REAL-TIME SYSTEMS
To My Family and Friends
CONTENTS

PREFACE xiii

LIST OF FIGURES xix

1 INTRODUCTION 1
1.1 What Is Time? / 3
1.2 Simulation / 4
1.3 Testing / 5
1.4 Verification / 6
1.5 Run-Time Monitoring / 7
1.6 Useful Resources / 8

2 ANALYSIS AND VERIFICATION OF NON-REAL-TIME SYSTEMS 10
2.1 Symbolic Logic / 10
2.2 Automata and Languages / 28
2.3 Historical Perspective and Related Work / 37
2.4 Summary / 38
Exercises / 39

3 REAL-TIME SCHEDULING AND SCHEDULABILITY ANALYSIS 41
3.1 Determining Computation Time / 43
3.2 Uniprocessor Scheduling / 44
3.3 Multiprocessor Scheduling / 66  
3.4 Available Scheduling Tools / 72  
3.5 Available Real-Time Operating Systems / 75  
3.6 Historical Perspective and Related Work / 76  
3.7 Summary / 77  
Exercises / 84

4 MODEL CHECKING OF FINITE-STATE SYSTEMS  
4.1 System Specification / 87  
4.2 Clarke–Emerson–Sistla Model Checker / 89  
4.3 Extensions to CTL / 93  
4.4 Applications / 93  
4.5 Complete CTL Model Checker in C / 96  
4.6 Symbolic Model Checking / 116  
4.7 Real-Time CTL / 120  
4.8 Available Tools / 126  
4.9 Historical Perspective and Related Work / 127  
4.10 Summary / 129  
Exercises / 131

5 VISUAL FORMALISM, STATECHARTS, AND STATEMATE  
5.1 Statecharts / 135  
5.2 Activity-Charts / 140  
5.3 Module-Charts / 140  
5.4 STATEMATE / 142  
5.5 Available Tools / 143  
5.6 Historical Perspective and Related Work / 145  
5.7 Summary / 146  
Exercises / 147

6 REAL-TIME LOGIC, GRAPH-THEORETIC ANALYSIS, 
AND MODECHART  
6.1 Specification and Safety Assertions / 149  
6.2 Event-Action Model / 149  
6.3 Real-Time Logic / 150  
6.4 Restricted RTL Formulas / 152  
6.5 Checking for Unsatisfiability / 155  
6.6 Efficient Unsatisfiability Check / 157
6.7 Industrial Example: NASA X-38 Crew Return Vehicle / 161
6.8 Modechart Specification Language / 172
6.9 Verifying Timing Properties of Modechart Specifications / 175
6.10 Available Tools / 180
6.11 Historical Perspective and Related Work / 180
6.12 Summary / 181
Exercises / 183

7 VERIFICATION USING TIMED AUTOMATA / 187
7.1 Lynch–Vaandrager Automata-Theoretic Approach / 187
7.2 Alur–Dill Automata-Theoretic Approach / 193
7.3 Alur–Dill Region Automaton and Verification / 201
7.4 Available Tools / 205
7.5 Historical Perspective and Related Work / 207
7.6 Summary / 207
Exercises / 210

8 TIMED PETRI NETS / 212
8.1 Untimed Petri Nets / 212
8.2 Petri Nets with Time Extensions / 214
8.3 Time ER Nets / 220
8.4 Properties of High-Level Petri Nets / 224
8.5 Berthomieu–Diaz Analysis Algorithm for TPNs / 226
8.6 Milano Group’s Approach to HLTPN Analysis / 229
8.7 Practicality: Available Tools / 231
8.8 Historical Perspective and Related Work / 232
8.9 Summary / 233
Exercises / 236

9 PROCESS ALGEBRA / 237
9.1 Untimed Process Algebras / 237
9.2 Milner’s Calculus of Communicating Systems / 238
9.3 Timed Process Algebras / 241
9.4 Algebra of Communicating Shared Resources / 242
9.5 Analysis and Verification / 250
9.6 Relationships to Other Approaches / 255
9.7 Available Tools / 255
9.8 Historical Perspective and Related Work / 256
10 DESIGN AND ANALYSIS OF PROPOSITIONAL-LOGIC RULE-BASED SYSTEMS

10.1 Real-Time Decision Systems / 260
10.2 Real-Time Expert Systems / 262
10.3 Propositional-Logic Rule-Based Programs: the EQL Language / 263
10.4 State-Space Representation / 269
10.5 Computer-Aided Design Tools / 272
10.6 The Analysis Problem / 280
10.7 Industrial Example: Analysis of the Cryogenic Hydrogen Pressure Malfunction Procedure of the Space Shuttle Vehicle Pressure Control System / 286
10.8 The Synthesis Problem / 294
10.9 Specifying Termination Conditions in Estella / 301
10.10 Two Industrial Examples / 317
10.11 The Estella-General Analysis Tool / 324
10.12 Quantitative Timing Analysis Algorithms / 333
10.13 Historical Perspective and Related Work / 360
10.14 Summary / 363
Exercises / 365

11 TIMING ANALYSIS OF PREDICATE-LOGIC RULE-BASED SYSTEMS

11.1 The OPS5 Language / 369
11.2 Cheng–Tsai Timing Analysis Methodology / 373
11.3 Cheng–Chen Timing Analysis Methodology / 399
11.4 Historical Perspective and Related Work / 430
11.5 Summary / 432
Exercises / 435

12 OPTIMIZATION OF RULE-BASED SYSTEMS

12.1 Introduction / 437
12.2 Background / 438
12.3 Basic Definitions / 439
12.4 Optimization Algorithm / 445
12.5 Experimental Evaluation / 455
This text is based on two rich sources: (1) materials in lecture notes I taught to senior and graduate-level computer science and electrical engineering students at Rice University and at the University of Houston, and (2) my research in the area of timing analysis and verification of real-time systems since the late 1980s, especially in the emerging area of embedded rule-based systems. Condensed forms of key concepts appearing in this text have been presented in my tutorials and seminars at many major international conferences. The focus is on the formal analysis and verification of real-time systems. The text is self-contained in that it includes a presentation of basic real-time scheduling algorithms and schedulability analysis as well as a description of the necessary background in logic and automata theory for understanding the more advanced materials. It provides a clear presentation of the concepts underlying the formal methods for real-time systems design.

Many of the systems and devices used in our modern society must provide a response that is both correct and timely. More and more computer systems are built as integral parts of many of these systems to monitor and control their functions and operations. These embedded systems often operate in environments where safety is a major concern. Examples range from simple systems such as climate-control systems, toasters, and rice cookers to highly complex systems such as airplanes and space shuttles. Other examples include hospital patient-monitoring devices and braking controllers in automobiles. Toward the goal of ensuring that these safety-critical systems operate as specified by the design and safety requirements, we have to develop sound methodologies and apply the corresponding tools to analyze and verify that these systems meet their specifications.

Much has been written in the area of formal analysis and verification of real-time systems in the form of technical papers that assume an advanced mathematical