Contents

Preface xix

1 Introduction **1**
1.1 The Subject Matter 1
1.2 Our Viewpoint 4
1.3 Overview of Chapters 2–25 6
1.4 Bibliographic Notes 13
1.5 Notation 14

Part I Synchronous Network Algorithms **15**

2 Modelling I: Synchronous Network Model **17**
2.1 Synchronous Network Systems 17
2.2 Failures 19
2.3 Inputs and Outputs 20
2.4 Executions 20
2.5 Proof Methods 21
2.6 Complexity Measures 21
2.7 Randomization 22
2.8 Bibliographic Notes 23

3 Leader Election in a Synchronous Ring **25**
3.1 The Problem 25
3.2 Impossibility Result for Identical Processes 27
3.3 A Basic Algorithm 27
3.4 An Algorithm with $O(n \log n)$ Communication Complexity . . . 31
3.5 Non-Comparison-Based Algorithms 35
3.5.1 The *TimeSlice* Algorithm 35
3.5.2 The *VariableSpeeds* Algorithm 36
3.6 Lower Bound for Comparison-Based Algorithms 38

3.7 Lower Bound for Non-Comparison-Based Algorithms* 44
3.8 Bibliographic Notes 46
3.9 Exercises 47

4 Algorithms in General Synchronous Networks **51**
4.1 Leader Election in a General Network 52
4.1.1 The Problem 52
4.1.2 A Simple Flooding Algorithm 52
4.1.3 Reducing the Communication Complexity 54
4.2 Breadth-First Search 57
4.2.1 The Problem 57
4.2.2 A Basic Breadth-First Search Algorithm 58
4.2.3 Applications 60
4.3 Shortest Paths 61
4.4 Minimum Spanning Tree 63
4.4.1 The Problem 63
4.4.2 Basic Theory 64
4.4.3 The Algorithm 66
4.5 Maximal Independent Set 71
4.5.1 The Problem 71
4.5.2 A Randomized Algorithm 71
4.5.3 Analysis* 74
4.6 Bibliographic Notes 76
4.7 Exercises 77

5 Distributed Consensus with Link Failures **81**
5.1 The Coordinated Attack Problem—Deterministic Version 82
5.2 The Coordinated Attack Problem—Randomized Version 86
5.2.1 Formal Modelling 87
5.2.2 An Algorithm 88
5.2.3 A Lower Bound on Disagreement 93
5.3 Bibliographic Notes 95
5.4 Exercises 95

6 Distributed Consensus with Process Failures **99**
6.1 The Problem 100
6.2 Algorithms for Stopping Failures 102
6.2.1 A Basic Algorithm 103
6.2.2 Reducing the Communication 105
6.2.3 Exponential Information Gathering Algorithms 108

6.2.4 Byzantine Agreement with Authentication 115
6.3 Algorithms for Byzantine Failures 116
6.3.1 An Example 117
6.3.2 EIG Algorithm for Byzantine Agreement 119
6.3.3 General Byzantine Agreement Using Binary Byzantine Agreement 123
6.3.4 Reducing the Communication Cost 125
6.4 Number of Processes for Byzantine Agreement 129
6.5 Byzantine Agreement in General Graphs 135
6.6 Weak Byzantine Agreement 139
6.7 Number of Rounds with Stopping Failures 142
6.8 Bibliographic Notes 152
6.9 Exercises 153

7 More Consensus Problems **161**
7.1 k-Agreement 161
7.1.1 The Problem 162
7.1.2 An Algorithm 162
7.1.3 Lower Bound* 164
7.2 Approximate Agreement 177
7.3 The Commit Problem 182
7.3.1 The Problem 182
7.3.2 Two-Phase Commit 184
7.3.3 Three-Phase Commit 185
7.3.4 Lower Bound on the Number of Messages 189
7.4 Bibliographic Notes 192
7.5 Exercises 192

Part II Asynchronous Algorithms **197**

8 Modelling II: Asynchronous System Model **199**
8.1 I/O Automata 200
8.2 Operations on Automata 206
8.2.1 Composition 207
8.2.2 Hiding 212
8.3 Fairness 212
8.4 Inputs and Outputs for Problems 215
8.5 Properties and Proof Methods 216
8.5.1 Invariant Assertions 216

8.5.2 Trace Properties 216
8.5.3 Safety and Liveness Properties 218
8.5.4 Compositional Reasoning 221
8.5.5 Hierarchical Proofs 224
8.6 Complexity Measures 228
8.7 Indistinguishable Executions 229
8.8 Randomization 229
8.9 Bibliographic Notes 230
8.10 Exercises 231

Part IIA Asynchronous Shared Memory Algorithms 235

9 Modelling III: Asynchronous Shared Memory Model 237
9.1 Shared Memory Systems 237
9.2 Environment Model 241
9.3 Indistinguishable States 244
9.4 Shared Variable Types 244
9.5 Complexity Measures 250
9.6 Failures 251
9.7 Randomization 251
9.8 Bibliographic Notes 251
9.9 Exercises 252

10 Mutual Exclusion 255
10.1 Asynchronous Shared Memory Model 256
10.2 The Problem 259
10.3 Dijkstra's Mutual Exclusion Algorithm 265
10.3.1 The Algorithm 265
10.3.2 A Correctness Argument 269
10.3.3 An Assertional Proof of the Mutual Exclusion Condition . 272
10.3.4 Running Time 274
10.4 Stronger Conditions for Mutual Exclusion Algorithms 276
10.5 Lockout-Free Mutual Exclusion Algorithms 278
10.5.1 A Two-Process Algorithm 278
10.5.2 An n-Process Algorithm 283
10.5.3 Tournament Algorithm 289
10.6 An Algorithm Using Single-Writer Shared Registers 294
10.7 The Bakery Algorithm 296
10.8 Lower Bound on the Number of Registers 300

10.8.1 Basic Facts . . . 301
10.8.2 Single-Writer Shared Variables . . . 302
10.8.3 Multi-Writer Shared Variables . . . 302
10.9 Mutual Exclusion Using Read-Modify-Write Shared Variables . . 309
10.9.1 The Basic Problem . . . 310
10.9.2 Bounded Bypass . . . 311
10.9.3 Lockout-Freedom . . . 319
10.9.4 A Simulation Proof . . . 322
10.10 Bibliographic Notes . . . 326
10.11 Exercises . . . 327

11 Resource Allocation **335**
11.1 The Problem . . . 336
11.1.1 Explicit Resource Specifications and Exclusion Specifications . . . 336
11.1.2 Resource-Allocation Problem . . . 337
11.1.3 Dining Philosophers Problem . . . 339
11.1.4 Restricted Form of Solutions . . . 341
11.2 Nonexistence of Symmetric Dining Philosophers Algorithms . . . 341
11.3 Right-Left Dining Philosophers Algorithm . . . 344
11.3.1 Waiting Chains . . . 344
11.3.2 The Basic Algorithm . . . 346
11.3.3 A Generalization . . . 349
11.4 Randomized Dining Philosophers Algorithm* . . . 354
11.4.1 The Algorithm* . . . 354
11.4.2 Correctness* . . . 357
11.5 Bibliographic Notes . . . 367
11.6 Exercises . . . 367

12 Consensus **371**
12.1 The Problem . . . 372
12.2 Agreement Using Read/Write Shared Memory . . . 376
12.2.1 Restrictions . . . 376
12.2.2 Terminology . . . 376
12.2.3 Bivalent Initializations . . . 377
12.2.4 Impossibility for Wait-Free Termination . . . 378
12.2.5 Impossibility for Single-Failure Termination . . . 383
12.3 Agreement Using Read-Modify-Write Shared Memory . . . 387
12.4 Other Types of Shared Memory . . . 388
12.5 Computability in Asynchronous Shared Memory Systems* . . . 389

12.6 Bibliographic Notes 391
12.7 Exercises 392

13 Atomic Objects **397**
13.1 Definitions and Basic Results 398
13.1.1 Atomic Object Definition 398
13.1.2 A Canonical Wait-Free Atomic Object Automaton 408
13.1.3 Composition of Atomic Objects 411
13.1.4 Atomic Objects versus Shared Variables 411
13.1.5 A Sufficient Condition for Showing Atomicity 419
13.2 Implementing Read-Modify-Write Atomic Objects in Terms of Read/Write Variables 420
13.3 Atomic Snapshots of Shared Memory 421
13.3.1 The Problem 422
13.3.2 An Algorithm with Unbounded Variables 423
13.3.3 An Algorithm with Bounded Variables* 428
13.4 Read/Write Atomic Objects 434
13.4.1 The Problem 434
13.4.2 Another Lemma for Showing Atomicity 434
13.4.3 An Algorithm with Unbounded Variables 436
13.4.4 A Bounded Algorithm for Two Writers 440
13.4.5 An Algorithm Using Snapshots 447
13.5 Bibliographic Notes 449
13.6 Exercises 450

Part IIB Asynchronous Network Algorithms **455**

14 Modelling IV: Asynchronous Network Model **457**
14.1 Send/Receive Systems 457
14.1.1 Processes 458
14.1.2 Send/Receive Channels 458
14.1.3 Asynchronous Send/Receive Systems 464
14.1.4 Properties of Send/Receive Systems with Reliable FIFO Channels 464
14.1.5 Complexity Measures 466
14.2 Broadcast Systems 466
14.2.1 Processes 466
14.2.2 Broadcast Channel 467
14.2.3 Asynchronous Broadcast Systems 468

14.2.4 Properties of Broadcast Systems with Reliable Broadcast Channels . . . 468
14.2.5 Complexity Measures . . . 469
14.3 Multicast Systems . . . 469
14.3.1 Processes . . . 469
14.3.2 Multicast Channel . . . 470
14.3.3 Asynchronous Multicast Systems . . . 471
14.4 Bibliographic Notes . . . 471
14.5 Exercises . . . 471

15 Basic Asynchronous Network Algorithms 475
15.1 Leader Election in a Ring . . . 475
15.1.1 The *LCR* Algorithm . . . 476
15.1.2 The *HS* Algorithm . . . 482
15.1.3 The Peterson Leader-Election Algorithm . . . 482
15.1.4 A Lower Bound on Communication Complexity . . . 486
15.2 Leader Election in an Arbitrary Network . . . 495
15.3 Spanning Tree Construction, Broadcast and Convergecast . . . 496
15.4 Breadth-First Search and Shortest Paths . . . 501
15.5 Minimum Spanning Tree . . . 509
15.5.1 Problem Statement . . . 509
15.5.2 The Synchronous Algorithm: Review . . . 510
15.5.3 The *GHS* Algorithm: Outline . . . 511
15.5.4 In More Detail . . . 513
15.5.5 Specific Messages . . . 517
15.5.6 Complexity Analysis . . . 519
15.5.7 Proving Correctness for the *GHS* Algorithm . . . 521
15.5.8 A Simpler "Synchronous" Strategy . . . 522
15.5.9 Application to Leader Election . . . 523
15.6 Bibliographic Notes . . . 523
15.7 Exercises . . . 524

16 Synchronizers 531
16.1 The Problem . . . 532
16.2 The Local Synchronizer . . . 535
16.3 The Safe Synchronizer . . . 541
16.3.1 Front-End Automata . . . 542
16.3.2 Channel Automata . . . 544
16.3.3 The Safe Synchronizer . . . 544
16.3.4 Correctness . . . 545

16.4 Safe Synchronizer Implementations 546
16.4.1 Synchronizer *Alpha* 546
16.4.2 Synchronizer *Beta* 547
16.4.3 Synchronizer *Gamma* 548
16.5 Applications 553
16.5.1 Leader Election 553
16.5.2 Breadth-First Search 554
16.5.3 Shortest Paths 554
16.5.4 Broadcast and Acknowledgment 555
16.5.5 Maximal Independent Set 555
16.6 Lower Bound on Time 555
16.7 Bibliographic Notes 560
16.8 Exercises 560

17 Shared Memory versus Networks **565**
17.1 Transformations from the Shared Memory Model to the Network Model 566
17.1.1 The Problem 566
17.1.2 Strategies Assuming No Failures 567
17.1.3 An Algorithm Tolerating Process Failures 575
17.1.4 An Impossibility Result for $\frac{n}{2}$ Failures 580
17.2 Transformations from the Network Model to the Shared Memory Model 582
17.2.1 Send/Receive Systems 583
17.2.2 Broadcast Systems 585
17.2.3 Impossibility of Agreement in Asynchronous Networks . . 586
17.3 Bibliographic Notes 586
17.4 Exercises 587

18 Logical Time **591**
18.1 Logical Time for Asynchronous Networks 591
18.1.1 Send/Receive Systems 592
18.1.2 Broadcast Systems 594
18.2 Adding Logical Time to Asynchronous Algorithms 596
18.2.1 Advancing the Clock 597
18.2.2 Delaying Future Events 598
18.3 Applications 600
18.3.1 Banking System 600
18.3.2 Global Snapshots 604
18.3.3 Simulating a Single State Machine 606

18.4 Transforming Real-Time Algorithms to Logical-Time Algorithms* 610
18.5 Bibliographic Notes 612
18.6 Exercises 612

19 Global Snapshots and Stable Properties 617
19.1 Termination-Detection for Diffusing Algorithms 618
19.1.1 The Problem 618
19.1.2 The *DijkstraScholten* Algorithm 619
19.2 Consistent Global Snapshots 625
19.2.1 The Problem 625
19.2.2 The *ChandyLamport* Algorithm 627
19.2.3 Applications 632
19.3 Bibliographic Notes 636
19.4 Exercises 637

20 Network Resource Allocation 641
20.1 Mutual Exclusion 641
20.1.1 The Problem 641
20.1.2 Simulating Shared Memory 643
20.1.3 Circulating Token Algorithm 643
20.1.4 An Algorithm Based on Logical Time 646
20.1.5 Improvements to the *LogicalTimeME* Algorithm 649
20.2 General Resource Allocation 653
20.2.1 The Problem 653
20.2.2 Coloring Algorithm 654
20.2.3 Algorithms Based on Logical Time 655
20.2.4 Acyclic Digraph Algorithm 656
20.2.5 Drinking Philosophers* 658
20.3 Bibliographic Notes 665
20.4 Exercises 665

21 Asynchronous Networks with Process Failures 669
21.1 The Network Model 670
21.2 Impossibility of Agreement in the Presence of Faults 671
21.3 A Randomized Algorithm 672
21.4 Failure Detectors 677
21.5 k-Agreement 681
21.6 Approximate Agreement 682
21.7 Computability in Asynchronous Networks* 684
21.8 Bibliographic Notes 685
21.9 Exercises 686

22 Data Link Protocols **691**
22.1 The Problem 692
22.2 Stenning's Protocol 693
22.3 Alternating Bit Protocol 697
22.4 Bounded Tag Protocols Tolerating Reordering 703
22.4.1 Impossibility Result for Reordering and Duplication . . . 704
22.4.2 A Bounded Tag Protocol Tolerating Loss and Reordering 706
22.4.3 Nonexistence of Efficient Protocols Tolerating Loss and Reordering 712
22.5 Tolerating Crashes 715
22.5.1 A Simple Impossibility Result 716
22.5.2 A Harder Impossibility Result 718
22.5.3 A Practical Protocol 721
22.6 Bibliographic Notes 728
22.7 Exercises 729

Part III Partially Synchronous Algorithms **733**

23 Partially Synchronous System Models **735**
23.1 MMT Timed Automata 736
23.1.1 Basic Definitions 736
23.1.2 Operations 741
23.2 General Timed Automata 744
23.2.1 Basic Definitions 745
23.2.2 Transforming MMT Automata into General Timed Automata 751
23.2.3 Operations 754
23.3 Properties and Proof Methods 756
23.3.1 Invariant Assertions 757
23.3.2 Timed Trace Properties 759
23.3.3 Simulations 760
23.4 Modelling Shared Memory and Network Systems 768
23.4.1 Shared Memory Systems 768
23.4.2 Networks 768
23.5 Bibliographic Notes 769
23.6 Exercises 770

24 Mutual Exclusion with Partial Synchrony **773**
24.1 The Problem . . . 773
24.2 A Single-Register Algorithm . . . 774
24.3 Resilience to Timing Failures . . . 784
24.4 Impossibility Results . . . 788
24.4.1 A Lower Bound on the Time . . . 788
24.4.2 Impossibility Result for Eventual Time Bounds* . . . 789
24.5 Bibliographic Notes . . . 790
24.6 Exercises . . . 791

25 Consensus with Partial Synchrony **795**
25.1 The Problem . . . 795
25.2 A Failure Detector . . . 796
25.3 Basic Results . . . 798
25.3.1 Upper Bound . . . 798
25.3.2 Lower Bound . . . 801
25.4 An Efficient Algorithm . . . 803
25.4.1 The Algorithm . . . 803
25.4.2 Safety Properties . . . 805
25.4.3 Liveness and Complexity . . . 806
25.5 A Lower Bound Involving the Timing Uncertainty* . . . 810
25.6 Other Results* . . . 818
25.6.1 Synchronous Processes, Asynchronous Channels* . . . 818
25.6.2 Asynchronous Processes, Synchronous Channels* . . . 819
25.6.3 Eventual Time Bounds* . . . 819
25.7 Postscript . . . 823
25.8 Bibliographic Notes . . . 823
25.9 Exercises . . . 824

Bibliography **829**

Index **857**