

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 <i>TimeSlice</i> Algorithm	35
3.5.2 The <i>VariableSpeeds</i> Algorithm	36
3.6 Lower Bound for Comparison-Based Algorithms	38

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	467
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

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