Massachusetts Institute of Technology — Handout 3
6.852: Distributed Algorithms
Prof. Nancy Lynch — September 4, 2003

Supplementary Reading List

1 The number of rounds required for consensus

[1] Marcos Kawazoe Aguilera and Sam Toueg. A simple bivalency proof that t-resilient consensus requires $t+1$ rounds. *Information Processing Letters*, 71(3-4):155-158, August 1999.
`http://theory.lcs.mit.edu/classes/6.852/03/papers/bivalency-full.ps.`

[2] Idit Keidar and Sergio Rajsbaum. On the cost of fault-tolerant consensus when there are no faults -a tutorial. Technical Report MIT-LCS-TR-821, May 2001. Preliminary version in *SIGACT News*, 32(2):45-63, Distributed Computing column, June 2001 (published in May 15th).
`http://theory.lcs.mit.edu/classes/6.852/03/papers/TR821.ps.gz.`

2 The wait-free consensus hierarchy

[1] Prasad Jayanti. Wait-free computing. In Jean-Michel Hélary, Michel Raynal (Eds.): *Distributed Algorithms, 9th International Workshop, WDAG '95, Le Mont-Saint-Michel, France, September 13-15, 1995, (Proceedings)*, volume 972 of *Lecture Notes in Computer Science*, Springer 1995.
Online version not available.

[2] Maurice Herlihy. Wait-free synchronization. *ACM Transactions on Programming Languages and Systems*, 13(1):124–149, January 1991.
`http://theory.lcs.mit.edu/classes/6.852/03/papers/p124-herlihy.pdf.`

[3] Prasad Jayanti and Sam Toueg. Some results on the impossibility, universality, and decidability of consensus. *6th International Workshop on Distributed Algorithms (WDAG)*, pages 69–84, Haifa Israel, 1992.
Online version not available.

[4] Prasad Jayanti. Robust wait-free hierarchies. *Journal of the ACM*, 44(4): 592-614, 1997.
`http://theory.lcs.mit.edu/classes/6.852/03/papers/p592-jayanti.pdf.`

[5] Gary Peterson, Rida Bazzi, and Gil Neiger. A gap theorem for consensus types. *In Proceedings of the Thirteenth ACM Symposium on Principles of Distributed Computing*, pages 344–353, Los Angeles, CA, August 1994.
`http://theory.lcs.mit.edu/classes/6.852/03/papers/p344-peterson.pdf.`

[6] Rida A. Bazzi, Gil Neiger, Gary L. Peterson. On the use of registers in achieving wait-free consensus. *Proceedings of the 13th ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, pages 354–362, Los Angeles, CA, August 1994.
`http://theory.lcs.mit.edu/classes/6.852/03/papers/p354-bazzi.pdf.`

[7] Elizabeth Borowsky, Eli Gafni, Yehuda Afek. Consensus power makes (some) sense! *In Proceedings of the Thirteenth ACM Symposium on Principles of Distributed Computing*, pages 363–372, Los Angeles, CA, August 1994.
`http://theory.lcs.mit.edu/classes/6.852/03/papers/p363-borowsky.pdf.`

[8] Wai-Kau Lo and Vassos Hadzilacos. All of us are smarter than any of us: nondeterministic wait-free hierarchies are not robust. *SIAM Journal on Computing*, 30(3):689–728, 2001.
`http://theory.lcs.mit.edu/classes/6.852/03/papers/p689-lo.ps`

3 Wait-free vs. f-fault-tolerant data objects

[1] Elizabeth Borowsky, Eli Gafni, Nancy Lynch, and Sergio Rajsbaum. The BG distributed simulation algorithm. *Distributed Computing*, 14:127-146, 2001.
`http://theory.lcs.mit.edu/classes/6.852/03/papers/bg-dist-sim.pdf`

[2] T.D. Chandra, V. Hadzilacos, P. Jayanti and S. Toueg. Wait-freedom vs. t-resiliency and the robustness of the h^r_m hierarchy. *Proceedings of the 13th ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, pages 334-343, Los Angeles, CA, August 1994.
`http://theory.lcs.mit.edu/classes/6.852/03/papers/p334-chandra.pdf.`

[3] T. D. Chandra, V. Hadzilacos, P. Jayanti, and S. Toueg. Generalized irreducibility of consensus and the equivalence of t-resilient and wait-free implementations of consensus. Submitted for publication.
`href=http://theory.lcs.mit.edu/classes/6.852/03/papers/chandra-etal.ps`

[4] W.K. Lo and V. Hadzilacos. On the power of shared object types to implement one-resilient consensus. *Distributed Computing*, 13(4):219–238, 2000.
`http://theory.lcs.mit.edu/classes/6.852/03/papers/p219-lo.pdf.`

[5] Paul Attie, Nancy Lynch, and Sergio Rajsbaum. Boosting fault-tolerance in asynchronous message passing systems is impossible. Technical Report MIT-LCS-TR-877, MIT Laboratory for Computer Science, Cambridge, MA, December 2002.
`http://theory.lcs.mit.edu/classes/6.852/03/papers/MIT-LCS-TR-877.ps.gz`

4 Vector timestamps

[1] Friedemann Mattern. Virtual time and global states of distributed systems. In Michel Cosnard et al., editors, *Parallel and Distributed Algorithms: Proceedings of the International Workshop on Parallel and Distributed Algorithms* (Chateau de Bonas, Gers, France, October, 1988), pages 215–226. North Holland, 1989. (Reprinted in: Z. Yang, T.A. Marsland (Eds.), "Global States and Time in Distributed Systems", IEEE, 1994, pp. 123-133.)
`http://theory.lcs.mit.edu/classes/6.852/03/papers/VirtTimeGlobStatesFull.pdf.`

[2] Colin Fidge. Logical time in distributed computing systems. *IEEE Computer*, 24(8):28–33, August 1991.
`http://theory.lcs.mit.edu/classes/6.852/03/papers/ieeecomputer.pdf`

5 Reliable broadcast

[1] Vassos Hadzilacos and Sam Toueg. Fault-tolerant broadcasts and related problems. In Sape Mullender, editor, *Distributed Systems*, chapter 5, pages 97–145. ACM Press and Addison-Wesley, 1993. Second Edition.

[2] V. Hadzilacos and S. Toueg. A modular approach to the specification and implementation of fault-tolerant broadcasts. Technical Report TR94-1425, Department of Computer Science, Cornell University, Ithaca NY, May 1994.
`http://theory.lcs.mit.edu/classes/6.852/03/papers/TR94-1425.ps.gz`

6 Failure detectors and consensus

[1] Tushar Deepak Chandra and Sam Toueg. Unreliable failure detectors for reliable distributed systems. *Journal of the ACM*, 43(2):225–267, March 1996.
`http://theory.lcs.mit.edu/classes/6.852/03/papers/CT96-JACM.ps`

[2] Tushar Deepak Chandra, Vassos Hadzilacos, and Sam Toueg. The weakest failure detector for solving consensus. *Journal of the ACM*, 43(4):685–722, July 1996.
`http://theory.lcs.mit.edu/classes/6.852/03/papers/Chandra/jacm96.pdf`

[3] Marcos Kawazoe Aguilera and Wei Chen and Sam Toueg. Failure detection and consensus in the crash-recovery model. To appear in *Distributed Computing*.
`http://theory.lcs.mit.edu/classes/6.852/03/papers/crashrecoveryfinaldcversion.ps`

[4] L. Lamport. The part-time parliament. *ACM Transactions on Computer Systems*, 16(2):133–169, May 1998. Also Research Report 49, Digital Equipment Corporation Systems Research Center, Palo Alto, CA, September 1989.
`http://theory.lcs.mit.edu/classes/6.852/03/papers/lamport-paxos.ps`

7 Self-stabilization

[1] Edsger W. Dijkstra. Self-stabilizing systems in spite of distributed control. *Communication of the ACM*, 17(11):643–644, November 1974.
`http://theory.lcs.mit.edu/classes/6.852/03/papers/p643-Dijkstra.pdf`

[2] Shlomi Dolev. *Self-Stabilization*. MIT Press, 2000.

8 Dynamic distributed algorithms

[1] Nancy Lynch and Alex Shvartsman. RAMBO: A reconfigurable atomic memory service for dynamic networks. Technical Report MIT-LCS-TR-856, MIT Laboratory for Computer Science, Cambridge, MA, 2002.
`http://theory.lcs.mit.edu/classes/6.852/03/papers/rambo-tr.ps`

[2] Seth Gilbert, Nancy Lynch, and Alex Shvartsman. RAMBO II: Rapidly reconfigurable atomic memory for dynamic networks. Proceedings of the International Conference on Dependable Systems and Networks (DSN), San Francisco, CA, pages 259-268, June 22nd - 25th, 2003.
`http://theory.lcs.mit.edu/classes/6.852/03/papers/RamboII-DSN.ps.gz`

[3] Shlomi Dolev, Seth Gilbert, Nancy Lynch, Alex Shvartsman, and Jennifer Welch. GeoQuorums: Implementing atomic memory in ad hoc networks. To appear in *DISC 2003: 17th International Symposium on Distributed Computing*, Sorrento, Italy, October, 2003.
`http://theory.lcs.mit.edu/classes/6.852/03/papers/GeoQuoroms-DISC.ps.gz`