCOLN Lincoln, NE
B.S. in Mathematics and Computer Science, June 1984. With highest distinction. Ranked first in graduating class; member Phi Beta Kappa and Mortar Board; designated “Notable Person of the University,” Chancellor’s Scholar, Regents’ Scholar, and Joel Stebbins Scholar.

Awards

YOUNG ALUMNI ACHIEVEMENT AWARD, UNIVERSITY OF NEBRASKA–LINCOLN, MAY 1997

CERTIFICATE OF RECOGNITION, DIGITAL EQUIPMENT CORPORATION, WILDFIRE VERIFICATION PROJECT, APRIL 1998

NATIONAL CHAMPION, ACM NATIONAL SCHOLASTIC PROGRAMMING COMPETITION, UNIVERSITY OF NEBRASKA-LINCOLN TEAM, 1983

Employment

INTEL CORPORATION (2006–PRESENT) Hudson, MA
Principal Engineer (2006–present)

Protocol verification focusing on tools and methodologies for modeling, specifying, and proving the correctness of concurrent protocols. Examples include a tool for extracting a formal model from informal protocol design documents, and verification of transactional memory implementations.

Danny Dolev (Hebrew University) recognized two papers among the most highly cited papers ever published at ACM PODC in his keynote talk at PODC's 25th Anniversary Celebration.

HEWLETT PACKARD COMPANY (2002–2005) Cambridge, MA
COMPAQ COMPUTER CORPORATION (1998–2002) Cambridge, MA
DIGITAL EQUIPMENT CORPORATION (1989–1998) Cambridge, MA
Principal Member of Technical Staff/Master Technologist (1999–present)
Senior Member of Technical Staff/Consulting Engineer (1996–1999)
Member of Technical Staff (1989–1996)

My research at the Cambridge Research Lab focused on fault-tolerant distributed and parallel algorithms and their correctness. Work included formal methods for specification and verification of hardware and software, and promotion of these methods within the company; electronic commerce and web search; database systems; concurrent data structures; processor synchronization; real-time systems; security; models of computation; program correctness; complexity of algorithms; and computational biology.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY (1984–1989) Cambridge, MA
Member of the Theory of Distributed Systems research group at the Laboratory for Computer Science under Professor Nancy A. Lynch. Ph.D. thesis research studied the role of reasoning about processors' states of knowledge in the design and analysis of distributed algorithms. M.S. thesis research studied models of distributed computation and the design of hierarchical proof techniques for distributed algorithm verification.

IBM ALMADEN RESEARCH CENTER (SUMMER, 1988) San Jose, CA
Member of the Systems Fundamentals Department under Dr. Joseph Halpern. Worked with Joe Halpern on definitions of probabilistic knowledge, Danny Dolev on algorithms for processor renaming in asynchronous distributed systems, and Flaviu Cristian on the effects of network partitions on the consistency of replicated data.

IBM ALMADEN RESEARCH CENTER (AUGUST, 1987) San Jose, CA
Visitor in the theory group. Worked with Joe Halpern and Yoram Moses on the analysis of interactive and zero-knowledge proof systems in terms of a formal notion of knowledge.

Teaching Experience

MASSACHUSETTS INSTITUTE OF TECHNOLOGY (SPRING 1985, FALL 1987) Cambridge, MA
Teaching assistant for an undergraduate course on computer architecture. Duties included tutoring and grading papers for a recitation of 25 students, and helping students in the lab construct a small computer. Supervisor: Professor Steve Ward.

UNIVERSITY OF NEBRASKA–LINCOLN (FALL 1982, SPRING 1983, SPRING 1984) Lincoln, NE
Teaching assistant for a second semester calculus course. Duties included teaching two recitations of second semester calculus (about 30 students each); responsible for recitation lectures, homework, grading, and exams. Supervisors: Professors David Skoug and David Logan.

UNIVERSITY OF NEBRASKA–LINCOLN (FALL 1983) Lincoln, NE
Instructor for a precalculus course. Duties included teaching two sections of about 40

students each; responsible for lectures, homework, grading, and exams. Supervisor: Professor David Logan.

Publications

Conferences

1. Murali Talupur and Mark Tuttle. Going with the flow: Parameterized verification using message flows. In *Proceedings of the 8th International Conference on Formal Methods in Computer-Aided Design (FMCAD)*, pages 69–76, November 2008.
2. John O’Leary, Bratin Saha, and Mark R. Tuttle. Brief announcement: Model checking transactional memory with Spin. In *Proceedings of the 27th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, page 424, August 2008.
3. David James, Tim Leonard, John O’Leary, Murali Talupur, and Mark R. Tuttle. Brief announcement: Extracting models from design documents with Mapster. In *Proceedings of the 27th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, page 456, August 2008.
4. Noga Alon, Chen Avin, Michal Koucký, Gady Kozma, Zvi Lotker, and Mark R. Tuttle. Many random walks are faster than one. In *Proceedings of the 20th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 119–128, June 2008.
5. John O’Leary, Bratin Saha, and Mark R. Tuttle. Model checking transactional memory. In *ETAPS Workshop on Designing Correct Circuits (DCC)*, April 2008.
6. Murali Talupur, Sava Krstic, John O’Leary, and Mark R. Tuttle. Parametric verification of industrial cache protocols. In *ETAPS Workshop on Designing Correct Circuits (DCC)*, April 2008.
7. Ariel Cohen, John W. O’Leary, Amir Pnueli, Mark R. Tuttle, and Lenore D. Zuck. Verifying correctness of transactional memories. In *Proceedings of the 7th International Conference on Formal Methods in Computer-Aided Design (FMCAD)*, pages 37–44, November 2007.
8. Zvi Lotker, Boaz Patt-Shamir, and Mark R. Tuttle. Publish and perish: Definition and analysis of an n -person publication impact game. In *Proceedings of the 18th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 11–18, August 2006.
9. Zvi Lotker, Boaz Patt-Shamir, and Mark Tuttle. Brief Announcement: Timing games and shared memory. In *Proceedings of the 19th International Symposium on Distributed Computing (DISC)*, September 2005.
10. Baruch Awerbuch, Yossi Azar, Zvi Lotker, Boaz Patt-Shamir, and Mark Tuttle. Collaborate with strangers to find own preferences. In *Proceedings of the 17th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 263–269, July 2005.
11. Baruch Awerbuch, Boaz Patt-Shamir, David Peleg, and Mark Tuttle. Adaptive col-laboration in peer-to-peer systems. In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 71–80, June 2005.
12. Baruch Awerbuch, Boaz Patt-Shamir, David Peleg, and Mark Tuttle. Improved rec-ommendation systems. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1174–1183, January 2005.

13. Maurice Herlihy, Sergio Rajsbaum, and Mark Tuttle. An axiomatic approach to computing the connectivity of synchronous and asynchronous systems. In *Proceedings of the Sixth Workshop on Geometric and Topological Methods in Concurrency and Distributed Computing (GETCO)*, Electronic Notes in Theoretical Computer Science. Elsevier, October 2004.
14. Baruch Awerbuch, Boaz Patt-Shamir, David Peleg, and Mark Tuttle. Collaboration of untrusting peers with changing interests. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC)*, pages 112–119, May 2004.
15. Sandip Ray, John Matthews, and Mark Tuttle. Certifying compositional model checking algorithms in ACL2. In *Proceedings of the 4th International Workshop on the ACL2 Theorem Prover and Its Applications (ACL2)*, July 2003.
16. David Lomet and Mark Tuttle. A theory of redo recovery. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 397–406. ACM, June 2003.
17. Leslie Lamport, John Matthews, Mark Tuttle, and Yuan Yu. Specifying and verifying systems with TLA+. In *Proceedings of the Tenth ACM SIGOPS European Workshop: Can we really depend on an OS?*, pages 45–48, September 2002.
18. Maurice Herlihy, Sergio Rajsbaum, and Mark Tuttle. A new synchronous lower bound for set agreement. In *Proceedings of the 15th International Conference on Distributed Computing (DISC)*, volume 2180 of *Lecture Notes in Computer Science*, pages 136–150. Springer-Verlag, October 2001.
19. Maurice Herlihy, Sergio Rajsbaum, and Mark Tuttle. An overview of synchronous message-passing and topology. In *Proceedings of the Workshop on Geometry and Topology in Concurrency Theory (GETCO)*, volume NS-00-3 of *Basic Research in Computer Science Notes Series*. Department of Computer Science, University of Aarhus, August 2000.
20. Homayoon Akhiani, Damien Doligez, Paul Harter, Leslie Lamport, Joshua Scheid, Mark Tuttle, and Yuan Yu. Cache coherence verification with TLA+. In *Proceedings of the World Congress On Formal Methods In The Development Of Computing Systems, FM'99, Industrial Panel*, volume 1709 of *Lecture Notes in Computer Science*, page 1871, September 1999.
21. David B. Lomet and Mark R. Tuttle. Logical logging to extend recovery to new domains. In Alex Delis, Christos Faloutsos, and Shahram Ghandeharizadeh, editors, *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 73–84. ACM, June 1999.
22. Maurice Herlihy, Sergio Rajsbaum, and Mark R. Tuttle. Unifying synchronous and asynchronous message-passing models. In *Proceedings of the 17th Annual ACM Symposium on Principles of Distributed Computing*, pages 133–142. ACM, June 1998.
23. David Lomet and Mark R. Tuttle. Redo recovery after system crashes. In Umesh Dayal, Peter Gray, and Shojiro Nishio, editors, *Proceedings of the 21st International Conference on Very Large Data Bases*, pages 457–468, Zurich, September 1995. Morgan Kaufmann Publishers.
24. Soma Chaudhuri and Mark R. Tuttle. Fast increment registers. In Gerard Tel and Paul Vitányi, editors, *Proceedings of the 8th International Workshop on Distributed*

Algorithms, volume 857 of *Lecture Notes in Computer Science*, pages 74–88. Springer-Verlag, Berlin, October 1994.

25. Soma Chaudhuri, Maurice Herlihy, Nancy Lynch, and Mark R. Tuttle. A tight lower bound for k -set agreement. In *Proceedings of the 34th IEEE Symposium on Foundations of Computer Science*, pages 206–215, November 1993.
26. Soma Chaudhuri, Maurice Herlihy, Nancy Lynch, and Mark R. Tuttle. A tight lower bound for processor coordination. In *Third International Workshop on Responsive Computer Systems*, pages 4–15, September 1993.
27. Martín Abadi and Mark R. Tuttle. A semantics for a logic of authentication. In *Proceedings of the 10th Annual ACM Symposium on Principles of Distributed Computing*, pages 201–216. ACM, August 1991.
28. Michael Merritt, Francesmary Modugno, and Mark R. Tuttle. Time-constrained automata. In J. C. M. Baeten and J. F. Groote, editors, *Proceedings of the Second International Conference on Concurrency Theory (Concur'91)*, volume 527 of *Lecture Notes in Computer Science*, pages 408–423. Springer-Verlag, Berlin, August 1991.
29. Gil Neiger and Mark R. Tuttle. Common knowledge and consistent simultaneous coordination. In J. van Leeuwen and N. Santoro, editors, *Proceedings of the 4th International Workshop on Distributed Algorithms*, volume 486 of *Lecture Notes in Computer Science*, pages 334–352. Springer-Verlag, September 1990.
30. Maurice P. Herlihy and Mark R. Tuttle. Wait-free computation in message-passing systems. In *Proceedings of the 9th Annual ACM Symposium on Principles of Distributed Computing*, pages 347–362. ACM, August 1990.
31. Joseph Y. Halpern and Mark R. Tuttle. Knowledge, probability, and adversaries. In *Proceedings of the 8th Annual ACM Symposium on Principles of Distributed Computing*, pages 103–118. ACM, August 1989.
32. Joseph Y. Halpern, Yoram Moses, and Mark R. Tuttle. A knowledge-based analysis of zero knowledge. In *Proceedings of the 20th ACM Symposium on Theory of Computing*, pages 132–147, May 1988.
33. Nancy A. Lynch and Mark R. Tuttle. Hierarchical correctness proofs for distributed algorithms. In *Proceedings of the 6th Annual ACM Symposium on Principles of Distributed Computing*, pages 137–151. ACM, August 1987.
34. Yoram Moses and Mark R. Tuttle. Programming simultaneous actions using common knowledge. In *Proceedings of the 27th IEEE Symposium on Foundations of Computer Science*, pages 208–221. IEEE, October 1986.

Journals

1. Zvi Lotker, Boaz Patt-Shamir, and Mark R. Tuttle. A game of timing and visibility. *Games and Economic Behavior*, 62(22):643–660, March 2008.
2. Baruch Awerbuch, Yossi Azar, Zvi Lotker, Boaz Patt-Shamir, and Mark Tuttle. Collaborate with strangers to find own preferences. *Theory of Computing Systems*, 42(1):27–41, January 2008.
3. Rajeev Joshi, Leslie Lamport, John Matthews, Serdar Tasiran, Mark Tuttle, and Yuan Yu. Checking cache-coherence protocols with TLA+. *Formal Methods in System Design: Special Issue on Industrial Applications*, 22(2):125–131, March 2003.

4. Maurice Herlihy, Sergio Rajsbaum, and Mark Tuttle. An overview of synchronous message-passing and topology. *Electronic Notes in Theoretical Computer Science*, 39(2), 2001.
5. Soma Chaudhuri, Maurice Herlihy, Nancy Lynch, and Mark R. Tuttle. Tight bounds for k -set agreement. *Journal of the ACM*, 47(5):912–943, September 2000.
6. Soma Chaudhuri, Maurice Herlihy, and Mark R. Tuttle. Wait-free implementations in message-passing systems. *Theoretical Computer Science*, 220(1):211–245, June 1999.
7. Joseph Y. Halpern and Mark R. Tuttle. Knowledge, probability, and adversaries. *Journal of the ACM*, 40(4):917–962, September 1993.
8. Gil Neiger and Mark R. Tuttle. Common knowledge and consistent simultaneous coordination. *Distributed Computing*, 6(3):181–192, 1993.
9. Nancy A. Lynch and Mark R. Tuttle. An introduction to input/output automata. *CWI Quarterly*, 2(3):219–246, September 1989.
10. Yoram Moses and Mark R. Tuttle. Programming simultaneous actions using common knowledge. *Algorithmica*, 3(1):121–169, 1988.

Book chapters

1. David Lomet and Mark R. Tuttle. Redo recovery after system crashes. In Vijay Kumar and Meichun Hsu, editors, *Recovery Mechanisms in Database Systems*, pages 101–124. Prentice Hall, Upper Saddle River, New Jersey, 1999.

Reports

1. Soma Chaudhuri, Maurice Herlihy, Nancy Lynch, and Mark R. Tuttle. Tight bounds for k -set agreement. Technical Report 98/4, DEC Cambridge Research Lab, May 1998.
2. Soma Chaudhuri, Maurice Herlihy, and Mark R. Tuttle. Wait-free implementations in message-passing systems. Technical Report 98/5, DEC Cambridge Research Lab, May 1998.
3. Mark R. Tuttle. Knowledge and distributed computation. Technical Report MIT/LCS/TR-477, Massachusetts Institute of Technology, Laboratory for Computer Science, May 1990.
4. Nancy A. Lynch and Mark R. Tuttle. Hierarchical correctness proofs for distributed algorithms. Technical Report MIT/LCS/TR-387, MIT Laboratory for Computer Science, April 1987.

Other

1. Leslie Lamport, Mark Tuttle, and Yuan Yu. The wildfire verification challenge problem. See <http://research.microsoft.com/users/lamport/tla/wildfire-challenge.html>, July 2000.
2. Homayoon Akhiani, Damien Doligez, Paul Harter, Leslie Lamport, Joshua Scheid, Mark Tuttle, and Yuan Yu. TLA+ verification of cache coherence protocols. Unpublished manuscript, February 1999.
3. Hagit Attiya and Mark R. Tuttle. Bounds for slotted ℓ -exclusion. Technical memo, MIT, February 1989.
4. Mark R. Tuttle. A game-theoretic characterization of eventual common knowledge. Technical memo, MIT, October 1988.

Patents

1. Baruch Awerbuch, Boaz Patt-Shamir, David Peleg, Mark Tuttle. Improved recommendation systems. Application pending, December 2004.
2. Eytan Adar, Boaz Patt-Shamir, Mark Tuttle. Collaborative web searches. Application pending, December 2004.
3. David B. Lomet and Mark R. Tuttle. Database Computer System Using Logical Logging to Extend Recovery. Application pending, April 1999.

Invited Talks

1. *Reasoning about distributed computation*. Computer Science and Engineering 40th Anniversary Distinguished Alumni Talks, University of Nebraska - Lincoln, September 2008.
2. *Analyzing Cache Coherence with TLA+*. Workshop on Formal Specification and Verification Methods for Shared Memory Systems at International Conference on Formal Methods in Computer-Aided Design (FMCAD), October 2000.
3. *Using TLA+ at COMPAQ*. TLA+ User Group Meeting at World Congress On Formal Methods In The Development Of Computing Systems (FM), September, 1999.
4. *Unifying synchronous and asynchronous message-passing models*. Computer Science Colloquium series, Brown University, March 1999.
5. *Lower bounds for agreement*. Mathematics and Statistics Colloquium series, University of Nebraska–Lincoln, May 1997.
6. *A tight lower bound for k-set agreement*. Computer Science Colloquium series, Dartmouth University, January 1995.
7. *Build models, not logics*. Panel, Computer Security Foundations Workshop, June 1993.
8. *How fast are wait-free concurrent objects?* POCS Seminar Series, MIT Laboratory for Computer Science, April 1993.

Theses

1. Costas Busch, *A Study on Distributed Structures*, Brown University, 2000, thesis committee member.
2. George Varghese, *Self-Stabilization by Local Checking and Correction*, MIT, 1992, thesis committee member.

Service

1. Computer Science and Engineering Industrial Advisory Panel, University of Nebraska–Lincoln, (1997-2000).
2. Founder and Treasurer, Theoretical Aspects of Reasoning about Knowledge, a non-profit corporation to run a biannual scientific conference by the same name (1991-present).
3. International Math Olympiad, US Team training center staff member (summer 1984).
4. Chair, Steering Committee, ACM Principles of Distributed Computing, 2003-2006.
5. Program Committees for ACM Symposium on Principles of Distributed Computing (PODC 1994, 1996, 2001, 2004), EATCS International Symposium on Distributed Computing (DISC 2000, 2008, 2009), IEEE Symposium on Network Computing and Applications (NAC 2008, 2009), International Conference on Distributed Computing Systems (ICDCS 2007), International Conference on Distributed Computing and Networking (ICDCN 2006), International Conference On Principles Of Distributed

Systems (OPODIS 2006), and Theoretical Aspects of Reasoning about Knowledge (TARK 1994).

6. Journal referee, major journals include Journal of the ACM (JACM), ACM Transactions on Computer Systems (TOCS), ACM Transactions on Database Systems (TODS), ACM Transactions on Programming Languages and Systems (TOPLAS), Distributed Computing, IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Software Engineering (TSE), Information and Computation, Information and Communication, Information Processing Letters (IPL), International Journal of Parallel Processing (IJPP), Theory of Computer Systems, Journal on Software and System Modeling, SIAM Journal of Computation.
7. Conference referee, major conferences include ACM Principles of Distributed Computing (PODC), ACM Symposium on Operating System Principles (SOSP), ACM Symposium on the Theory of Computing (STOC), IEEE Symposium on Foundations of Computer Science (FOCS), Theoretical Aspects of Reasoning about Knowledge (TARK), Computer Aided Verification (CAV), Design Automation Conference (DAC), European Symposium on Algorithms (ESA), Sigplan, Tricomm.
8. Grant reviews for National Science Foundation.
9. Member of Association for Computing Machinery, Sigma Xi.

Last updated on November 16, 2008.