

Contents

1	Introduction	1
1.1	<i>Distributed Systems</i>	1
1.2	<i>Theory of Distributed Computing</i>	2
1.3	<i>Overview</i>	3
1.4	<i>Relationship of Theory to Practice</i>	4

Part I Fundamentals

2	Basic Algorithms in Message-Passing Systems	9
2.1	<i>Formal Models for Message Passing Systems</i>	9
2.1.1	<i>Systems</i>	9
2.1.2	<i>Complexity Measures</i>	13
2.1.3	<i>Pseudocode Conventions</i>	14
2.2	<i>Broadcast and Convergecast on a Spanning Tree</i>	15
2.3	<i>Flooding and Building a Spanning Tree</i>	19
2.4	<i>Constructing a Depth-First Search Spanning Tree for a Specified Root</i>	23
2.5	<i>Constructing a Depth-First Search Spanning Tree without a Specified Root</i>	25

3	<i>Leader Election in Rings</i>	31
3.1	<i>The Leader Election Problem</i>	31
3.2	<i>Anonymous Rings</i>	32
3.3	<i>Asynchronous Rings</i>	34
3.3.1	<i>An $O(n^2)$ Algorithm</i>	34
3.3.2	<i>An $O(n \log n)$ Algorithm</i>	35
3.3.3	<i>An $\Omega(n \log n)$ Lower Bound</i>	38
3.4	<i>Synchronous Rings</i>	42
3.4.1	<i>An $O(n)$ Upper Bound</i>	43
3.4.2	<i>An $\Omega(n \log n)$ Lower Bound for Restricted Algorithms</i>	48
4	<i>Mutual Exclusion in Shared Memory</i>	59
4.1	<i>Formal Model for Shared Memory Systems</i>	60
4.1.1	<i>Systems</i>	60
4.1.2	<i>Complexity Measures</i>	62
4.1.3	<i>Pseudocode Conventions</i>	62
4.2	<i>The Mutual Exclusion Problem</i>	63
4.3	<i>Mutual Exclusion Using Powerful Primitives</i>	65
4.3.1	<i>Binary Test&Set Registers</i>	65
4.3.2	<i>Read-Modify-Write Registers</i>	66
4.3.3	<i>Lower Bound on the Number of Memory States</i>	69
4.4	<i>Mutual Exclusion Using Read/Write Registers</i>	71
4.4.1	<i>The Bakery Algorithm</i>	71
4.4.2	<i>A Bounded Mutual Exclusion Algorithm for Two Processors</i>	73
4.4.3	<i>A Bounded Mutual Exclusion Algorithm for n Processors</i>	77
4.4.4	<i>Lower Bound on the Number of Read/Write Registers</i>	80
4.4.5	<i>Fast Mutual Exclusion</i>	84
5	<i>Fault-Tolerant Consensus</i>	91
5.1	<i>Synchronous Systems with Crash Failures</i>	92
5.1.1	<i>Formal Model</i>	92
5.1.2	<i>The Consensus Problem</i>	93
5.1.3	<i>A Simple Algorithm</i>	93
5.1.4	<i>Lower Bound on the Number of Rounds</i>	95
5.2	<i>Synchronous Systems with Byzantine Failures</i>	99

5.2.1	<i>Formal Model</i>	100
5.2.2	<i>The Consensus Problem Revisited</i>	100
5.2.3	<i>Lower Bound on the Ratio of Faulty Processors</i>	101
5.2.4	<i>An Exponential Algorithm</i>	103
5.2.5	<i>A Polynomial Algorithm</i>	106
5.3	<i>Impossibility in Asynchronous Systems</i>	108
5.3.1	<i>Shared Memory—The Wait-Free Case</i>	109
5.3.2	<i>Shared Memory—The General Case</i>	111
5.3.3	<i>Message Passing</i>	119
6	<i>Causality and Time</i>	125
6.1	<i>Capturing Causality</i>	126
6.1.1	<i>The Happens-Before Relation</i>	126
6.1.2	<i>Logical Clocks</i>	128
6.1.3	<i>Vector Clocks</i>	129
6.1.4	<i>Shared Memory Systems</i>	133
6.2	<i>Examples of Using Causality</i>	133
6.2.1	<i>Consistent Cuts</i>	134
6.2.2	<i>A Limitation of the Happens-Before Relation: The Session Problem</i>	137
6.3	<i>Clock Synchronization</i>	140
6.3.1	<i>Modeling Physical Clocks</i>	140
6.3.2	<i>The Clock Synchronization Problem</i>	143
6.3.3	<i>The Two Processors Case</i>	144
6.3.4	<i>An Upper Bound</i>	146
6.3.5	<i>A Lower Bound</i>	148
6.3.6	<i>Practical Clock Synchronization: Estimating Clock Differences</i>	149
 <i>Part II Simulations</i>		
7	<i>A Formal Model for Simulations</i>	157
7.1	<i>Problem Specifications</i>	157
7.2	<i>Communication Systems</i>	158
7.2.1	<i>Asynchronous Point-to-Point Message Passing</i>	159
7.2.2	<i>Asynchronous Broadcast</i>	159
7.3	<i>Processes</i>	160
7.4	<i>Admissibility</i>	163

7.5	<i>Simulations</i>	164
7.6	<i>Pseudocode Conventions</i>	165
8	<i>Broadcast and Multicast</i>	167
8.1	<i>Specification of Broadcast Services</i>	168
8.1.1	<i>The Basic Service Specification</i>	168
8.1.2	<i>Broadcast Service Qualities</i>	169
8.2	<i>Implementing a Broadcast Service</i>	171
8.2.1	<i>Basic Broadcast Service</i>	172
8.2.2	<i>Single-Source FIFO Ordering</i>	172
8.2.3	<i>Totally Ordered Broadcast</i>	172
8.2.4	<i>Causality</i>	175
8.2.5	<i>Reliability</i>	177
8.3	<i>Multicast in Groups</i>	179
8.3.1	<i>Specification</i>	180
8.3.2	<i>Implementation</i>	181
8.4	<i>An Application: Replication</i>	183
8.4.1	<i>Replicated Database</i>	183
8.4.2	<i>The State Machine Approach</i>	183
9	<i>Distributed Shared Memory</i>	189
9.1	<i>Linearizable Shared Memory</i>	190
9.2	<i>Sequentially Consistent Shared Memory</i>	192
9.3	<i>Algorithms</i>	193
9.3.1	<i>Linearizability</i>	193
9.3.2	<i>Sequential Consistency</i>	194
9.4	<i>Lower Bounds</i>	198
9.4.1	<i>Adding Time and Clocks to the Layered Model</i>	198
9.4.2	<i>Sequential Consistency</i>	199
9.4.3	<i>Linearizability</i>	199
10	<i>Fault-Tolerant Simulations of Read/Write Objects</i>	207
10.1	<i>Fault-Tolerant Shared Memory Simulations</i>	208
10.2	<i>Simple Read/Write Register Simulations</i>	209
10.2.1	<i>Multi-Valued from Binary</i>	210
10.2.2	<i>Multi-Reader from Single-Reader</i>	215
10.2.3	<i>Multi-Writer from Single-Writer</i>	219
10.3	<i>Atomic Snapshot Objects</i>	222

10.3.1	<i>Handshaking Procedures</i>	223
10.3.2	<i>A Bounded Memory Simulation</i>	225
10.4	<i>Simulating Shared Registers in Message-Passing Systems</i>	229
11	<i>Simulating Synchrony</i>	239
11.1	<i>Synchronous Message-Passing Specification</i>	240
11.2	<i>Simulating Synchronous Processors</i>	241
11.3	<i>Simulating Synchronous Processors and Synchronous Communication</i>	243
11.3.1	<i>A Simple Synchronizer</i>	243
11.3.2	<i>Application: Constructing a Breadth-First Search Tree</i>	247
11.4	<i>Local vs. Global Simulations</i>	247
12	<i>Improving the Fault Tolerance of Algorithms</i>	251
12.1	<i>Overview</i>	251
12.2	<i>Modeling Synchronous Processors and Byzantine Failures</i>	253
12.3	<i>Simulating Identical Byzantine Failures on Top of Byzantine Failures</i>	255
12.3.1	<i>Definition of Identical Byzantine</i>	255
12.3.2	<i>Simulating Identical Byzantine</i>	256
12.4	<i>Simulating Omission Failures on Top of Identical Byzantine Failures</i>	258
12.4.1	<i>Definition of Omission</i>	259
12.4.2	<i>Simulating Omission</i>	259
12.5	<i>Simulating Crash Failures on Top of Omission Failures</i>	264
12.5.1	<i>Definition of Crash</i>	264
12.5.2	<i>Simulating Crash</i>	265
12.6	<i>Application: Consensus in the Presence of Byzantine Failures</i>	268
12.7	<i>Asynchronous Identical Byzantine on Top of Byzantine Failures</i>	269
12.7.1	<i>Definition of Asynchronous Identical Byzantine</i>	269
12.7.2	<i>Definition of Asynchronous Byzantine</i>	270
12.7.3	<i>Simulating Asynchronous Identical Byzantine</i>	270
13	<i>Fault-Tolerant Clock Synchronization</i>	277

13.1	<i>Problem Definition</i>	277
13.2	<i>The Ratio of Faulty Processors</i>	279
13.3	<i>A Clock Synchronization Algorithm</i>	284
13.3.1	<i>Timing Failures</i>	284
13.3.2	<i>Byzantine Failures</i>	290
13.4	<i>Practical Clock Synchronization: Identifying Faulty Clocks</i>	291
<i>Part III Advanced Topics</i>		
14	<i>Randomization</i>	297
14.1	<i>Leader Election: A Case Study</i>	297
14.1.1	<i>Weakening the Problem Definition</i>	297
14.1.2	<i>Synchronous One-Shot Algorithm</i>	299
14.1.3	<i>Synchronous Iterated Algorithm and Expectation</i>	300
14.1.4	<i>Asynchronous Systems and Adversaries</i>	302
14.1.5	<i>Impossibility of Uniform Algorithms</i>	303
14.1.6	<i>Summary of Probabilistic Definitions</i>	303
14.2	<i>Mutual Exclusion with Small Shared Variables</i>	305
14.3	<i>Consensus</i>	308
14.3.1	<i>The General Algorithm Scheme</i>	309
14.3.2	<i>A Common Coin with Constant Bias</i>	314
14.3.3	<i>Tolerating Byzantine Failures</i>	315
14.3.4	<i>Shared Memory Systems</i>	316
15	<i>Wait-Free Simulations of Arbitrary Objects</i>	321
15.1	<i>Example: A FIFO Queue</i>	322
15.2	<i>The Wait-Free Hierarchy</i>	326
15.3	<i>Universality</i>	327
15.3.1	<i>A Nonblocking Simulation Using Compare&Swap</i>	328
15.3.2	<i>A Nonblocking Algorithm Using Consensus Objects</i>	329
15.3.3	<i>A Wait-Free Algorithm Using Consensus Objects</i>	332
15.3.4	<i>Bounding the Memory Requirements</i>	335
15.3.5	<i>Handling Nondeterminism</i>	337

15.3.6	<i>Employing Randomized Consensus</i>	338
16	<i>Problems Solvable in Asynchronous Systems</i>	343
16.1	<i>k-Set Consensus</i>	344
16.2	<i>Approximate Agreement</i>	352
16.2.1	<i>Known Input Range</i>	352
16.2.2	<i>Unknown Input Range</i>	354
16.3	<i>Renaming</i>	356
16.3.1	<i>The Wait-Free Case</i>	357
16.3.2	<i>The General Case</i>	359
16.3.3	<i>Long-Lived Renaming</i>	360
16.4	<i>k-Exclusion and k-Assignment</i>	361
16.4.1	<i>An Algorithm for k-Exclusion</i>	362
16.4.2	<i>An Algorithm for k-Assignment</i>	364
17	<i>Solving Consensus in Eventually Stable Systems</i>	369
17.1	<i>Preserving Safety in Shared Memory Systems</i>	370
17.2	<i>Failure Detectors</i>	372
17.3	<i>Solving Consensus using Failure Detectors</i>	373
17.3.1	<i>Solving Consensus with $\circ S$</i>	373
17.3.2	<i>Solving Consensus with S</i>	375
17.3.3	<i>Solving Consensus with Ω</i>	376
17.4	<i>Implementing Failure Detectors</i>	377
17.5	<i>State Machine Replication with Failure Detectors</i>	377
	<i>References</i>	381
	<i>Index</i>	401