## Part I  Distributed Graph Algorithms

1  **Basic Definitions and Network Traversal Algorithms**  
   1.1  Distributed Algorithms  
   1.1.1  Definition  
   1.1.2  An Introductory Example: Learning the Communication Graph  
   1.2  Parallel Traversal: Broadcast and Convergecast  
   1.2.1  Broadcast and Convergecast  
   1.2.2  A Flooding Algorithm  
   1.2.3  Broadcast/Convergecast Based on a Rooted Spanning Tree  
   1.2.4  Building a Spanning Tree  
   1.3  Breadth-First Spanning Tree  
   1.3.1  Breadth-First Spanning Tree Built Without Centralized Control  
   1.3.2  Breadth-First Spanning Tree Built with Centralized Control  
   1.4  Depth-First Traversal  
   1.4.1  A Simple Algorithm  
   1.4.2  Application: Construction of a Logical Ring  
   1.5  Summary  
   1.6  Bibliographic Notes  
   1.7  Exercises and Problems  

2  **Distributed Graph Algorithms**  
   2.1  Distributed Shortest Path Algorithms  
   2.1.1  A Distributed Adaptation of Bellman–Ford’s Shortest Path Algorithm  
   2.1.2  A Distributed Adaptation of Floyd–Warshall’s Shortest Paths Algorithm  
   2.2  Vertex Coloring and Maximal Independent Set  
   2.2.1  On Sequential Vertex Coloring
2.2.2 Distributed \((\Delta + 1)\)-Coloring of Processes .......... 43
2.2.3 Computing a Maximal Independent Set ............ 46
2.3 Knot and Cycle Detection .................................. 50
2.3.1 Directed Graph, Knot, and Cycle ................. 50
2.3.2 Communication Graph, Logical Directed Graph, and Reachability ................................ 51
2.3.3 Specification of the Knot Detection Problem ........ 51
2.3.4 Principle of the Knot/Cycle Detection Algorithm .... 52
2.3.5 Local Variables ................................ 53
2.3.6 Behavior of a Process ................................ 54
2.4 Summary ........................................... 57
2.5 Bibliographic Notes .................................. 58
2.6 Exercises and Problems ................................ 58

3 An Algorithmic Framework to Compute Global Functions on a Process Graph ............... 59
3.1 Distributed Computation of Global Functions ............ 59
3.1.1 Type of Global Functions .......................... 59
3.1.2 Constraints on the Computation ....................... 60
3.2 An Algorithmic Framework ................................ 61
3.2.1 A Round-Based Framework ............................. 61
3.2.2 When the Diameter Is Not Known ....................... 64
3.3 Distributed Determination of Cut Vertices ............... 66
3.3.1 Cut Vertices ........................................ 66
3.3.2 An Algorithm Determining Cut Vertices ............. 67
3.4 Improving the Framework ................................ 69
3.4.1 Two Types of Filtering ................................ 69
3.4.2 An Improved Algorithm ............................... 70
3.5 The Case of Regular Communication Graphs ............ 72
3.5.1 Tradeoff Between Graph Topology and Number of Rounds .... 72
3.5.2 De Bruijn Graphs ................................ 73
3.6 Summary ........................................... 75
3.7 Bibliographic Notes .................................. 76
3.8 Problem ........................................... 76

4 Leader Election Algorithms .................................. 77
4.1 The Leader Election Problem .............................. 77
4.1.1 Problem Definition ................................ 77
4.1.2 Anonymous Systems: An Impossibility Result ......... 78
4.1.3 Basic Assumptions and Principles of the Election Algorithms ................................ 79
4.2 A Simple \(O(n^2)\) Leader Election Algorithm for Unidirectional Rings .................. 79
4.2.1 Context and Principle ................................ 79
4.2.2 The Algorithm ..................................... 80
4.2.3 Time Cost of the Algorithm ......................... 80
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.4</td>
<td>Message Cost of the Algorithm</td>
<td>81</td>
</tr>
<tr>
<td>4.2.5</td>
<td>A Simple Variant</td>
<td>82</td>
</tr>
<tr>
<td>4.3</td>
<td>An $O(n \log n)$ Leader Election Algorithm for Bidirectional Rings</td>
<td>83</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Context and Principle</td>
<td>83</td>
</tr>
<tr>
<td>4.3.2</td>
<td>The Algorithm</td>
<td>84</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Time and Message Complexities</td>
<td>85</td>
</tr>
<tr>
<td>4.4</td>
<td>An $O(n \log n)$ Election Algorithm for Unidirectional Rings</td>
<td>86</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Context and Principles</td>
<td>86</td>
</tr>
<tr>
<td>4.4.2</td>
<td>The Algorithm</td>
<td>88</td>
</tr>
<tr>
<td>4.4.3</td>
<td>Discussion: Message Complexity and FIFO Channels</td>
<td>89</td>
</tr>
<tr>
<td>4.5</td>
<td>Two Particular Cases</td>
<td>89</td>
</tr>
<tr>
<td>4.6</td>
<td>Summary</td>
<td>90</td>
</tr>
<tr>
<td>4.7</td>
<td>Bibliographic Notes</td>
<td>90</td>
</tr>
<tr>
<td>4.8</td>
<td>Exercises and Problems</td>
<td>91</td>
</tr>
<tr>
<td>5</td>
<td>Mobile Objects Navigating a Network</td>
<td>93</td>
</tr>
<tr>
<td>5.1</td>
<td>Mobile Object in a Process Graph</td>
<td>93</td>
</tr>
<tr>
<td>5.1.1</td>
<td>Problem Definition</td>
<td>93</td>
</tr>
<tr>
<td>5.1.2</td>
<td>Mobile Object Versus Mutual Exclusion</td>
<td>94</td>
</tr>
<tr>
<td>5.1.3</td>
<td>A Centralized (Home-Based) Algorithm</td>
<td>94</td>
</tr>
<tr>
<td>5.1.4</td>
<td>The Algorithms Presented in This Chapter</td>
<td>95</td>
</tr>
<tr>
<td>5.2</td>
<td>A Navigation Algorithm for a Complete Network</td>
<td>96</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Underlying Principles</td>
<td>96</td>
</tr>
<tr>
<td>5.2.2</td>
<td>The Algorithm</td>
<td>97</td>
</tr>
<tr>
<td>5.3</td>
<td>A Navigation Algorithm Based on a Spanning Tree</td>
<td>100</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Principles of the Algorithm:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tree Invariant and Proxy Behavior</td>
<td>101</td>
</tr>
<tr>
<td>5.3.2</td>
<td>The Algorithm</td>
<td>102</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Discussion and Properties</td>
<td>104</td>
</tr>
<tr>
<td>5.3.4</td>
<td>Proof of the Algorithm</td>
<td>106</td>
</tr>
<tr>
<td>5.4</td>
<td>An Adaptive Navigation Algorithm</td>
<td>108</td>
</tr>
<tr>
<td>5.4.1</td>
<td>The Adaptivity Property</td>
<td>109</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Principle of the Implementation</td>
<td>109</td>
</tr>
<tr>
<td>5.4.3</td>
<td>An Adaptive Algorithm Based on a Distributed Queue</td>
<td>111</td>
</tr>
<tr>
<td>5.4.4</td>
<td>Properties</td>
<td>113</td>
</tr>
<tr>
<td>5.4.5</td>
<td>Example of an Execution</td>
<td>114</td>
</tr>
<tr>
<td>5.5</td>
<td>Summary</td>
<td>115</td>
</tr>
<tr>
<td>5.6</td>
<td>Bibliographic Notes</td>
<td>115</td>
</tr>
<tr>
<td>5.7</td>
<td>Exercises and Problems</td>
<td>116</td>
</tr>
</tbody>
</table>

Part II Logical Time and Global States in Distributed Systems

6 Nature of Distributed Computations and the Concept of a Global State

6.1 A Distributed Execution Is a Partial Order on Local Events

6.1.1 Basic Definitions
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1.2</td>
<td>A Distributed Execution Is a Partial Order on Local Events</td>
<td>122</td>
</tr>
<tr>
<td>6.1.3</td>
<td>Causal Past, Causal Future, Concurrency, Cut</td>
<td>123</td>
</tr>
<tr>
<td>6.1.4</td>
<td>Asynchronous Distributed Execution with Respect to Physical Time</td>
<td>125</td>
</tr>
<tr>
<td>6.2</td>
<td>A Distributed Execution Is a Partial Order on Local States</td>
<td>127</td>
</tr>
<tr>
<td>6.3</td>
<td>Global State and Lattice of Global States</td>
<td>129</td>
</tr>
<tr>
<td>6.3.1</td>
<td>The Concept of a Global State</td>
<td>129</td>
</tr>
<tr>
<td>6.3.2</td>
<td>Lattice of Global States</td>
<td>129</td>
</tr>
<tr>
<td>6.3.3</td>
<td>Sequential Observations</td>
<td>131</td>
</tr>
<tr>
<td>6.4</td>
<td>Global States Including Process States and Channel States</td>
<td>132</td>
</tr>
<tr>
<td>6.4.1</td>
<td>Global State Including Channel States</td>
<td>132</td>
</tr>
<tr>
<td>6.4.2</td>
<td>Consistent Global State Including Channel States</td>
<td>133</td>
</tr>
<tr>
<td>6.4.3</td>
<td>Consistent Global State Versus Consistent Cut</td>
<td>134</td>
</tr>
<tr>
<td>6.5</td>
<td>On-the-Fly Computation of Global States</td>
<td>135</td>
</tr>
<tr>
<td>6.5.1</td>
<td>Global State Computation Is an Observation Problem</td>
<td>135</td>
</tr>
<tr>
<td>6.5.2</td>
<td>Problem Definition</td>
<td>136</td>
</tr>
<tr>
<td>6.5.3</td>
<td>On the Meaning of the Computed Global State</td>
<td>136</td>
</tr>
<tr>
<td>6.5.4</td>
<td>Principles of Algorithms Computing a Global State</td>
<td>137</td>
</tr>
<tr>
<td>6.6</td>
<td>A Global State Algorithm Suited to FIFO Channels</td>
<td>138</td>
</tr>
<tr>
<td>6.6.1</td>
<td>Principle of the Algorithm</td>
<td>138</td>
</tr>
<tr>
<td>6.6.2</td>
<td>The Algorithm</td>
<td>140</td>
</tr>
<tr>
<td>6.6.3</td>
<td>Example of an Execution</td>
<td>141</td>
</tr>
<tr>
<td>6.7</td>
<td>A Global State Algorithm Suited to Non-FIFO Channels</td>
<td>143</td>
</tr>
<tr>
<td>6.7.1</td>
<td>The Algorithm and Its Principles</td>
<td>144</td>
</tr>
<tr>
<td>6.7.2</td>
<td>How to Compute the State of the Channels</td>
<td>144</td>
</tr>
<tr>
<td>6.8</td>
<td>Summary</td>
<td>146</td>
</tr>
<tr>
<td>6.9</td>
<td>Bibliographic Notes</td>
<td>146</td>
</tr>
<tr>
<td>6.10</td>
<td>Exercises and Problems</td>
<td>147</td>
</tr>
<tr>
<td>7</td>
<td>Logical Time in Asynchronous Distributed Systems</td>
<td>149</td>
</tr>
<tr>
<td>7.1</td>
<td>Linear Time</td>
<td>149</td>
</tr>
<tr>
<td>7.1.1</td>
<td>Scalar (or Linear) Time</td>
<td>150</td>
</tr>
<tr>
<td>7.1.2</td>
<td>From Partial Order to Total Order: The Notion of a Timestamp</td>
<td>151</td>
</tr>
<tr>
<td>7.1.3</td>
<td>Relating Logical Time and Timestamps with Observations</td>
<td>152</td>
</tr>
<tr>
<td>7.1.4</td>
<td>Timestamps in Action: Total Order Broadcast</td>
<td>153</td>
</tr>
<tr>
<td>7.2</td>
<td>Vector Time</td>
<td>159</td>
</tr>
<tr>
<td>7.2.1</td>
<td>Vector Time and Vector Clocks</td>
<td>159</td>
</tr>
<tr>
<td>7.2.2</td>
<td>Vector Clock Properties</td>
<td>162</td>
</tr>
<tr>
<td>7.2.3</td>
<td>On the Development of Vector Time</td>
<td>163</td>
</tr>
<tr>
<td>7.2.4</td>
<td>Relating Vector Time and Global States</td>
<td>165</td>
</tr>
<tr>
<td>7.2.5</td>
<td>Vector Clocks in Action: On-the-Fly Determination of a Global State Property</td>
<td>166</td>
</tr>
<tr>
<td>7.2.6</td>
<td>Vector Clocks in Action: On-the-Fly Determination of the Immediate Predecessors</td>
<td>170</td>
</tr>
<tr>
<td>7.3</td>
<td>On the Size of Vector Clocks</td>
<td>173</td>
</tr>
</tbody>
</table>
7.3.1 A Lower Bound on the Size of Vector Clocks ........ 174
7.3.2 An Efficient Implementation of Vector Clocks .......... 176
7.3.3 k-Restricted Vector Clock ................................ 181

7.4 Matrix Time .............................................. 182
7.4.1 Matrix Clock: Definition and Algorithm ............ 182
7.4.2 A Variant of Matrix Time in Action: Discard Old Data .. 184

7.5 Summary ............................................... 186
7.6 Bibliographic Notes ..................................... 186
7.7 Exercises and Problems .................................. 187

8 Asynchronous Distributed Checkpointing ...................... 189
8.1 Definitions and Main Theorem ............................ 189
8.1.1 Local and Global Checkpoints ...................... 189
8.1.2 Z-Dependency, Zigzag Paths, and Z-Cycles .......... 190
8.1.3 The Main Theorem .................................. 192
8.2 Consistent Checkpointing Abstractions .................. 196
8.2.1 Z-Cycle-Freedom .................................. 196
8.2.2 Rollback-Dependency Trackability .................. 197
8.2.3 On Distributed Checkpointing Algorithms .......... 198
8.3 Checkpointing Algorithms Ensuring Z-Cycle Prevention .... 199
8.3.1 An Operational Characterization of Z-Cycle-Freedom .. 199
8.3.2 A Property of a Particular Dating System .......... 199
8.3.3 Two Simple Algorithms Ensuring Z-Cycle Prevention .. 201
8.3.4 On the Notion of an Optimal Algorithm for Z-Cycle Prevention ........................................... 203
8.4 Checkpointing Algorithms Ensuring Rollback-Dependency Trackability .............................................. 203
8.4.1 Rollback-Dependency Trackability (RDT) ............ 203
8.4.2 A Simple Brute Force RDT Checkpointing Algorithm .. 205
8.4.3 The Fixed Dependency After Send (FDAS) RDT Checkpointing Algorithm ...................................... 206
8.4.4 Still Reducing the Number of Forced Local Checkpoints ........................................ 207
8.5 Message Logging for Uncoordinated Checkpointing ........ 211
8.5.1 Uncoordinated Checkpointing ........................ 211
8.5.2 To Log or Not to Log Messages on Stable Storage ... 211
8.5.3 A Recovery Algorithm ................................ 214
8.5.4 A Few Improvements .................................. 215
8.6 Summary ............................................... 216
8.7 Bibliographic Notes ..................................... 216
8.8 Exercises and Problems .................................. 217

9 Simulating Synchrony on Top of Asynchronous Systems .... 219
9.1 Synchronous Systems, Asynchronous Systems, and Synchronizers 219
9.1.1 Synchronous Systems ............................... 219
9.1.2 Asynchronous Systems and Synchronizers .......... 221
9.1.3 On the Efficiency Side ................................ 222
9.2 Basic Principle for a Synchronizer .................................. 223
  9.2.1 The Main Problem to Solve .................................. 223
  9.2.2 Principle of the Solutions ................................. 224
9.3 Basic Synchronizers: \( \alpha \) and \( \beta \) ......................... 224
  9.3.1 Synchronizer \( \alpha \) .................................. 224
  9.3.2 Synchronizer \( \beta \) .................................. 227
9.4 Advanced Synchronizers: \( \gamma \) and \( \delta \) ..................... 230
  9.4.1 Synchronizer \( \gamma \) .................................. 230
  9.4.2 Synchronizer \( \delta \) .................................. 234
9.5 The Case of Networks with Bounded Delays ..................... 236
  9.5.1 Context and Hypotheses .................................. 236
  9.5.2 The Problem to Solve .................................. 237
  9.5.3 Synchronizer \( \lambda \) .................................. 238
  9.5.4 Synchronizer \( \mu \) .................................. 239
  9.5.5 When the Local Physical Clocks Drift ...................... 240
9.6 Summary .......................................................... 242
9.7 Bibliographic Notes ................................................ 243
9.8 Exercises and Problems ............................................. 244

Part III Mutual Exclusion and Resource Allocation

10 Permission-Based Mutual Exclusion Algorithms .................. 247
  10.1 The Mutual Exclusion Problem .............................. 247
    10.1.1 Definition ............................................. 247
    10.1.2 Classes of Distributed Mutex Algorithms .......... 248
  10.2 A Simple Algorithm Based on Individual Permissions ....... 249
    10.2.1 Principle of the Algorithm .......................... 249
    10.2.2 The Algorithm ........................................ 251
    10.2.3 Proof of the Algorithm ................................ 252
    10.2.4 From Simple Mutex to Mutex on Classes of Operations .. 255
  10.3 Adaptive Mutex Algorithms Based on Individual Permissions ... 256
    10.3.1 The Notion of an Adaptive Algorithm ................. 256
    10.3.2 A Timestamp-Based Adaptive Algorithm ................ 257
    10.3.3 A Bounded Adaptive Algorithm ........................ 259
    10.3.4 Proof of the Bounded Adaptive Mutex Algorithm ...... 262
  10.4 An Algorithm Based on Arbiter Permissions .................... 264
    10.4.1 Permissions Managed by Arbiters ...................... 264
    10.4.2 Permissions Versus Quorums .......................... 265
    10.4.3 Quorum Construction .................................. 266
    10.4.4 An Adaptive Mutex Algorithm Based on Arbiter Permissions ... 268
  10.5 Summary .......................................................... 273
  10.6 Bibliographic Notes .............................................. 273
  10.7 Exercises and Problems ........................................... 274
11 Distributed Resource Allocation .......................... 277
  11.1 A Single Resource with Several Instances ............... 277
      11.1.1 The k-out-of-M Problem .......................... 277
      11.1.2 Mutual Exclusion with Multiple Entries:
            The 1-out-of-M Mutex Problem ....................... 278
      11.1.3 An Algorithm for the k-out-of-M Mutex Problem .... 280
      11.1.4 Proof of the Algorithm .......................... 283
      11.1.5 From Mutex Algorithms to k-out-of-M Algorithms ... 285
  11.2 Several Resources with a Single Instance .................. 285
      11.2.1 Several Resources with a Single Instance .......... 286
      11.2.2 Incremental Requests for Single Instance Resources:
            Using a Total Order .............................. 287
      11.2.3 Incremental Requests for Single Instance Resources:
            Reducing Process Waiting Chains .................. 290
      11.2.4 Simultaneous Requests for Single Instance Resources
            and Static Sessions .............................. 292
      11.2.5 Simultaneous Requests for Single Instance Resources
            and Dynamic Sessions ........................... 293
  11.3 Several Resources with Multiple Instances .................. 295
  11.4 Summary ............................................. 297
  11.5 Bibliographic Notes .................................... 298
  11.6 Exercises and Problems .................................. 299

Part IV High-Level Communication Abstractions

12 Order Constraints on Message Delivery ..................... 303
  12.1 The Causal Message Delivery Abstraction .................. 303
      12.1.1 Definition of Causal Message Delivery .............. 304
      12.1.2 A Causality-Based Characterization
            of Causal Message Delivery ...................... 305
      12.1.3 Causal Order
            with Respect to Other Message Ordering Constraints ... 306
  12.2 A Basic Algorithm for Point-to-Point Causal Message Delivery ........ 306
      12.2.1 A Simple Algorithm .............................. 306
      12.2.2 Proof of the Algorithm .......................... 309
      12.2.3 Reduce the Size of Control Information
            Carried by Messages ............................. 310
  12.3 Causal Broadcast ........................................ 313
      12.3.1 Definition and a Simple Algorithm ................. 313
      12.3.2 The Notion of a Causal Barrier .................... 315
      12.3.3 Causal Broadcast with Bounded Lifetime Messages ... 317
  12.4 The Total Order Broadcast Abstraction ..................... 320
      12.4.1 Strong Total Order Versus Weak Total Order ........ 320
      12.4.2 An Algorithm Based on a Coordinator Process
            or a Circulating Token ........................... 322
12.3 An Inquiry-Based Algorithm ........................................ 324
12.4 An Algorithm for Synchronous Systems ....................... 326
12.5 Playing with a Single Channel .................................... 328
   12.5.1 Four Order Properties on a Channel ...................... 328
   12.5.2 A General Algorithm Implementing These Properties .... 329
12.6 Summary ........................................................................ 332
12.7 Bibliographic Notes .................................................... 332
12.8 Exercises and Problems .............................................. 333

13 Rendezvous (Synchronous) Communication ....................... 335
   13.1 The Synchronous Communication Abstraction ............... 335
      13.1.1 Definition .................................................. 335
      13.1.2 An Example of Use ........................................ 337
      13.1.3 A Message Pattern-Based Characterization ............. 338
      13.1.4 Types of Algorithms Implementing Synchronous Communications .................................................. 341
   13.2 Algorithms for Nondeterministic Planned Interactions .... 341
      13.2.1 Deterministic and Nondeterministic Communication Contexts .................................................. 341
      13.2.2 An Asymmetric (Static) Client–Server Implementation . 342
      13.2.3 An Asymmetric Token-Based Implementation ........... 345
   13.3 An Algorithm for Nondeterministic Forced Interactions .... 350
      13.3.1 Nondeterministic Forced Interactions .................. 350
      13.3.2 A Simple Algorithm ........................................ 350
      13.3.3 Proof of the Algorithm .................................... 352
   13.4 Rendezvous with Deadlines in Synchronous Systems ....... 354
      13.4.1 Synchronous Systems and Rendezvous with Deadline . 354
      13.4.2 Rendezvous with Deadline Between Two Processes .... 355
      13.4.3 Introducing Nondeterministic Choice .................... 358
      13.4.4 \( n \)-Way Rendezvous with Deadline ............... 360
   13.5 Summary ..................................................................... 361
   13.6 Bibliographic Notes ................................................ 361
   13.7 Exercises and Problems .......................................... 362

Part V Detection of Properties on Distributed Executions

14 Distributed Termination Detection ................................. 367
   14.1 The Distributed Termination Detection Problem ............ 367
      14.1.1 Process and Channel States ............................. 367
      14.1.2 Termination Predicate .................................... 368
      14.1.3 The Termination Detection Problem .................... 369
      14.1.4 Types and Structure of Termination Detection Algorithms 369
   14.2 Termination Detection in the Asynchronous Atomic Model . 370
      14.2.1 The Atomic Model ......................................... 370
14.2.2 The Four-Counter Algorithm ............................. 371
14.2.3 The Counting Vector Algorithm .......................... 373
14.2.4 The Four-Counter Algorithm vs. the Counting Vector Algorithm ............................. 376
14.3 Termination Detection in Diffusing Computations ............................. 376
  14.3.1 The Notion of a Diffusing Computation .................. 376
  14.3.2 A Detection Algorithm Suited to Diffusing Computations ............................. 377
14.4 A General Termination Detection Algorithm ............................. 378
  14.4.1 Wave and Sequence of Waves ............................ 379
  14.4.2 A Reasoned Construction ................................ 381
14.5 Termination Detection in a Very General Distributed Model ............................. 385
  14.5.1 Model and Nondeterministic Atomic Receive Statement ............................. 385
  14.5.2 The Predicate fulfilled() ................................ 387
  14.5.3 Static vs. Dynamic Termination: Definition ............................. 388
  14.5.4 Detection of Static Termination ................................ 390
  14.5.5 Detection of Dynamic Termination ............................. 393
14.6 Summary ......................................................... 396
14.7 Bibliographic Notes .................................................. 396
14.8 Exercises and Problems .................................................. 397

15 Distributed Deadlock Detection .................................................. 401
  15.1 The Deadlock Detection Problem .................................................. 401
    15.1.1 Wait-For Graph (WFG) .................................................. 401
    15.1.2 AND and OR Models Associated with Deadlock .................................................. 403
    15.1.3 Deadlock in the AND Model .................................................. 403
    15.1.4 Deadlock in the OR Model .................................................. 404
    15.1.5 The Deadlock Detection Problem .................................................. 404
    15.1.6 Structure of Deadlock Detection Algorithms .................................................. 405
  15.2 Deadlock Detection in the One-at-a-Time Model .................................................. 405
    15.2.1 Principle and Local Variables .................................................. 406
    15.2.2 A Detection Algorithm .................................................. 406
    15.2.3 Proof of the Algorithm .................................................. 407
  15.3 Deadlock Detection in the AND Communication Model .................................................. 408
    15.3.1 Model and Principle of the Algorithm .................................................. 409
    15.3.2 A Detection Algorithm .................................................. 409
    15.3.3 Proof of the Algorithm .................................................. 411
  15.4 Deadlock Detection in the OR Communication Model .................................................. 413
    15.4.1 Principle .................................................. 413
    15.4.2 A Detection Algorithm .................................................. 416
    15.4.3 Proof of the Algorithm .................................................. 419
  15.5 Summary .................................................. 421
  15.6 Bibliographic Notes .................................................. 421
  15.7 Exercises and Problems .................................................. 422
Part VI  Distributed Shared Memory

16  Atomic Consistency (Linearizability) ................................................. 427
  16.1 The Concept of a Distributed Shared Memory ............................. 427
  16.2 The Atomicity Consistency Condition ................................. 429
     16.2.1 What Is the Issue? .................................................... 429
     16.2.2 An Execution Is a Partial Order on Operations ................. 429
     16.2.3 Atomicity: Formal Definition ................................... 430
  16.3 Atomic Objects Compose for Free .......................................... 432
  16.4 Message-Passing Implementations of Atomicity ...................... 435
     16.4.1 Atomicity Based on a Total Order Broadcast Abstraction .... 435
     16.4.2 Atomicity of Read/Write Objects Based on Server Processes ... 437
     16.4.3 Atomicity Based on a Server Process and Copy Invalidation ... 438
     16.4.4 Introducing the Notion of an Owner Process .................... 439
     16.4.5 Atomicity Based on a Server Process and Copy Update .... 443
  16.5 Summary ............................................................................. 444
  16.6 Bibliographic Notes ............................................................ 444
  16.7 Exercises and Problems ....................................................... 445

17  Sequential Consistency ................................................................. 447
  17.1 Sequential Consistency .......................................................... 447
     17.1.1 Definition ................................................................. 447
     17.1.2 Sequential Consistency Is Not a Local Property ............... 449
     17.1.3 Partial Order for Sequential Consistency ....................... 450
     17.1.4 Two Theorems for Sequentially Consistent Read/Write Registers .... 451
     17.1.5 From Theorems to Algorithms .................................... 453
  17.2 Sequential Consistency from Total Order Broadcast ................. 453
     17.2.1 A Fast Read Algorithm for Read/Write Objects .............. 453
     17.2.2 A Fast Write Algorithm for Read/Write Objects .......... 455
     17.2.3 A Fast Enqueue Algorithm for Queue Objects .......... 456
  17.3 Sequential Consistency from a Single Server ......................... 456
     17.3.1 The Single Server Is a Process .................................. 456
     17.3.2 The Single Server Is a Navigating Token .................... 459
  17.4 Sequential Consistency with a Server per Object .................... 460
     17.4.1 Structural View ....................................................... 460
     17.4.2 The Object Managers Must Cooperate ....................... 461
     17.4.3 An Algorithm Based on the OO Constraint ..................... 462
  17.5 A Weaker Consistency Condition: Causal Consistency ............... 464
     17.5.1 Definition ............................................................. 464
     17.5.2 A Simple Algorithm ................................................ 466
     17.5.3 The Case of a Single Object .................................... 467
  17.6 A Hierarchy of Consistency Conditions ............................... 468
Distributed Algorithms for Message-Passing Systems
Raynal, M.
2013, XXXI, 500 p. 198 illus., 106 illus. in color., Hardcover
ISBN: 978-3-642-38122-5