Logical Foundations of Cyber-Physical Systems

André Platzer

Logical Foundations of Cyber-Physical Systems

Springer

André Platzer
Computer Science Department
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA

The content of the book and the image used on the book cover are based upon work performed at Carnegie Mellon University and supported by the National Science Foundation under NSF CAREER Award CNS-1054246.

ISBN 978-3-319-63587-3 ISBN 978-3-319-63588-0 (eBook)
https://doi.org/10.1007/978-3-319-63588-0

Library of Congress Control Number: 2018946565

© Springer International Publishing AG, part of Springer Nature 2018
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Printed on acid-free paper

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Endorsements

This excellent textbook marries design and analysis of cyber-physical systems with a logical and computational way of thinking. The presentation is exemplary for finding the right balance between rigorous mathematical formalization and illustrative case studies rooted in practical problems in system design.

Rajeev Alur, University of Pennsylvania

This book provides a wonderful introduction to cyber-physical systems, covering fundamental concepts from computer science and control theory from the perspective of formal logic. The theory is brought to life through many didactic examples, illustrations, and exercises. A wealth of background material is provided in the text and in an appendix for each chapter, which makes the book self-contained and accessible to university students of all levels.

Goran Frehse, Université Grenoble Alpes

[The author] has developed major important tools for the design and control of those cyber-physical systems that increasingly shape our lives. This book is a ‘must’ for computer scientists, engineers, and mathematicians designing cyber-physical systems.

Anil Nerode, Cornell University

As computing interfaces increasingly with our physical world, resulting in so-called cyber-physical systems, our foundations of computing need to be enriched with suitable physical models. This book strikes a wonderful balance between rigorous foundations for this next era of computing with illustrative examples and applications that drive the developed methods and tools. A must read book for anyone interested in the development of a modern and computational system science for cyber-physical systems.

George J. Pappas, University of Pennsylvania

This definitive textbook on cyber-physical systems lays the formal foundations of their behavior in terms of a single logical framework. Platzer’s logic stands out among all other approaches because it provides a uniform treatment of both the discrete and continuous nature of cyber-physical systems, and does not shy away from their complex behavior due to stochasticity, uncertainty, and adversarial agents in the environment. His computational thinking approach makes this work accessible to practicing engineers who need to specify and verify that cyber-physical systems are safe.

Jeannette M. Wing, Columbia University

Foreword

I first met André when he was just finishing his PhD and gave a job talk at CMU (he got the job). I was a visiting researcher and got to take the young faculty candidate out for lunch. André talked about verifying cyber-physical systems (CPS) using "differential dynamic logic" and theorem proving. I was skeptical, for one because related approaches had only seen modest success, and also because my money was on a different horse. A few years before, I had developed a model checker (PHAVer), and was working on a second one, called SpaceEx. At the time, these were the only verification tools that, on the push of a button, could verify certain benchmarks from CPS and other domains involving continuous variables that change with time. I was quite proud of them and, for me, algorithmic verification was the way to go. But André was determined to make theorem proving work in practice, and indeed, he advanced the field to an extent that I did not think possible. André and his team first developed the logical framework, then built a very capable theorem prover for CPS (KeYmaera), successfully applied it to industrial case studies like airplane collision avoidance, and, finally, addressed important application issues such as validating the model at runtime.

The book in front of you provides a comprehensive introduction on how to reason about cyber-physical systems using the language of logic and deduction. Along the way, you will become familiar with many fundamental concepts from computer science, applied mathematics, and control theory, all of which are essential for CPS. The book can be read without much prior knowledge, since all necessary background material is provided in the text and in appendices for many chapters. The book is structured in the following four parts. In the first part, you will learn how to model CPS with continuous variables and programming constructs, how to specify requirements and how to check whether the model satisfies the requirements using proof rules. The second part adds differential equations for modeling the physical world. The third part introduces the concept of an adversary, who can take actions that the system can not influence directly. In a control system, the adversary can be the environment, which influences the system behavior through noise and other disturbances. Making decisions in the presence of an adversary means trying to be prepared for the worst case. The fourth part adds further elements for reasoning soundly and efficiently about systems in applications, such as using real arithmetic and – my favorite – monitor conditions. Monitor conditions are checked while the system is in operation. As long as they hold, one can be sure that not only the model but also the actual CPS implementation satisfy the safety requirements.

By now André and his group have handled an impressive number of case studies that are beyond the capabilities of any model checker I know. Fortunately for me and my horse, the converse is also still true, since some problems can in practice only be solved numerically using algorithmic approaches. If your goal is to obtain a rock-solid foundation for CPS from the beautiful and elegant perspective of logics, then this is the book for you.

Goran Frehse, Associate Professor, Université Grenoble Alpes, Grenoble, 2017

Acknowledgements

This textbook is based on the lecture notes for the *Foundations of Cyber-Physical Systems* undergraduate course I taught in the Computer Science Department at Carnegie Mellon University. The textbook would have been impossible without the feedback from the students and helpful discussions with the teaching assistants João Martins, Annika Peterson, Nathan Fulton, Anastassia Kornilova, Brandon Bohrer, and especially Sarah Loos who TAed for the first instance in Fall 2013 and co-instructed for the intensive courses at ENS Lyon, France, in Spring 2014 and at MAP-i, Braga, Portugal in Summer 2014. Based on the experience with earlier Ph.D.-level courses, this course was originally designed as an undergraduate course but then extended to master's students and eventually Ph.D. students.

I appreciate the feedback of all my students on this textbook, but also by my postdocs Stefan Mitsch, Jean-Baptiste Jeannin, Khalil Ghorbal, and Jan-David Quesel. I am especially thankful to Sarah Loos's formative comments on the earliest draft and Yong Kiam Tan's careful extensive feedback for the final version. I am also grateful to Jessica Packer's exhaustive consistency checking on the textbook structuring and to Julia Platzer for crucial advice on illustrations. I am most indebted to the developers Stefan Mitsch and Nathan Fulton of the KeYmaera X prover for verifying cyber-physical systems, and very much appreciate also the KeYmaera X contributions by Brandon Bohrer, Yong Kiam Tan, Jan-David Quesel, and Marcus Völp. For help with the book process, I am grateful to the copyeditor and Ronan Nugent from Springer. Especially, however, I thank my family, without whose patience and support this book would not exist.

This textbook captures findings from the NSF CAREER Award on *Logical Foundations of Cyber-Physical Systems*, which I am very grateful to have received. I also benefitted from Helen Gill's advice as a program manager when this project started.

Funding

This material is based upon work supported by the National Science Foundation under NSF CAREER Award CNS-1054246.

Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Pittsburgh, December 2017 *André Platzer*

Disclaimer

This book is presented solely for educational purposes. While best efforts have been used in preparing this book, the author and publisher make no representations or warranties of any kind and assume no liabilities of any kind with respect to the accuracy or completeness of the contents and specifically disclaim any implied warranties of merchantability or fitness of use for a particular purpose. Neither the author nor the publisher shall be held liable or responsible to any person or entity with respect to any loss or incidental or consequential damages caused, or alleged to have been caused, directly or indirectly, by the information contained herein. No warranty may be created or extended by sales representatives or written sales materials.

Contents

1 Cyber-Physical Systems: Overview 1
1.1 Introduction 1
1.1.1 Cyber-Physical Systems Analysis by Example 2
1.1.2 Application Domains 3
1.1.3 Significance 4
1.1.4 The Importance of Safety 4
1.2 Hybrid Systems Versus Cyber-Physical Systems 6
1.3 Multi-dynamical Systems 7
1.4 How to Learn About Cyber-Physical Systems 10
1.5 Computational Thinking for Cyber-Physical Systems 12
1.6 Learning Objectives 12
1.7 Structure of This Textbook 15
1.8 Summary 18
References 19

Part I Elementary Cyber-Physical Systems 25

2 Differential Equations & Domains 27
2.1 Introduction 27
2.2 Differential Equations as Models of Continuous Physical Processes 28
2.3 The Meaning of Differential Equations 31
2.4 A Tiny Compendium of Differential Equation Examples 33
2.5 Domains of Differential Equations 39
2.6 Syntax of Continuous Programs 41
2.6.1 Continuous Programs 41
2.6.2 Terms 42
2.6.3 First-Order Formulas 43
2.7 Semantics of Continuous Programs 45
2.7.1 Terms 45

2.7.2 First-Order Formulas 47
2.7.3 Continuous Programs 51
2.8 Summary 52
2.9 Appendix 53
2.9.1 Existence Theorems 53
2.9.2 Uniqueness Theorems 54
2.9.3 Linear Differential Equations with Constant Coefficients . . 55
2.9.4 Continuation and Continuous Dependency 57
Exercises 58
References 61

3 Choice & Control 63
3.1 Introduction 63
3.2 A Gradual Introduction to Hybrid Programs 65
3.2.1 Discrete Change in Hybrid Programs 66
3.2.2 Compositions of Hybrid Programs 66
3.2.3 Decisions in Hybrid Programs 68
3.2.4 Choices in Hybrid Programs 69
3.2.5 Tests in Hybrid Programs 71
3.2.6 Repetitions in Hybrid Programs 73
3.3 Hybrid Programs 75
3.3.1 Syntax 75
3.3.2 Semantics 77
3.4 Hybrid Program Design 82
3.4.1 To Brake, or Not to Brake, That Is the Question 82
3.4.2 A Matter of Choice 83
3.5 Summary 84
3.6 Appendix: Modeling the Motion of a Robot Around a Bend 85
Exercises 87
References 92

4 Safety & Contracts 95
4.1 Introduction 95
4.2 A Gradual Introduction to CPS Contracts 97
4.2.1 The Adventures of Quantum the Bouncing Ball 98
4.2.2 How Quantum Discovered a Crack in the Fabric of Time . . 101
4.2.3 How Quantum Learned to Deflate 103
4.2.4 Postcondition Contracts for CPS 105
4.2.5 Precondition Contracts for CPS 106
4.3 Logical Formulas for Hybrid Programs 107
4.4 Differential Dynamic Logic 110
4.4.1 Syntax 110
4.4.2 Semantics 112
4.5 CPS Contracts in Logic 115
4.6 Identifying Requirements of a CPS 118

4.7 Summary 122
4.8 Appendix 123
4.8.1 Intermediate Conditions for a Proof of Sequential Compositions 124
4.8.2 A Proof of Choice 126
4.8.3 A Proof of Tests 127
Exercises 129
References 134

5 Dynamical Systems & Dynamic Axioms 137
5.1 Introduction 137
5.2 Intermediate Conditions for CPS 139
5.3 Dynamic Axioms for Dynamical Systems 142
5.3.1 Nondeterministic Choices 142
5.3.2 Soundness of Axioms 146
5.3.3 Assignments 146
5.3.4 Differential Equations 148
5.3.5 Tests 151
5.3.6 Sequential Compositions 152
5.3.7 Loops 155
5.3.8 Diamonds 156
5.4 A Proof of a Short Bouncing Ball 157
5.5 Summary 159
5.6 Appendix 160
5.6.1 Modal Modus Ponens Has Implications on Boxes 160
5.6.2 Vacuous State Change if Nothing Relevant Ever Changes . . 162
5.6.3 Gödel Generalizes Validities into Boxes 162
5.6.4 Monotonicity of Postconditions 163
5.6.5 Of Free and Bound Variables 164
5.6.6 Free and Bound Variable Analysis 165
Exercises 168
References 172

6 Truth & Proof 173
6.1 Introduction 173
6.2 Truth and Proof 175
6.2.1 Sequents 177
6.2.2 Proofs 179
6.2.3 Propositional Proof Rules 180
6.2.4 Soundness of Proof Rules 185
6.2.5 Proofs with Dynamics 187
6.2.6 Quantifier Proof Rules 190
6.3 Derived Proof Rules 192
6.4 A Sequent Proof for the Single-Hop Bouncing Ball 193
6.5 Real Arithmetic 195

6.5.1 Real Quantifier Elimination 196
6.5.2 Instantiating Real-Arithmetic Quantifiers 199
6.5.3 Weakening Real Arithmetic by Removing Assumptions . . 200
6.5.4 Structural Proof Rules in Sequent Calculus 201
6.5.5 Substituting Equations into Formulas 203
6.5.6 Abbreviating Terms to Reduce Complexity 203
6.5.7 Creatively Cutting Real Arithmetic to Transform Questions 204
6.6 Summary 205
Exercises 206
References 209

7 Control Loops & Invariants 211
7.1 Introduction 211
7.2 Control Loops 213
7.3 Induction for Loops 215
7.3.1 Induction Axiom for Loops 215
7.3.2 Induction Rule for Loops 217
7.3.3 Loop Invariants 220
7.3.4 Contextual Soundness Requirements 223
7.4 A Proof of a Happily Repetitive Bouncing Ball 225
7.5 Splitting Postconditions into Separate Cases 230
7.6 Summary 232
7.7 Appendix 233
7.7.1 Loops of Proofs 233
7.7.2 Breaking Loops of Proofs 235
7.7.3 Invariant Proofs of Loops 238
7.7.4 Alternative Forms of the Induction Axiom 239
Exercises 241
References 243

8 Events & Responses 245
8.1 Introduction 245
8.2 The Need for Control 247
8.2.1 Events in Control 248
8.2.2 Event Detection 250
8.2.3 Dividing Up the World 255
8.2.4 Event Firing 259
8.2.5 Event-Triggered Verification 260
8.2.6 Event-Triggered Control Paradigm 261
8.2.7 Physics Versus Control Distinctions 263
8.3 Summary 263
Exercises 264
References 266

9 Reactions & Delays 267
9.1 Introduction 267
9.2 Delays in Control 269
9.2.1 The Impact of Delays on Event Detection 272
9.2.2 Model-Predictive Control Basics 273
9.2.3 Design-by-Invariant 275
9.2.4 Sequencing and Prioritizing Reactions 276
9.2.5 Time-Triggered Verification 279
9.3 Summary 281
Exercises 282
References 284

Part II Differential Equations Analysis 285

10 Differential Equations & Differential Invariants 287
10.1 Introduction 287
10.2 A Gradual Introduction to Differential Invariants 289
10.2.1 Global Descriptive Power of Local Differential Equations . 290
10.2.2 Intuition for Differential Invariants 291
10.2.3 Deriving Differential Invariants 293
10.3 Differentials 295
10.3.1 Syntax 295
10.3.2 Semantics of Differential Symbols 296
10.3.3 Semantics of Differential Terms 299
10.3.4 Derivation Lemma with Equations of Differentials 301
10.3.5 Differential Lemma 303
10.3.6 Differential Invariant Term Axiom 304
10.3.7 Differential Substitution Lemmas 306
10.4 Differential Invariant Terms 308
10.5 A Differential Invariant Proof by Generalization 310
10.6 Example Proofs 311
10.7 Summary 313
10.8 Appendix 315
10.8.1 Differential Equations Versus Loops 315
10.8.2 Differential Invariant Terms and Invariant Functions 318
Exercises 320
References 321

11 Differential Equations & Proofs 323
11.1 Introduction 323
11.2 Recap: Ingredients for Differential Equation Proofs 326
11.3 Differential Weakening 327
11.4 Operators in Differential Invariants 329
11.4.1 Equational Differential Invariants 329

11.4.2 Differential Invariant Proof Rule 331
11.4.3 Differential Invariant Inequalities 332
11.4.4 Disequational Differential Invariants 335
11.4.5 Conjunctive Differential Invariants 336
11.4.6 Disjunctive Differential Invariants 338
11.5 Differential Invariants 339
11.6 Example Proofs 341
11.7 Assuming Invariants 343
11.8 Differential Cuts 346
11.9 Differential Weakening Again 350
11.10 Differential Invariants for Solvable Differential Equations 350
11.11 Summary 352
11.12 Appendix: Proving Aerodynamic Bouncing Balls 353
Exercises 358
References 360

12 Ghosts & Differential Ghosts 363
12.1 Introduction 363
12.2 Recap 366
12.3 A Gradual Introduction to Ghost Variables 366
12.3.1 Discrete Ghosts 366
12.3.2 Proving Bouncing Balls with Sneaky Solutions 368
12.3.3 Differential Ghosts of Time 374
12.3.4 Constructing Differential Ghosts 375
12.4 Differential Ghosts 378
12.5 Substitute Ghosts 383
12.6 Limit Velocity of an Aerodynamic Ball 384
12.7 Axiomatic Ghosts 387
12.8 Summary 388
12.9 Appendix 389
12.9.1 Arithmetic Ghosts 389
12.9.2 Nondeterministic Assignments & Ghosts of Choice 390
12.9.3 Differential-Algebraic Ghosts 392
Exercises 394
References 395

13 Differential Invariants & Proof Theory 397
13.1 Introduction 397
13.2 Recap 400
13.3 Comparative Deductive Study: Relativity Theory for Proofs 401
13.4 Equivalences of Differential Invariants 402
13.5 Differential Invariants & Arithmetic 403
13.6 Differential Invariant Equations 405
13.7 Equational Incompleteness 407
13.8 Strict Differential Invariant Inequalities 410

13.9 Differential Invariant Equations as Differential Invariant Inequalities 412
13.10 Differential Invariant Atoms 413
13.11 Summary 414
13.12 Appendix: Curves Playing with Norms and Degrees 414
Exercises 416
References 416

Part III Adversarial Cyber-Physical Systems 419

14 Hybrid Systems & Games 421
14.1 Introduction 421
14.2 A Gradual Introduction to Hybrid Games 424
14.2.1 Choices & Nondeterminism 424
14.2.2 Control & Dual Control 426
14.2.3 Demon's Derived Controls 427
14.3 Syntax of Differential Game Logic 428
14.3.1 Hybrid Games 429
14.3.2 Differential Game Logic Formulas 432
14.3.3 Examples 433
14.4 An Informal Operational Game Tree Semantics 439
14.5 Summary 443
Exercises 444
References 447

15 Winning Strategies & Regions 449
15.1 Introduction 449
15.2 Semantics of Differential Game Logic 451
15.2.1 Limits of Reachability Relations 451
15.2.2 Set-Valued Semantics of Differential Game Logic Formulas 452
15.2.3 Winning-Region Semantics of Hybrid Games 453
15.3 Semantics of Repetition in Hybrid Games 458
15.3.1 Repetitions with Advance Notice 458
15.3.2 Repetitions as Infinite Iterations 460
15.3.3 Inflationary Semantics of Repetition 465
15.3.4 Characterizing Winning Repetitions Implicitly 469
15.4 Semantics of Hybrid Games 473
15.5 Summary 476
Exercises 476
References 478

16 Winning & Proving Hybrid Games 479
16.1 Introduction 479
16.2 Semantical Considerations 481
16.2.1 Monotonicity 481
16.2.2 Determinacy 482

16.3 Dynamic Axioms for Hybrid Games . . . 484
16.3.1 Determinacy . . . 484
16.3.2 Monotonicity . . . 485
16.3.3 Assignments . . . 486
16.3.4 Differential Equations . . . 486
16.3.5 Challenge Games . . . 488
16.3.6 Choice Games . . . 488
16.3.7 Sequential Games . . . 490
16.3.8 Dual Games . . . 490
16.3.9 Repetition Games . . . 492
16.3.10 Proof Rules for Repetition Games . . . 494
16.4 Example Proofs . . . 495
16.5 Axiomatization . . . 498
16.5.1 Soundness . . . 499
16.5.2 Completeness . . . 501
16.6 There and Back Again Game . . . 503
16.7 Summary . . . 504
Exercises . . . 504
References . . . 507

17 Game Proofs & Separations 509
17.1 Introduction . . . 509
17.2 Recap: Hybrid Games . . . 510
17.3 Separating Axioms . . . 511
17.4 Repetitive Diamonds – Convergence Versus Iteration . . . 516
17.5 Summary . . . 519
17.6 Appendix: Relating Differential Game Logic and Differential Dynamic Logic . . . 520
Exercises . . . 521
References . . . 521

Part IV Comprehensive CPS Correctness 523

18 Axioms & Uniform Substitutions 525
18.1 Introduction . . . 525
18.2 Axioms Versus Axiom Schemata . . . 528
18.3 What Axioms Want . . . 530
18.4 Differential Dynamic Logic with Interpretations . . . 533
18.4.1 Syntax . . . 533
18.4.2 Semantics . . . 535
18.5 Uniform Substitution . . . 536
18.5.1 Uniform Substitution Rule . . . 537
18.5.2 Examples . . . 539
18.5.3 Uniform Substitution Application . . . 542

18.5.4 Uniform Substitution Lemmas 545
18.5.5 Soundness 546
18.6 Axiomatic Proof Calculus for dL 547
18.7 Differential Axioms 549
18.8 Summary 551
18.9 Appendix: Uniform Substitution of Rules and Proofs 551
Exercises 552
References 555

19 Verified Models & Verified Runtime Validation 557
19.1 Introduction 557
19.2 Fundamental Challenges with Inevitable Models 559
19.3 Runtime Monitors 562
19.4 Model Compliance 565
19.5 Provably Correct Monitor Synthesis 568
19.5.1 Logical State Relations 569
19.5.2 Model Monitors 571
19.5.3 Correct-by-Construction Synthesis 571
19.6 Summary 573
Exercises 574
References 575

20 Virtual Substitution & Real Equations 577
20.1 Introduction 577
20.2 Framing the Miracle 580
20.3 Quantifier Elimination 583
20.3.1 Homomorphic Normalization for Quantifier Elimination . . 585
20.3.2 Substitution Base 587
20.3.3 Term Substitutions for Linear Equations 588
20.4 Square Root $\sqrt{\cdot}$ Virtual Substitutions for Quadratics 590
20.4.1 Square Root Algebra 592
20.4.2 Virtual Substitutions of Square Roots 595
20.5 Optimizations 599
20.6 Summary 599
20.7 Appendix: Real Algebraic Geometry 600
Exercises 601
References 603

21 Virtual Substitution & Real Arithmetic 607
21.1 Introduction 607
21.2 Recap: Square Root $\sqrt{\cdot}$ Virtual Substitutions for Quadratics 609
21.3 Infinity ∞ Virtual Substitution 609
21.4 Infinitesimal ε Virtual Substitution 612
21.5 Quantifier Elimination by Virtual Substitution for Quadratics . . . 618
21.6 Optimizations 623

21.7 Summary 624
21.8 Appendix: Semialgebraic Geometry 625
Exercises 625
References 627

Index **629**

Operators & Axioms **637**

List of Figures

1.1 Airplane example: Which control decisions are safe for aircraft collision avoidance? . . . 2
1.2 Multi-dynamical systems aspects of CPS . . . 8
1.3 Dependencies and suggested reading sequences of the chapters . . 17

2.1 Vector field with one solution of a differential equation . . . 29
2.2 Vector field with one solution of accelerated straight-line motion . 30
2.3 Discretizations of differential equations with a discretization time step . . . 31
2.4 Differential equation solution condition . . . 32
2.5 Constant differential equation . . . 33
2.6 Linear differential equation . . . 33
2.7 A solution of the rotational differential equations . . . 36
2.8 Another solution of the rotational differential equations with initial values 1 . . . 37
2.9 A faster solution of the rotational differential equations with initial values 1 . . . 37
2.10 A solution of the time square oscillator and the damped oscillator . 38
2.11 System $x' = f(x) \,\&\, Q$ follows the differential equation $x' = f(x)$ for any duration but cannot leave evolution domain Q . . . 41
2.12 Illustration of the dynamics of continuous programs . . . 51

3.1 An illustration of the behavior of an instantaneous discrete change . 66
3.2 Fixed acceleration, velocity, and position change over time . . . 67
3.3 Acceleration, velocity, and position change over time . . . 68
3.4 Transition semantics and example dynamics of hybrid programs . . 78
3.5 Nested transition semantics pattern for $(\alpha;\beta)^*$ . . . 81
3.6 Nested transition semantics pattern for $(\alpha \cup \beta)^*$ . . . 81
3.7 Transition structure of the acceleration/braking example . . . 83
3.8 Illustration of a Dubins path consisting of a sequence of lines and maximally curved circle segments . . . 86

3.9 Illustration of the Dubins dynamics of a point (x,y) moving in direction (v,w) along a dashed curve with angular velocity ω 86
3.10 Hybrid automaton for a car that can accelerate or brake 91

4.1 Sample trajectory of a bouncing ball 98
4.2 Sample trajectory of a bouncing ball with a crack in the floor . . . 101
4.3 Sample trajectory of a bouncing ball over its hybrid time domain . 102
4.4 Hybrid time domain for the sample trajectory of a bouncing ball . . 103
4.5 Sample trajectory of a bouncing ball that ultimately lies down flat . 104
4.6 Transition semantics of modalities in dL formulas 115
4.7 Sample trajectory of a bouncing ball in an anti-gravity field 118
4.8 Sample trajectory of a bouncing ball with anti-damping 120
4.9 Sample trajectory of a bouncing ball with upwards initial velocity . 120
4.10 Sample trajectory of a bouncing ball dribbling with fast initial velocity 121

5.1 Sample trajectory of a single-hop bouncing ball 140
5.2 Intermediate conditions for sequential compositions 142
5.3 Illustration of dynamic axiom for sequential composition 154
5.4 Summary of sound differential dynamic logic axioms from this chapter 160
5.5 Additional axioms and proof rules for hybrid systems 171

6.1 Propositional proof rules of sequent calculus 180
6.2 A simple propositional example proof in sequent calculus 184
6.3 A simple example proof with dynamics in sequent calculus 189
6.4 Quantifier sequent calculus proof rules 190
6.5 Sequent calculus proof for gravity above ground 193
6.6 Proof rules of the dL sequent calculus considered in this chapter . . 206

7.1 Successively using induction axiom I at each state reached after running iterations of α^* 218
7.2 Sequent calculus proof shape for bouncing ball 226
7.3 Sequent calculus proof for bouncing ball with split 231
7.4 Summary of proof rules for loops, generalization, monotonicity, and splitting boxes 232
7.5 Loops of proofs: iterating and splitting the box 234
7.6 Loops of proofs: iterating and generalizing the box 236
7.7 Loops of proofs: intermediate generalizations 238
7.8 Derivation of backwards unwinding axiom from alternative induction axiom 240

8.1 Sample trajectory of a bouncing ball bouncing freely 248
8.2 Sample trajectory of a ping-pong ball 249
8.3 Sample trajectory of a ping-pong ball which misses one event . . . 250

8.4 Sample trajectory of a ping-pong ball, sometimes actuating early, sometimes late . . . 252
8.5 Sample trajectory of a ping-pong ball with the controller firing multiple times for the same event . . . 258
8.6 Sample trajectory of a ping-pong ball with the controller firing multiple times for the same event on the event boundary . . . 258

9.1 Sample trajectory of a ping-pong ball, sometimes actuating early, sometimes late . . . 270
9.2 Sample trajectory of a time-triggered ping-pong ball missing the first event . . . 272
9.3 Sample trajectory of a time-triggered ping-pong ball missing different events with different sampling periods . . . 274
9.4 Sample trajectory of a time-triggered ping-pong ball failing to control on the ground . . . 276
9.5 Sample trajectory of a time-triggered ping-pong ball stuck on the ground . . . 277

10.1 Vector field and one solution of a differential equation that does not enter the unsafe regions . . . 292
10.2 One scenario for the rotational dynamics and relationship of a direction vector to the radius and angle . . . 293
10.3 Differential invariant remains true in the direction of the dynamics . 294
10.4 Semantics of differential symbol x' along differential equation . . . 299
10.5 Differential form semantics of differentials: their value depends on the point as well as on the direction of the vector field at that point . 301
10.6 Differential invariant of the indicated dynamics . . . 311
10.7 Two differential invariants of the indicated self-crossing dynamics . 312
10.8 Two differential invariants of the indicated dynamics for the Motzkin polynomial . . . 313
10.9 Axioms for differential invariant terms of differential equations without solutions . . . 314

11.1 Differential weakening axiom DW . . . 328
11.2 Equal rate of change from equal initial value . . . 330
11.3 Differential invariant for safety . . . 332
11.4 Lesser or equal rate of change from lesser or equal initial value . . 333
11.5 Cubic dynamics proof . . . 334
11.6 Cubic dynamics . . . 334
11.7 Lesser or equal rate of change from lesser initial value . . . 334
11.8 Unsound attempt to use disequalities . . . 335
11.9 Linear evolution of $x' = 1$ . . . 335
11.10 Different rates of change from different initial values do not prove anything . . . 336
11.11 Soundness proof for conjunctive differential invariant axiom 337

11.12 Differential invariant proof for bouncing ball in gravity 338
11.13 Soundness proof for disjunctive differential invariant axiom 339
11.14 Damped-oscillator time trajectory and invariant in phase space . . . 342
11.15 Illustration of the Dubins dynamics of a point (x,y) moving in direction (v,w) along a dashed curve with angular velocity ω 343
11.16 If the solution of the differential equation can never leave region C and enter the red region $\neg C$, then this unreachable region $\neg C$ can be cut out of the state space without changing the dynamics of the system 347
11.17 Trajectory with vector field and evolution of an increasingly damped oscillator 348
11.18 Differential cut proof for the increasingly damped oscillator 348
11.19 If the solution of the differential equation can never leave region D and enter the region $\neg D$, then this unreachable region $\neg D$ can also be cut out of the state space without changing the dynamics of the system 349
11.20 Axioms and proof rules for differential invariants and differential cuts of differential equations 353

12.1 Exponential decay along $x' = -x$ always makes matters worse for $x > 0$ 376
12.2 Differential ghost y as counterweight for exponential decay along $x' = -x$ 378
12.3 Explosive differential ghosts that do not exist long enough would unsoundly limit the duration of solutions 379
12.4 Differential ghost y to balance exponential growth along $x' = x$ . . 381
12.5 Differential ghost y as counterweight for square resistance along $x' = -x^2$ 383
12.6 Velocity of aerodynamic ball approaches limit velocity 385
12.7 Dubins aircraft dynamics 387
12.8 Reparametrize for differential axiomatization 388
12.9 Axioms and proof rules for ghosts and differential ghosts where y is new 389
12.10 Axioms for nondeterministic assignments 391

13.1 Equivalent solutions with quite different differential structure . . . 405
13.2 Differential invariance chart 414
13.3 p-norm inclusions 416

14.1 Turning hybrid game α into the dual hybrid game α^{d} corresponds to turning a chessboard around by 180° so that the players control the choices in α^{d} that the opponent has in α 427
14.2 Angel and Demon accelerating or braking by a and d, respectively, the cart at position x, which is moving with velocity x 431

14.3 Velocities and accelerations of two robots on a one-dimensional planet . . . 435
14.4 Goalie in robot soccer moves and, if within radius 1, can capture the ball . . . 438
14.5 Operational game semantics for hybrid games of dGL . . . 440
14.6 The filibuster game formula looks as though it might be non-determined and not have a truth-value . . . 442

15.1 Denotational semantics of hybrid games as Angel's winning region 456
15.2 Denotational semantics of hybrid games as Demon's winning region 457
15.3 Monotonicity: it is easier to win into larger sets of winning states $Y \supseteq X$ . . . 457
15.4 Game trees for $x = 1 \land a = 1 \to \langle((x:=a;a:=0) \cap x:=0)^*\rangle x \neq 1$. 459
15.5 Iteration $\varsigma_\alpha^n(X)$ of $\varsigma_\alpha(\cdot)$ from winning condition X . . . 461
15.6 Winning regions $\varsigma_\alpha(Z)$ of sets $Z \subseteq \varsigma_{\alpha^*}(X)$ are already included in $\varsigma_{\alpha^*}(X)$ since $\varsigma_\alpha(Z)$ is just one more round away from Z . . . 463
15.7 Iteration $\varsigma_\alpha^{\omega+1}(X)$ of $\varsigma_\alpha(\cdot)$ from winning condition $X = [0,1)$ stops when applying $\varsigma_\alpha(\cdot)$ to the ωth infinite iteration $\varsigma_\alpha^\omega(X)$ . . . 464
15.8 Illustration of infinitely many ordinals up to ω^ω . . . 465
15.9 Transfinite iteration $\varsigma_\alpha^\infty(X)$ of $\varsigma_\alpha(\cdot)$ from winning condition X results in winning region $\varsigma_{\alpha^*}(X)$ of repetition . . . 466
15.10 Illustration of denotational semantics of winning region of hybrid game repetitions . . . 471

16.1 Proof of the two-robot dance . . . 497
16.2 Differential game logic axiomatization . . . 498
16.3 "There and back again game" . . . 503
16.4 Differential game logic derived axioms for box modalities . . . 505
16.5 Differential game logic derived axioms for Demon's controls . . . 505
16.6 More hybrid systems axioms, some of which are sound for hybrid games . . . 506

17.1 Differential game logic axiomatization (repeated) . . . 511
17.2 Separating axioms sound for hybrid systems but not hybrid games . 512
17.3 dGL Angel proof for non-game system Example 17.1 $x \geq 0 \to \langle(x:=x-1)^*\rangle 0 \leq x < 1$ . . . 518
17.4 dGL Angel proof for demonic choice game Example 17.2 $x = 1 \land a = 1 \to \langle(x:=a;a:=0 \cap x:=0)^*\rangle x \neq 1$ . . . 518
17.5 dGL Angel proof for 2-Nim-type game Example 17.3 $x \geq 0 \to \langle(x:=x-1 \cap x:=x-2)^*\rangle 0 \leq x < 2$ . . . 518
17.6 dGL Angel proof for hybrid game Example 17.4 $\langle(x:=1;x' = 1^{\mathrm{d}} \cup x:=x-1)^*\rangle 0 \leq x < 1$ . . . 519

18.1 Recursive application of uniform substitution σ . . . 542
18.2 Differential dynamic logic axioms and proof rules . . . 548
18.3 Differential equation axioms and differential axioms . . . 549

19.1 ModelPlex monitors sit between controller and actuator to check the controller's decisions for compliance with the model based on sensor data with veto leading to a safe fallback action 563
19.2 Use of ModelPlex monitors along a system run 564
19.3 Sample run of a bouncing ball that ultimately lies down flat 566

20.1 The geometric counterpart of quantifier elimination for $\exists y$ is projection onto the x axis 585
20.2 Roots of different quadratic functions p 591
20.3 Polynomial equations describe (real) affine (algebraic) varieties 601

21.1 Illustration of the value of different quadratic functions p where $p_{\bar{\mathsf{x}}}^{-\infty} \equiv \mathit{true}$ 612
21.2 Illustration of the sign after the second root for quadratic functions 617
21.3 Illustration of roots e and infinitesimal offsets $e+\varepsilon$ checked by virtual substitution 620
21.4 Systems of polynomial inequalities describe semialgebraic sets 626

List of Tables

2.1 Operators and meaning in first-order logic of real arithmetic (FOL) . 52

3.1 Statements and effects of hybrid programs (HPs) 85
3.2 Classification of hybrid programs and correspondence to dynamical systems 90

4.1 Operators and (informal) meaning in differential dynamic logic (dL) 123

10.1 Correspondence map between loops and differential equations . . . 316

14.1 Operators and (informal) meaning in differential game logic (dGL) . 443
14.2 Statements and effects of hybrid games (HGs) 444

20.1 Overview of decidability notions (e.g., for the validity problem) . . 581
20.2 The miracle of reals: overview of FOL validity problems 582

List of Expeditions

2.1 Naming conventions 44
2.2 Semantic brackets $[\![\cdot]\!] : \mathrm{Trm} \to (\mathscr{S} \to \mathbb{R})$ 47
3.1 Operator precedence for hybrid programs 77
3.2 HP semantics $[\![\cdot]\!] : \mathrm{HP} \to \wp(\mathscr{S} \times \mathscr{S})$ 81
4.1 Three Laws of Robotics 105
4.2 Invariant contracts for CPS 107
4.3 Operator precedence for differential dynamic logic 113
4.4 Set-valued dL semantics $[\![\cdot]\!] : \mathrm{Fml} \to \wp(\mathscr{S})$ 115
4.5 Principle of Cartesian Doubt 123
5.1 Admissibility caveats for the $p(x)$ notation in axioms 148
6.1 Quantifier elimination 197
9.1 Zeno paradox 278
10.1 Denotational semantics 298
10.2 Differential algebra 317
10.3 Semantics of differential algebra 318
10.4 Lie characterization of invariant functions 319
13.1 Proving differences in set theory and linear algebra 409
13.2 Topology in real analysis 411
15.1 Ordinal numbers 467
15.2 Ordinal arithmetic 468
21.1 Infinite challenges with infinities in extended reals 613
21.2 Nonstandard analysis: infinite challenges with infinitesimal ε 618

List of Theorems

T 2.1 Peano’s existence theorem 53
T 2.2 Picard-Lindelöf uniqueness theorem 54
P 2.1 Linear differential equations with constant coefficients 55
P 2.2 Continuation of solutions 57
P 2.3 Lipschitz estimation 58
L 5.1 $[\cup]$ axiom of nondeterministic choice 144
L 5.2 $[:=]$ assignment axiom 147
L 5.3 $[']$ solution axiom 149
L 5.4 $[']$ solution with domain axiom 149
L 5.5 $[?]$ test axiom 151
L 5.6 $[;]$ composition axiom 153
L 5.7 $[^*]$ iteration axiom 155
L 5.8 $\langle\cdot\rangle$ duality axiom 157
T 5.1 Soundness 159
L 5.9 K modal modus ponens axiom 161
L 5.10 $[]\wedge$ boxes distribute over conjunctions 161
L 5.11 V vacuous axiom 162
L 5.12 G Gödel generalization rule 162
L 5.13 $M[\cdot]$ monotonicity rule 163
L 6.1 $\wedge R$ conjunction rule 185
T 6.1 Soundness 186
L 6.2 Contextual equivalence 188
L 6.3 $\mathbb{R}$ real arithmetic 195
T 6.2 Tarski’s quantifier elimination 197
L 6.4 $i\forall$ reintroducing universal quantifiers 198
L 6.5 $[:=]_=$ equational assignment rule 204
L 7.1 I induction axiom 217
L 7.2 ind induction rule 219
L 7.3 Loop invariant rule 221
P 7.1 Quantum is safe 230
L 7.4 MR monotonicity right rule 237

L 7.5 $\overleftarrow{[*]}$ backwards iteration axiom 239
L 7.6 $[**]$ double iteration axiom 240
P 8.1 Event-triggered ping-pong is safe 263
P 9.1 Time-triggered ping-pong is safe 280
L 10.1 Derivation lemma 302
L 10.2 Differential lemma 304
L 10.3 Differential invariant term axiom 305
L 10.4 Mean-value theorem 305
L 10.5 Differential assignment 307
L 10.6 DE differential effect axiom 307
L 10.7 Differential invariant term rule 309
T 10.1 Lie's characterization of invariant terms 319
L 11.1 DW differential weakening axiom 327
L 11.2 dW differential weakening proof rule 328
L 11.3 dI differential invariant proof rule 331
L 11.4 DI differential invariant axiom 340
L 11.5 dC differential cut proof rule 346
P 11.1 Increasingly damped oscillation 347
L 11.6 DC differential cut axiom 349
P 11.2 Aerodynamic Quantum is safe 357
L 12.1 iG discrete ghost rule 367
L 12.2 DG differential ghost axiom 379
L 12.3 dG differential ghost rule 380
L 12.4 dA differential auxiliaries rule 380
P 12.1 Aerodynamic velocity limits 386
L 13.1 Differential invariants and propositional logic 402
L 13.2 Differential invariants and arithmetic 403
P 13.1 Equational deductive power 406
P 13.2 Equational incompleteness 408
P 13.3 Strict barrier incompleteness 410
P 13.4 Equational definability 412
T 13.1 Atomic incompleteness 413
L 15.1 Monotonicity 456
L 15.2 Intersection closure 471
L 15.3 Transfinite inflation leads to a least fixpoint 473
T 16.1 Consistency & determinacy 483
L 16.1 $[\cdot]$ determinacy axiom 484
L 16.2 M monotonicity rule 485
L 16.3 $\langle :=\rangle$ assignment axiom 486
L 16.4 $\langle ' \rangle$ solution axiom 487
L 16.5 $\langle ' \rangle$ solution with domain axiom 487
L 16.6 $\langle ? \rangle$ test axiom 488
L 16.7 $\langle \cup \rangle$ axiom of choice 489
L 16.8 $\langle ; \rangle$ composition axiom 490
L 16.9 $\langle ^d \rangle$ duality axiom 491

L 16.10 $\langle{}^*\rangle$ iteration axiom 493
L 16.11 FP fixpoint rule 494
P 16.1 Push-around carts are safe 496
P 16.2 Robot dance is safe 497
T 16.2 Soundness of dGL 500
T 16.3 Relative completeness of dGL 502
L 16.12 Evolution domain reduction 503
L 18.1 V vacuous axiom 536
L 18.2 $[:=]$ assignment axiom 536
T 18.1 Uniform substitution 538
L 18.3 Uniform substitution for formulas 546
T 18.2 Axiomatization of dL 551
T 18.3 Uniform substitution of rules 552
P 19.1 Correct bouncing-ball model monitor 571
T 20.1 Virtual substitution of linear equations 589
L 20.1 Uniform substitution of linear equations 590
T 20.2 Virtual substitution of quadratic equations 591
L 20.2 Virtual substitution lemma for square roots 595
L 21.1 Virtual substitution lemma for infinities 612
L 21.2 Virtual substitution lemma for infinitesimals 617
T 21.1 Virtual substitution of quadratic constraints 618
T 21.2 Tarski-Seidenberg 625