Contents i

List of Algorithms vii

Preface ix

Part I Fundamentals 1

1 Introduction 3
- 1.1 Distributed Systems . . . 3
- 1.2 Theory of Distributed Computing . . . 4
- 1.3 Overview . . . 5
- 1.4 Relationship of Theory to Practice . . . 6

2 Basic Algorithms in Message Passing Systems 9
- 2.1 Formal Model for Message Passing Systems . . . 9
  - 2.1.1 Systems . . . 9
  - 2.1.2 Complexity Measures . . . 13
  - 2.1.3 Pseudocode Conventions . . . 14
- 2.2 Broadcast and Convergecast on a Spanning Tree . . . 15
- 2.3 Flooding and Building a Spanning Tree . . . 20
- 2.4 Constructing a Depth-First Search Spanning Tree for a Specified Root . . . 24
- 2.5 Constructing a Depth-First Search Spanning Tree without a Specified Root . . . 26

3 Leader Election in Rings 31
- 3.1 The Leader Election Problem . . . 31
- 3.2 Anonymous Rings . . . 32
- 3.3 Asynchronous Rings . . . 34
  - 3.3.1 An $O(n^2)$ Algorithm . . . 34

3.3.2 An $O(n \log n)$ Algorithm . . . 35
3.3.3 An $\Omega(n \log n)$ Lower Bound . . . 38
3.4 Synchronous Rings . . . 43
3.4.1 An $O(n)$ Upper Bound . . . 43
3.4.2 An $\Omega(n \log n)$ Lower Bound for Restricted Algorithms . . . 48

4 Mutual Exclusion in Shared Memory **61**
4.1 Formal Model for Shared Memory Systems . . . 62
4.1.1 Systems . . . 62
4.1.2 Complexity Measures . . . 64
4.1.3 Pseudocode Conventions . . . 64
4.2 The Mutual Exclusion Problem . . . 65
4.3 Mutual Exclusion Using Powerful Primitives . . . 67
4.3.1 Binary Test&Set Registers . . . 67
4.3.2 Read-Modify-Write Registers . . . 68
4.3.3 Lower Bound on the Number of Memory States . . . 70
4.4 Mutual Exclusion Using Read/Write Registers . . . 72
4.4.1 The Bakery Algorithm . . . 72
4.4.2 A Bounded Algorithm for Two Processors . . . 74
4.4.3 A Bounded Algorithm for n Processors . . . 78
4.4.4 Lower Bound on the Number of Read/Write Registers . . . 81
4.4.5 Fast Mutual Exclusion . . . 86

5 Fault-Tolerant Consensus **91**
5.1 Synchronous Systems with Crash Failures . . . 92
5.1.1 Formal Model . . . 92
5.1.2 The Consensus Problem . . . 93
5.1.3 A Simple Algorithm . . . 93
5.1.4 Lower Bound on the Number of Rounds . . . 95
5.2 Synchronous Systems with Byzantine Failures . . . 102
5.2.1 Formal Model . . . 102
5.2.2 The Consensus Problem Revisited . . . 102
5.2.3 Lower Bound on the Ratio of Faulty Processors . . . 103
5.2.4 An Exponential Algorithm . . . 105
5.2.5 A Polynomial Algorithm . . . 109
5.3 Impossibility in Asynchronous Systems . . . 111
5.3.1 Shared Memory—The Wait-Free Case . . . 112
5.3.2 Shared Memory—The General Case . . . 115
5.3.3 Message Passing . . . 123

6 Causality and Time **129**
6.1 Capturing Causality . . . 129
6.1.1 The Happens-Before Relation . . . 130
6.1.2 Logical Clocks . . . 132
6.1.3 Vector Clocks . . . 133
6.1.4 Shared Memory Systems . . . 138
6.2 Examples of Using Causality . . . 138
6.2.1 Consistent Cuts . . . 138
6.2.2 A Limitation of the Happens-Before Relation: The Session Problem . . . 142
6.3 Clock Synchronization . . . 145
6.3.1 Modeling Physical Clocks . . . 145
6.3.2 The Clock Synchronization Problem . . . 148
6.3.3 The Two Processors Case . . . 150
6.3.4 An Upper Bound . . . 152
6.3.5 A Lower Bound . . . 153

Part II Simulations **159**

7 A Formal Model for Simulations **161**
7.1 Problem Specifications . . . 161
7.2 Communication Systems . . . 162
7.2.1 Asynchronous Point-to-Point Message Passing . . . 163
7.2.2 Asynchronous Broadcast . . . 163
7.3 Processes . . . 164
7.4 Admissibility . . . 167
7.5 Simulations . . . 168
7.6 Pseudocode Conventions . . . 169

8 Broadcast and Multicast **171**
8.1 Specification of Broadcast Services . . . 172
8.1.1 The Basic Service Specification . . . 172
8.1.2 Broadcast Service Qualities . . . 173
8.2 Implementing a Broadcast Service . . . 176
8.2.1 Basic Broadcast Service . . . 176
8.2.2 Single-Source FIFO Ordering . . . 176
8.2.3 Totally Ordered Broadcast . . . 177
8.2.4 Causality . . . 180
8.2.5 Reliability . . . 182
8.3 Multicast in Groups . . . 184
8.4 An Application: Replication . . . 188

9 Distributed Shared Memory **195**
9.1 Linearizable Shared Memory 196
9.2 Sequentially Consistent Shared Memory 198
9.3 Algorithms 198
9.3.1 Linearizability 199
9.3.2 Sequential Consistency 200
9.4 Lower Bounds 204
9.4.1 Adding Time and Clocks to the Layered Model 204
9.4.2 Sequential Consistency 205
9.4.3 Linearizability 206

10 Fault-Tolerant Simulations of Read/Write Objects **213**
10.1 Fault-Tolerant Shared Memory Simulations 214
10.2 Simple Read/Write Register Simulations 216
10.2.1 Multi-Valued from Binary 216
10.2.2 Multi-Reader from Single-Reader 220
10.2.3 Multi-Writer from Single-Writer 225
10.3 Atomic Snapshot Objects 228
10.3.1 Handshaking Procedures 229
10.3.2 A Bounded Memory Simulation 231
10.4 Simulating Shared Registers in Message-Passing Systems 235

11 Simulating Synchrony **245**
11.1 Synchronous Message Passing Specification 246
11.2 Simulating Synchronous Processors 247
11.3 Simulating Synchronous Processors and Synchronous Communication 249
11.3.1 A Simple Synchronizer 250
11.3.2 Application: Constructing a Breadth-First Search Tree . 253
11.4 Local vs. Global Simulations 254

12 Improving the Fault-Tolerance of Algorithms **257**
12.1 Overview 257
12.2 Modeling Synchronous Processors and Byzantine Failures 259
12.3 Simulating Identical Byzantine Failures on Top of Byzantine Failures 261
12.3.1 Definition of Identical Byzantine 261
12.3.2 Simulating Identical Byzantine 262
12.4 Simulating Omission Failures on Top of Identical Byzantine Failures 265
12.4.1 Definition of Omission 265
12.4.2 Simulating Omission 266
12.5 Simulating Crash Failures on Top of Omission Failures 271

12.5.1 Definition of Crash . . . 271
12.5.2 Simulating Crash . . . 272
12.6 Application: Consensus in the Presence of Byzantine Failures . . 275
12.7 Asynchronous Identical Byzantine on Top of Byzantine Failures . 276
12.7.1 Definition of Asynchronous Identical Byzantine . . . 277
12.7.2 Definition of Asynchronous Byzantine . . . 277
12.7.3 Simulating Asynchronous Identical Byzantine . . . 278

13 Fault-Tolerant Clock Synchronization 283
13.1 Problem Definition . . . 283
13.2 The Ratio of Faulty Processors . . . 285
13.3 A Clock Synchronization Algorithm . . . 290
13.3.1 Timing Failures . . . 291
13.3.2 Byzantine Failures . . . 297

Part III Advanced Topics 301

14 Randomization 303
14.1 Leader Election: A Case Study . . . 303
14.1.1 Weakening the Problem Definition . . . 303
14.1.2 Synchronous One-Shot Algorithm . . . 305
14.1.3 Synchronous Iterated Algorithm and Expectation . . . 307
14.1.4 Asynchronous Systems and Adversaries . . . 308
14.1.5 Impossibility of Uniform Algorithms . . . 309
14.1.6 Summary of Probabilistic Definitions . . . 310
14.2 Mutual Exclusion with Small Shared Variables . . . 311
14.3 Consensus . . . 315
14.3.1 The General Algorithm Scheme . . . 316
14.3.2 A Common Coin with Constant Bias . . . 319
14.3.3 Tolerating Byzantine Failures . . . 324
14.3.4 Shared Memory Systems . . . 324

15 Wait-Free Simulations of Arbitrary Objects 329
15.1 Example: A FIFO Queue . . . 330
15.2 The Wait-Free Hierarchy . . . 334
15.3 Universality . . . 336
15.3.1 A Non-Blocking Simulation Using Compare&Swap . . . 336
15.3.2 A Non-Blocking Algorithm Using Consensus Objects . . . 338
15.3.3 A Wait-Free Algorithm Using Consensus Objects . . . 341
15.3.4 Bounding the Memory Requirements . . . 344
15.3.5 Handling Non-Determinism . . . 346
15.3.6 Employing Randomized Consensus . . . 347

16 Bounded Timestamps **351**
16.1 Single-Generator Timestamp System 351
16.1.1 Traceable Writing and Reading of a Single Register . . . 353
16.1.2 The Single-Generator Timestamp Algorithm 360
16.1.3 Reducing the Complexity 364
16.2 Application: A Bounded Simulation of Multi-Reader Registers . 365
16.3 Concurrent Timestamp System 366
16.3.1 A Multi-Reader Traceable Register 368
16.3.2 The Multiple-Generator Timestamp Algorithm 369
16.4 Application: A Bounded Simulation of Multi-Writer Registers . 373

17 Problems Solvable in Asynchronous Systems **377**
17.1 k-Set Consensus 377
17.2 Approximate Agreement 386
17.2.1 Known Input Range 387
17.2.2 Unknown Input Range 389
17.3 Renaming 391
17.3.1 The Wait-Free Case 392
17.3.2 The General Case 394
17.3.3 Long-lived Renaming 395
17.4 k-Exclusion and k-Assignment 397
17.4.1 An Algorithm for k-Exclusion 398
17.4.2 An Algorithm for k-Assignment 400

18 Sparse Network Covers **405**
18.1 Sparse Network Covers 405
18.2 Routing with Low Memory Overhead 409
18.2.1 The Problem 410
18.2.2 Overview of the Routing Algorithm 411
18.2.3 The Regional Routing Algorithm 411
18.3 Application to Synchronizers 413

Bibliography **417**

Index **437**