

Logical Analysis of Hybrid Systems

André Platzer

Logical Analysis of Hybrid Systems

Proving Theorems for Complex Dynamics

 Springer

Dr. André Platzer
Carnegie Mellon University
School of Computer Science
5000 Forbes Ave.
Pittsburgh PA 15213
USA
aplatzer@cs.cmu.edu

ISBN 978-3-642-14508-7 e-ISBN 978-3-642-14509-4
DOI 10.1007/978-3-642-14509-4
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2010934645

ACM Computing Classification (1998): F.4.1, F.3, D.2.4, I.2.3, G.1.7, I.2.8

© Springer-Verlag Berlin Heidelberg 2010

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, 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.

Cover design: KuenkelLopka GmbH, Heidelberg

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Foreword

Hybrid Systems are notoriously hard to analyze and verify. So far, techniques based on either explicit or implicit state reachability have failed to scale with the size of such systems. Statistical Model Checking may prove useful, but sacrifices absolute certainty about the correctness of the answer obtained. In both cases, numerical error may change the truth-value of the result from False to True or vice versa. An alternative is to use a combination of decision procedures for real arithmetic and interactive theorem proving. Andre Platzer’s Ph.D. thesis explores this alternative approach in great depth. He proposes a logic called Differential Dynamic Logic for specifying properties of Hybrid Systems, investigates the meta-theory of the logic, and gives inference rules for it. He has developed an extremely impressive graphical interface for the resulting tool KeYmaera, which is based on the KeY Prover for verifying Java programs, developed at the University of Karlsruhe, Chalmers, and Koblenz. Particularly noteworthy are the use of Differential Invariants for reasoning about complex Hybrid Systems and the examples that he is able to do: “The European Train Control System” and a curved flight roundabout maneuver for aircraft collision avoidance. Both examples are beyond the scope of current Hybrid System Model Checking tools. I believe that his verification tool is unique – there is no other one like it. I heartily recommend his book and theorem prover for those who need to verify complex cyber-physical systems.

Pittsburgh, February 2010

Edmund M. Clarke

Preface

The design of complex systems is essential to much of engineering and science. Equally essential is the effort to fully understand these systems and to develop tools and techniques that can steer us away from unsafe or incorrect designs. In civil engineering, for example, well-understood principles like statics can be used to analyse buildings before they are built, and refined architectural models can be used to predict whether a building will be safe or whether it might collapse during an earthquake. Similarly, in auto body design, wind tunnels and corresponding computer models based on computational fluid dynamics help engineers to gain an understanding of aerodynamic forces and wind resistance for energy efficiency before constructing the actual car and its chassis. Models and their analysis also play an important role in chip design and are used extensively in the semiconductor industry to prevent expensive bugs in hardware. Modelling and model analysis is thus an integral part of science and engineering and is used very effectively in many areas to ensure high-quality system designs, saving replacement cost and preventing dangerous side effects of malfunctioning designs.

Hybrid systems is an emergent area of growing importance, emphasising a systematic understanding of systems that combine discrete (e.g., digital) and continuous (e.g., analog or physical) effects. In fact, it is foreseeable that hybrid systems and the closely related notion of cyber-physical systems will soon play a ubiquitous role in engineering. Combinations of computation and control can lead to very powerful system designs, and computational aspects are being integrated into classical physical, mechanical, and chemical process controls on a routine basis today. The number of systems where both computational and physical aspects are important for really understanding them grows exponentially with modern technological advances. Hybrid systems occur frequently in automotive industries, aviation, railway applications, factory automation, process control, medical devices, mobile robotics, and mixed analog–digital chip design.

Despite the growing relevance in complex system designs, hybrid systems is an area where analytic approaches are still in their infancy. Hybrid systems occur ubiquitously and their analysis faces inherent complexity challenges. Hence, there is probably no other area where the gap is more noticeable between the tremendous

complexity of the systems we can build and the modest size of systems that we can analyse. Mankind can build systems that are significantly more complicated than people can understand analytically. This book presents an approach with logical analysis techniques that are intended to help overcome these difficulties and bridge the gap between design demand and analysis power.

In light of this growing interest in the field, the purpose of this book is to provide an introduction to hybrid systems analysis and, in particular, to present a coherent logical analysis approach for hybrid systems. One of the highly successful techniques used for analysing finite-state models in chip designs today is *model checking*, which was pioneered in 1981 by the 2007 ACM Turing Award Laureates Edmund M. Clarke, Allen Emerson, and Joseph Sifakis. Nowadays, model checking is used routinely in the semiconductor industry. Model checking is one of the inspirations for this work. Another area that is strongly related is interactive and *automated theorem proving*, which is also used in advanced industrial settings. Model checking and automated theorem proving complement each other to tackle various aspects of formal system verification. While both areas are ultimately rooted in logic, the basic operating principles are somewhat different. Model checking is based on systematically exploring the state space of a system in a clever way. Model checking searches for counterexamples, i.e., traces of a system that lead to a bug and that serve as a falsification of a correctness property. Impressive results have been demonstrated for finite-state systems where model checking is decidable. In theorem proving, in contrast, the notion of a proof is fundamental and represents a verification of a correctness property. In particular, a proof is a reason and explanation for *why* a system works. Automated theorem proving techniques that construct proofs automatically are another deep source of inspiration for the work presented here. In fact, several of the proof procedures presented in this book are inspired by theorem proving principles that have been used successfully for conventional object-oriented programs.

One important new aspect in hybrid systems is the cardinality and structure of the state space. In (sufficiently small) finite state spaces, for instance, exhaustive state exploration is still feasible, but becomes inherently impossible for the uncountable continuous state spaces of hybrid systems, especially with respect to their complicated interacting discrete and continuous dynamics. Most notably, the continuous dynamics of hybrid systems that is commonly described by differential equations poses significant new challenges compared to classical settings. Thus, verification techniques for differential equations are one very important part of hybrid systems analysis.

Outline

This book is intended as an introduction to hybrid systems and advanced analysis techniques for their dynamics. It covers basic and advanced notions of hybrid systems, specification languages for hybrid systems, verification approaches for hybrid systems, and application scenarios for hybrid systems verification. Starting from a basic background in mathematics and computer science, this book develops all

notions required for understanding and analysing hybrid systems. It also provides background material about logic and differential equations in the appendix.

This book presents a coherent logical foundation for hybrid systems analysis that will help the reader understand how behavioural properties of hybrid systems can be analysed successfully. The foundation developed here serves as a basis for advanced hybrid system analysis techniques. The hybrid systems analysis approach has also been implemented in the verification tool KeYmaera for hybrid systems, which is available for download at the book's Web page.

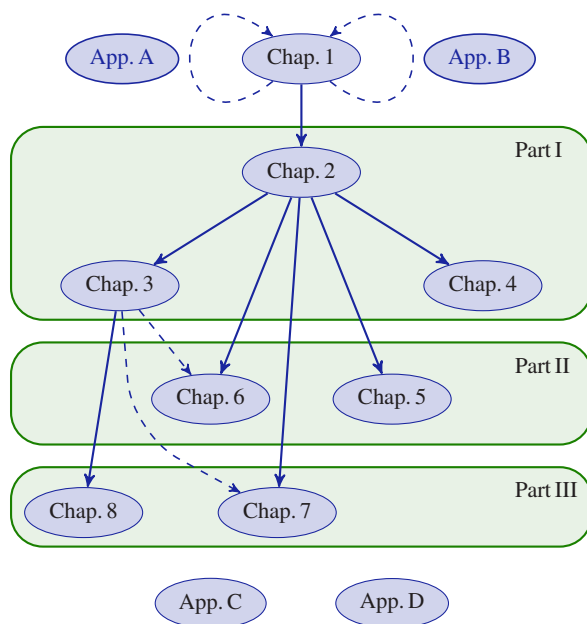
Part I describes specification and verification logics for hybrid systems that are the basis of hybrid systems analysis. It also covers constructive proof calculi that can be used to analyse and verify hybrid systems, including approaches for handling real arithmetic and differential equations. The chapters in Part I show a series of logical systems that are each presented in terms of their syntax, semantics, axiomatics, proof theory, and pragmatics. The syntax defines what can be said about hybrid system behaviour. The semantics gives meaning to the symbolic formulas and shows what we are ultimately interested in: truth, or, more precisely, what is true about the behaviour of the particular dynamics of a hybrid system. In the axiomatic parts, this book develops formal proof techniques that can be used by a human or machine to establish truth by proof. After all, truth that we do not know about is less helpful than truth that we can justify by giving a proof. The development of proof theory connects the semantic notion of truth in the real world with the syntactic device of formal proofs and shows that, in a sense of relative completeness, we can prove all true facts about hybrid systems from elementary properties of differential equations. Finally, this book shows the pragmatics of using verification procedures for analysing hybrid system scenarios. This includes both practical, algorithmic considerations of developing system analysis tools and various examples, application scenarios, and case studies that can be proven with the logics developed in Part I.

Part II focuses on the practical and algorithmic questions of how to turn the theoretical foundation from Part I into automated theorem proving procedures. This part also shows techniques for generating invariants and differential invariants of hybrid systems that are crucial for proving correctness, and shows how to overcome complexity challenges in real arithmetic verification. Part III shows how safety-critical properties of more advanced applications of hybrid systems in railway and aircraft control can be proven with the approach presented in Parts I and II. This part includes a study of collision avoidance in the European Train Control System (ETCS) and roundabout collision avoidance manoeuvres in air traffic control. Numerous examples, illustrations, and proofs throughout the text will also help the reader develop an intuition about hybrid systems behaviour and master the intricacies of the more subtle aspects in hybrid systems analysis.

How to Read This Book

The basic suggested reading sequence is linear (with additional consultation of the appendices for background information as needed). Except for the foundation of this

work that is laid out in Part I, however, the chapters are mostly kept self-contained so that they can also be studied independently. The following figure shows the reading order dependencies among the chapters (solid lines) and the partial dependencies of suggested reading sequences that hold for the advanced material of the respective chapters (dashed lines).



For background on classical first-order logic, we recommend you review App. A as needed. Depending on your interest, field of study, and preference, we recommend you either study the background information in App. A on first-order logic before reading Part I or use the material in App. A as a background reference book on demand while reading the main part of this book. Similarly, we recommend you review the background on ordinary differential equations in App. B either before or during the study of the main part. An intuitive approach to understanding differential equations and formal definitions of their semantics will be given throughout the text. Logic itself is also explained and illustrated intuitively during the main chapters, but some readers may also find it helpful to refresh, update, or learn about the basics of first-order logic from App. A before proceeding to the main part.

While there is a lot of flexibility in the reading sequence of the chapters, we strongly recommend you study the logical foundations of hybrid systems analysis in Chap. 2 of Part I before reading any other chapter of Parts I–III. Some more advanced sections in the applications in Part III also depend on the theory of differential invariants that is developed together with other extensions in Chap. 3.

Appendix C shows a formal relation of hybrid automata with hybrid programs. Appendix D gives more detail on the implementation of the approach put forth in this book in the verification tool KeYmaera. It also presents a survey of computational techniques for handling real arithmetic. Both App. C and D can be read as needed, after studying the introductory material and notions in Chap. 2. The most important formation rules for the logic and proof rules for the calculi are summarised at the end of the book.

Online Material for This Book

The Web page for this book provides online material, including the verification tool KeYmaera that implements our logical analysis approach for hybrid systems. We also provide slide material for parts of this book, an online tutorial for KeYmaera, and several KeYmaera problem files for examples from this book, including train and air traffic control studies. The book Web page is at the following URL:

<http://symbolaris.com/lahs/>

Acknowledgements

This book is based on my Ph.D. thesis and would not have been possible without the support of the PIs and collaborators on the projects that I have been working on. My sincere thanks go to Prof. Ernst-Rüdiger Olderog for his excellent advice and support, and for giving me the opportunity to work in one of the most fascinating areas of science in a group with a friendly and productive atmosphere. My advisor, Prof. Olderog, and the Director of AVACS, Prof. Werner Damm, both deserve my highest gratitude, not only for their continuous support and for their faith, but also for allowing me the freedom to pursue my own research ambitions in the stimulating context of the AVACS project (“Automatic Verification and Analysis of Complex Systems”). Ultimately, this made it possible for me to develop the logic and verification approach presented in this book.

I want to thank the external referees of my Ph.D. thesis, Prof. Tobias Nipkow from the Technical University of Munich and Prof. George J. Pappas from the University of Pennsylvania. It is an honour for me that they were willing to invest their valuable time and effort in the careful reviewing of my thesis. In fact, I am thankful to all members of my Ph.D. committee, Werner Damm, Ernst-Rüdiger Olderog, George J. Pappas, Tobias Nipkow, and Hardi Hungar for fruitful discussions and for the highest support they offered for my work.

I am especially grateful to Prof. Edmund M. Clarke, who invited me to Carnegie Mellon University several times, for his support, interest, and collaboration, and for sharing with me parts of his huge knowledge in all areas of formal methods. I further want to acknowledge the help by Prof. Peter H. Schmitt from the University of

Karlsruhe (TH), Profs. Bernhard Beckert and Ulrich Furbach from the University of Koblenz-Landau, Prof. Reiner Hähnle from the Chalmers University of Technology, Gothenburg, Sweden, Profs. Edmund M. Clarke and Frank Pfenning from Carnegie Mellon University, and Prof. Rajeev Goré from the Australian National University, Canberra, at various stages of my career.

I want to thank the program committee of the TABLEAUX 2007 conference for selecting my first paper on differential dynamic logic for the Best Paper Award, the first award at any TABLEAUX conference. This recognition has encouraged me to continue pursuing my research direction, which ultimately led to the results described in this book. I also thank the program committee of the FM 2009 conference for selecting my paper on formal verification of curved flight collision avoidance maneuvers for the Best Paper Award. I am very grateful to the ACM Doctoral Dissertation Award committee for honoring my Ph.D. thesis with the 2009 ACM Doctoral Dissertation Honorable Mention Award.

I am truly thankful to my colleagues at Carnegie Mellon University for their encouraging feedback about my work and for the friendly and constructive atmosphere at CMU. For many fruitful discussions I thank my colleagues and friends from Oldenburg, Ingo Brückner, Henning Dierks, Johannes Faber, Sibylle Fröschle, Jochen Hoenicke, Stephanie Kemper, Roland Meyer, Michael Möller, Jan-David Quesel, Tim Strazny, and especially my office mate Andreas Schäfer. Ernst-Rüdiger Olderog, Johannes Faber, Ingo Brückner, Roland Meyer, Henning Dierks, Silke Wagner, Nicole Betz, Alex Donzé, and especially Andreas Schäfer also deserve credit for proofreading some of my earlier papers, which formed the basis for this book. I also acknowledge Andreas Schäfer's helpful feedback from proofreading parts of this book. I appreciate the feedback of my students on this book.

Furthermore, I thank Jan-David Quesel for writing a Master's thesis under my supervision and for his invaluable support with the implementation of the verification tool KeYmaera based on the techniques that I present in this book and in prior publications. I also thank him for help with the experiments and ETCS. I am also thankful for indispensable and reliable help from Richard Bubel and Philipp Rümmer with the implementation internals of the KeY basis. I thank the whole KeY team for providing the impressive Java verification tool KeY as a basis for our implementation of KeYmaera.

For help with the book process, I thank Ronan Nugent from Springer.

Especially, I thank my parents, Rudolf and Brigitte Platzer, and my sister, Julia, for their continuous support and encouragement, and I thank my wife, Nicole, for her true faith in me. She also deserves credit for her invaluable help with some of the illustrations in this book.

Funding

This research was partly supported by the German Research Council (DFG) under grant SFB/TR 14 AVACS (“Automatic Verification and Analysis of Complex Systems”, see <http://www.avacs.org>); a Transregional Collaborative Research

Center of the Max Planck Institute and the Universities of Oldenburg, Saarbrücken, and Freiburg in Germany, with associated cooperations with the University of Pennsylvania, ETH Zürich, and the Academy of Sciences of the Czech Republic. It was further supported partly by a research fellowship of the German Academic Exchange Service (DAAD) and by a research award of the Floyd und Lili Biava Stiftung. Some part of this work was also supported by the National Science Foundation under grant nos. CNS-0931985 and CNS-0926181, including the NSF Expedition on Computational Modeling and Analysis of Complex Systems (CMACS); see <http://cmacs.cs.cmu.edu> for more information.

The views and conclusions contained in this book are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution or government.

Further Sources

This book is based on several sources, most notably the author's Ph.D. thesis [236]. Chapter 2 is an extended version of an article in the Journal of Automated Reasoning [235] and also covers some material from previous work at TABLEAUX [231] and HSCC [232]. Chapter 3 is an extended version of an article in the Journal of Logic and Computation [237], to which we now add a relative completeness argument and prove that DAL is a conservative extension of the sublogic $d\mathcal{L}$. We further combine the solution-based techniques from Chap. 2 with differential induction-based techniques from Chap. 3 by introducing the new extension of differential monotonicity relaxations. Chapter 4 is a substantially extended version of a previous paper at LFCS [233], to which we now add a complete and more elegant calculus and provide a modular relative completeness proof.

In Chap. 5, we extend a previous paper at VERIFY [230] with more details on iterative background closure strategies, including experimental evaluation, and complement this proof technique with a new iterative inflation strategy. Chapter 6 is based on joint work with Edmund M. Clarke at CAV [239] and in Formal Methods in System Design [240].

Chapter 7 is a substantially revised and improved version of joint work with Jan-David Quesel at HSCC [243] with extensions from follow-up work [244]. Chapter 8 is a significantly improved and detailed case study developed on the basis of joint work with Edmund M. Clarke at HSCC [238] and CAV [239] with subsequent extensions at FM [241].

Appendix B summarises classical results from the theory of differential equations from the literature [297]. Finally, App. D uses a few excerpts from joint work with Jan-David Quesel at IJCAR [242], adding an overall discussion of the KeYmaera verification tool that implements the approach presented in this book. Appendix D also adds a thorough description of computational back-ends for real arithmetic, with extensions from joint work with Philipp Rümmer and Jan-David Quesel [246].

Contents

1	Introduction	1
1.1	Technical Context	4
1.1.1	Hybrid Systems	4
1.1.2	Model Checking	12
1.1.3	Deductive Verification	14
1.1.4	Compositional Verification	16
1.1.5	Lifting Quantifier Elimination	19
1.1.6	Differential Induction and Differential Strengthening	20
1.2	Related Work	21
1.3	Contributions	25
1.4	Structure of This Book	25
 Part I Logics and Proof Calculi for Hybrid Systems		31
2	Differential Dynamic Logic $d\mathcal{L}$	33
2.1	Introduction	34
2.1.1	Structure of This Chapter	35
2.2	Syntax	35
2.2.1	Terms	37
2.2.2	Hybrid Programs	41
2.2.3	Formulas	47
2.3	Semantics	49
2.3.1	Valuation of Terms	50
2.3.2	Valuation of Formulas	51
2.3.3	Transition Semantics of Hybrid Programs	54
2.4	Collision Avoidance in Train Control	61
2.5	Proof Calculus	64
2.5.1	Substitution	65
2.5.2	Proof Rules	76

2.5.3	Deduction Modulo with Invertible Quantifiers and Real Quantifier Elimination	88
2.5.3.1	Lifting Quantifier Elimination by Invertible Quantifier Rules	88
2.5.3.2	Admissibility in Invertible Quantifier Rules	91
2.5.3.3	Quantifier Elimination and Modalities	93
2.5.3.4	Global Invertible Quantifier Rules	93
2.5.4	Verification Example	94
2.6	Soundness	97
2.7	Completeness	101
2.7.1	Incompleteness	102
2.7.2	Relative Completeness	103
2.7.3	Characterising Real Gödel Encodings	105
2.7.4	Expressibility and Rendition of Hybrid Program Semantics	106
2.7.5	Relative Completeness of First-Order Assertions	109
2.7.6	Relative Completeness of the Differential Logic Calculus	113
2.8	Relatively Semidecidable Fragments	114
2.9	Train Control Verification	118
2.9.1	Finding Inductive Candidates	118
2.9.2	Inductive Verification	119
2.9.3	Parameter Constraint Discovery	120
2.10	Summary	122
3	Differential-Algebraic Dynamic Logic DAL	123
3.1	Introduction	124
3.1.1	Related Work	128
3.1.2	Structure of This Chapter	130
3.2	Syntax	130
3.2.1	Terms	132
3.2.2	Differential-Algebraic Programs	132
3.2.3	Formulas	139
3.3	Semantics	141
3.3.1	Transition Semantics of Differential-Algebraic Programs	141
3.3.2	Valuation of Formulas	145
3.3.3	Time Anomalies	145
3.3.4	Conservative Extension	147
3.4	Collision Avoidance in Air Traffic Control	148
3.4.1	Flight Dynamics	148
3.4.2	Differential Axiomatisation	149
3.4.3	Aircraft Collision Avoidance Manoeuvres	150
3.4.4	Tangential Roundabout Manoeuvre	151
3.5	Proof Calculus	152
3.5.1	Motivation	153
3.5.2	Derivations and Differentiation	154
3.5.3	Differential Reduction and Differential Elimination	160

3.5.4	Proof Rules	162
3.5.5	Deduction Modulo by Side Deduction	168
3.5.6	Differential Induction with Differential Invariants	170
3.5.7	Differential Induction with Differential Variants	181
3.6	Soundness	185
3.7	Restricting Differential Invariants	188
3.8	Differential Monotonicity Relaxations	189
3.9	Relative Completeness	193
3.10	Deductive Strength of Differential Induction	194
3.11	Air Traffic Control Verification	197
3.11.1	Characterisation of Safe Roundabout Dynamics	197
3.11.2	Tangential Entry Procedures	200
3.11.3	Discussion	201
3.12	Summary	201
4	Differential Temporal Dynamic Logic dTL	203
4.1	Introduction	204
4.1.1	Related Work	205
4.1.2	Structure of This Chapter	206
4.2	Syntax	206
4.2.1	Hybrid Programs	207
4.2.2	State and Trace Formulas	207
4.3	Semantics	210
4.3.1	Trace Semantics of Hybrid Programs	210
4.3.2	Valuation of State and Trace Formulas	213
4.3.3	Conservative Temporal Extension	215
4.4	Safety Invariants in Train Control	216
4.5	Proof Calculus	217
4.5.1	Proof Rules	218
4.5.2	Verification Example	221
4.6	Soundness	221
4.7	Completeness	223
4.7.1	Incompleteness	223
4.7.2	Relative Completeness	224
4.7.3	Expressibility and Rendition of Hybrid Trace Semantics	225
4.7.4	Modular Relative Completeness Proof	226
4.8	Verification of Train Control Safety Invariants	227
4.9	Liveness by Quantifier Alternation	228
4.10	Summary	230
Part II Automated Theorem Proving for Hybrid Systems		231
5	Deduction Modulo Real Algebra and Computer Algebra	233
5.1	Introduction	234

- 5.1.1 Related Work 234
- 5.1.2 Structure of This Chapter 235
- 5.2 Tableau Procedures Modulo 235
- 5.3 Nondeterminisms in Tableau Modulo 238
 - 5.3.1 Nondeterminisms in Branch Selection 238
 - 5.3.2 Nondeterminisms in Formula Selection 239
 - 5.3.3 Nondeterminisms in Mode Selection 240
- 5.4 Iterative Background Closure 243
- 5.5 Iterative Inflation 246
- 5.6 Experimental Results 248
- 5.7 Summary 251
- 6 Computing Differential Invariants as Fixed Points 253**
 - 6.1 Introduction 254
 - 6.1.1 Related Work 255
 - 6.1.2 Structure of This Chapter 256
 - 6.2 Inductive Verification by Combining Local Fixed Points 256
 - 6.2.1 Verification by Symbolic Decomposition 257
 - 6.2.2 Discrete and Differential Induction, Differential Invariants 258
 - 6.2.3 Flight Dynamics in Air Traffic Control 260
 - 6.2.4 Local Fixed-Point Computation for Differential Invariants 262
 - 6.2.5 Dependency-Directed Induction Candidates 263
 - 6.2.6 Global Fixed-Point Computation for Loop Invariants 265
 - 6.2.7 Interplay of Local and Global Fixed-Point Loops 268
 - 6.3 Soundness 269
 - 6.4 Optimisations 271
 - 6.4.1 Sound Interleaving with Numerical Simulation 271
 - 6.4.2 Optimisations for the Verification Algorithm 272
 - 6.5 Experimental Results 272
 - 6.6 Summary 273

Part III Case Studies and Applications in Hybrid Systems Verification 275

- 7 European Train Control System 277**
 - 7.1 Introduction 278
 - 7.1.1 Related Work 280
 - 7.1.2 Structure of This Chapter 281
 - 7.2 Parametric European Train Control System 281
 - 7.2.1 Overview of the ETCS Cooperation Protocol 281
 - 7.2.2 Formal Model of Fully Parametric ETCS 284
 - 7.3 Parametric Verification of Train Control 286
 - 7.3.1 Controllability Discovery 287
 - 7.3.2 Iterative Control Refinement 288

- 7.3.3 Safety Verification 291
- 7.3.4 Liveness Verification 293
- 7.3.5 Full Correctness of ETCS 294
- 7.4 Disturbance and the European Train Control System 295
 - 7.4.1 Controllability Discovery 296
 - 7.4.2 Iterative Control Refinement 298
 - 7.4.3 Safety Verification 298
- 7.5 Experimental Results 299
- 7.6 Summary 301
- 8 Air Traffic Collision Avoidance 303**
 - 8.1 Introduction 304
 - 8.1.1 Related Work 307
 - 8.1.2 Structure of This Chapter 308
 - 8.2 Curved Flight in Roundabout Manoeuvres 309
 - 8.2.1 Flight Dynamics 309
 - 8.2.2 Roundabout Manoeuvre Overview 310
 - 8.2.3 Compositional Verification Plan 311
 - 8.2.4 Tangential Roundabout Manoeuvre Cycles 312
 - 8.2.5 Bounded Control Choices 315
 - 8.2.6 Flyable Entry Procedures 315
 - 8.2.7 Bounded Entry Duration 318
 - 8.2.8 Safe Entry Separation 319
 - 8.3 Synchronisation of Roundabout Manoeuvres 322
 - 8.3.1 Successful Negotiation 322
 - 8.3.2 Safe Exit Separation 326
 - 8.4 Compositional Verification 328
 - 8.5 Flyable Tangential Roundabout Manoeuvre 329
 - 8.6 Experimental Results 331
 - 8.7 Summary 333
- 9 Conclusion 335**

Part IV Appendix 339

- A First-Order Logic and Theorem Proving 341**
 - A.1 Overview 341
 - A.2 Syntax 346
 - A.2.1 Terms 346
 - A.2.2 Formulas 347
 - A.3 Semantics 348
 - A.3.1 Valuation of Terms 349
 - A.3.2 Valuation of Formulas 349
 - A.4 Proof Calculus 350
 - A.4.1 Proof Rules 351

A.4.2	Proof Example: Ground Proving Versus Free-Variable Proving	354
A.5	Soundness	356
A.6	Completeness	356
A.7	Computability Theory and Decidability	357
B	Differential Equations	359
B.1	Ordinary Differential Equations	359
B.2	Existence Theorems	363
B.3	Existence and Uniqueness Theorems	364
B.4	Linear Differential Equations with Constant Coefficients	365
C	Hybrid Automata	369
C.1	Syntax and Traces of Hybrid Automata	369
C.2	Embedding Hybrid Automata into Hybrid Programs	371
D	KeYmaera Implementation	377
D.1	KeYmaera: A Hybrid Theorem Prover for Hybrid Systems	377
D.1.1	Structure of This Appendix	379
D.2	Computational Back-ends for Real Arithmetic	380
D.2.1	Real-Closed Fields	381
D.2.2	Semialgebraic Geometry and Cylindrical Algebraic Decom- position	383
D.2.3	Nullstellensatz and Gröbner Bases	386
D.2.4	Real Nullstellensatz	392
D.2.5	Positivstellensatz and Semidefinite Programming	394
D.3	Discussion	396
D.4	Performance Measurements	399
	References	401
	Index	415
	Operators and Proof Rules	423

List of Figures

1.1	European Train Control System	2
1.2	ETCS: discrete evolution of acceleration a , continuous evolution of velocity v and of position z over time t	3
1.3	Collision avoidance manoeuvres in air traffic control	3
1.4	Hybrid automaton for an (overly) simplified train control system	5
1.5	Hybrid automaton and hybrid program of a simple bouncing ball	7
1.6	Switching between two damped oscillators	8
1.7	Hybrid automaton for switching damped oscillators	9
1.8	Stable trajectory switching between two damped oscillators	9
1.9	Instable trajectory switching between two damped oscillators	10
1.10	Simple water tank system	11
1.11	Successive state space exploration in finite-state model checking	12
1.12	Failed hybrid automaton decomposition attempt	17
1.13	Successful hybrid program decomposition	18
1.14	Dependencies and suggested reading sequence of chapters and appendices	28
2.1	Hybrid program rendition of hybrid automaton for (overly) simplified train control	36
2.2	Parametric bouncing ball	45
2.3	Parametric bouncing ball (with abbreviations resolved)	46
2.4	Transition semantics of modalities in $d\mathcal{L}$ formulas	52
2.5	Transition semantics and example dynamics of hybrid programs	56
2.6	Continuous flow along differential equation $x' = \theta$ over time	57
2.7	Transition structure and transition example in (overly) simple train control	59
2.8	ETCS train coordination protocol using dynamic movement authorities	61
2.9	ETCS transition structure and various choices of speed regulation for train speed control	63
2.10	Application of simultaneous substitutions	65

2.11	Rule schemata of the free-variable calculus for differential dynamic logic	79
2.12	Correspondence of dynamic proof rules and transition semantics	83
2.13	Simple propositional example proof	87
2.14	Deduction modulo for analysis of MA violation in braking mode	89
2.15	Controllable region of ETCS dynamics	90
2.16	Deduction modulo for analysis of MA-safety in braking mode	90
2.17a	Wrong rearrangement with deduction modulo by invertible quantifiers	91
2.17b	Correct reintroduction order with deduction modulo by invertible quantifiers	91
2.18	Bouncing ball proof (no evolution domain)	95
2.19a	Unsound attempt of induction without universal closure \forall^α	95
2.19b	Correct use of induction with universal closure \forall^α , i.e., $\forall x$	95
2.20	Bouncing ball proof (with evolution domain)	97
2.21	Characterisation of \mathbb{N} as zeros of solutions of differential equations	103
2.22	Fractional encoding principle of \mathbb{R} -Gödel encoding by bit interleaving	105
2.23	FOD definition characterising Gödel encoding of \mathbb{R} -sequences in one real number	106
2.24	Explicit rendition of hybrid program transition semantics in FOD	107
2.25	Evolution domain checks along backwards flow over time	108
3.1	Controllability violated in the presence of disturbance	138
3.2	Differential state flow	143
3.3	Zeno system run	146
3.4	Aircraft dynamics	148
3.5	Reparametrise for differential axiomatisation	149
3.6	Flight manoeuvres for collision avoidance in air traffic control	151
3.7	Flight control with tangential roundabout collision avoidance manoeuvres	152
3.8	Vector field and a solution of a differential equation	153
3.9	Rule schemata of the proof calculus for differential-algebraic dynamic logic	164
3.10	Side deduction for quantifier elimination rules	164
3.11	Nested side deductions and differential variants for progress property	169
3.12	Differential invariants	171
3.13a	Cubic dynamics proof	172
3.13b	Cubic dynamics	172
3.14	Unsound restriction of differential invariance	173
3.15a	Restricting differential invariance	174
3.15b	Linear dynamics	174
3.16	Proof of MA-safety in braking mode with disturbance	176
3.17	Trajectory and evolution of a damped oscillator	177
3.18	Trajectory switching between two damped oscillators	178

3.19	Parametric switched damped oscillator system	178
3.20	Instable trajectory switching between two damped oscillators	179
3.21	Parametric switched damped oscillator proof	180
3.22	Differential variants	182
3.23a	Monotonically decreasing convergent counterexample	184
3.23b	Convergent descent dynamics	184
3.23c	Non-inductive property in convergent descent	184
3.24a	Counterexample of unbounded dynamics without Lipschitz continuity	184
3.24b	Explosive dynamics with limited duration of solutions	184
3.25	Differential induction splitting over disjunctions for negative equations	189
3.26a	Counterexample for disjunctive monotonicity	193
3.26b	Interrupted dynamics	193
3.27	Quadrant sign selection regions of differential invariant	196
3.28	Circular dependencies for differential strengthening	196
3.29	Tangential construction for characteristics of roundabout dynamics	198
4.1	Trace semantics of dTL formulas	214
4.2	ETCS train coordination protocol phases	216
4.3	Rule schemata of the proof calculus for temporal differential dynamic logic	218
4.4	Correspondence of temporal proof rules and trace semantics	219
4.5	Explicit rendition of hybrid program trace semantics in FOD	225
4.6	Transformation rules for alternating temporal path and trace quantifiers	229
5.1	Deductive, real algebraic, computer algebraic prover combination	236
5.2	Tableau procedure for differential dynamic logics	237
5.3	Nondeterminisms in the tableau procedure for differential dynamic logics	237
5.4	Computational distraction in quantifier elimination	240
5.5	Eager and lazy quantifier elimination in proof search space	241
5.6	A large subgoal of first-order real arithmetic during ETCS verification	242
5.7a	Proof strategy priorities	244
5.7b	Iterative background closure (IBC) proof strategy	244
5.8	Iterative background closure (IBC) algorithm schema	245
5.9	General and/or-branching in proof strategies for differential dynamic logics	245
5.10	Iterative inflation order (IIO) algorithm schema	247
6.1	d \mathcal{L} -based verification by symbolic decomposition	257
6.2	Aircraft dynamics	261
6.3	Fixed-point algorithm for differential invariants (<i>Differential Saturation</i>)	262

6.4	Differential dependencies and variable clusters of flight dynamics	264
6.5	Fixed-point algorithm for discrete loop invariants (loop saturation)	266
6.6	Hybrid program rendition of hybrid automaton for simple water tank	267
6.7	Interplay of local and global fixed-point verification loops during symbolic decomposition	268
6.8	Robustness in counterexamples	271
6.9	Flyable aircraft roundabout	272
7.1	ETCS train cooperation protocol phases and dynamic movement authorities	282
7.2	ETCS track profile	283
7.3	Formal model of parametric ETCS cooperation protocol (skeleton)	284
7.4	Transition structure of ETCS skeleton	286
7.5	Controllable region of ETCS	288
7.6	ETCS cooperation protocol refined with parameter constraints	291
7.7	Proof sketch for ETCS safety	292
7.8	Controllability region changes in the presence of disturbance	295
7.9	Proof of ETCS controllability despite disturbance	297
7.10	Parametric ETCS cooperation protocol with disturbances	299
7.11	Parametric ETCS cooperation protocol with disturbances (full in- stantiation)	300
8.1	Evolution of collision avoidance manoeuvres in air traffic control	304
8.2	Non-flyable straight-line manoeuvre with instant turns	305
8.3	Flyable aircraft roundabout	309
8.4	Flight dynamics	309
8.5	Protocol cycle and construction of flyable roundabout manoeuvre	310
8.6	Non-flyable tangential roundabout collision avoidance manoeuvre NTRM	312
8.7	Tangential configuration \mathcal{T}	313
8.8	Flyable aircraft roundabout (multiple aircraft)	314
8.9	Tangential roundabout collision avoidance manoeuvre (four aircraft)	314
8.10	Flyable entry characteristics	316
8.11	Entry separation by bounded nondeterministic overapproximation	320
8.12	Some mutually agreeable negotiation choices for aircraft	323
8.13	Far separation for mutually agreeable negotiation choices	325
8.14a	Exit ray separation	327
8.14b	Incompatible exit rays	327
8.15	Flight control with flyable tangential roundabout collision avoidance	329
8.16	Verification loop for flyable tangential roundabout manoeuvres	330
8.17	Flight control with FTRM (synchronous instantiation)	332
9.1	Topics contributing to the logical analysis of hybrid systems	336
A.1	Rule schemata of the sequent calculus for first-order logic	352
A.2a	Ground proof example	354

A.2b	Free-variable proof example	354
A.3	Wrong proof attempt in first-order logic	355
B.1	Vector field and a solution of a differential equation	360
C.1	Hybrid automaton and corresponding hybrid program	370
C.2a	Hybrid automaton for water tank	373
C.2b	Hybrid program for water tank	373
C.3	Parametric bouncing ball	374
D.1	Architecture and plug-in structure of the KeYmaera prover	378
D.2	Screenshot of the KeYmaera user interface	379
D.3	KeYmaera proof strategy options	380
D.4	Projection of semialgebraic sets and quantifier elimination	384
D.5	Rule schemata of Gröbner calculus rules	389
D.6	Some algebraic varieties generated by one polynomial equation in two variables	391
D.7	Example proof using the real Nullstellensatz	393
D.8	Rule schema of Positivstellensatz calculus rule	395
D.9	Example proof using the Positivstellensatz	395

List of Tables

2.1	Statements and effects of hybrid programs (HPs)	42
2.2	Statements and control structures definable with hybrid programs . .	44
2.3	Operators and meaning in differential dynamic logic ($d\mathcal{L}$)	47
3.1	Comparison of DAL with DA-programs versus $d\mathcal{L}$ with hybrid programs	127
3.2	Statements and effects of differential-algebraic programs	137
3.3	Classification of differential-algebraic programs and correspondence to dynamical systems	139
3.4	Operators and meaning in differential-algebraic dynamic logic (DAL)	140
3.5	Embedding hybrid programs as DA-programs	147
4.1	Operators and meaning in differential temporal dynamic logic (dTL)	208
5.1	Experimental results for proof strategies (with standalone QE) I . . .	249
5.2	Experimental results for proof strategies (with standalone QE) II . . .	249
5.3	Experimental results for proof strategies (no standalone QE) I	250
5.4	Experimental results for proof strategies (no standalone QE) II	250
6.1	Experimental results for differential invariants as fixed points	273
7.1	Experimental results for the European Train Control System	300
8.1	Verification loop properties for flyable tangential roundabout manoeuvres	330
8.2	Experimental results for air traffic control (initial timeout = 10s) . . .	331
8.3	Experimental results for air traffic control (initial timeout = 4s) . . .	333
A.1	Intuitive meaning of logical operators in first-order logic	343

List of Theorems

L 2.1	Uniqueness	57
L 2.2	Substitution Lemma	70
L 2.3	Substitution property	75
L 2.4	Substitutions preserve validity	76
L 2.5	Quantifier elimination lifting	92
L 2.6	Coincidence lemma	93
T 2.1	Soundness of \mathbf{dL}	98
T 2.2	Incompleteness of \mathbf{dL}	102
T 2.3	Relative completeness of \mathbf{dL}	104
L 2.7	\mathbb{R} -Gödel encoding	105
L 2.8	Hybrid program rendition	106
L 2.9	\mathbf{dL} Expressibility	108
L 2.10	Derivability of sequents	109
L 2.11	Generalisation	110
P 2.1	Relative completeness of first-order safety	111
P 2.2	Relative completeness of first-order liveness	112
T 2.4	Relatively semidecidable fragment	114
L 2.12	Uniform Skolem symbols	115
P 3.1	Conservative extension	147
L 3.1	Derivation lemma	156
L 3.2	Differential substitution property	158
L 3.3	Differential transformation principle	158
L 3.4	Differential inequality elimination	161
L 3.5	Differential equation normalisation	161
L 3.6	Differential weakening	175
L 3.7	Closure properties of differential invariants	181
T 3.1	Soundness of DAL	185
P 3.2	Open differential induction	188
P 3.3	Differential monotonicity	191
T 3.2	Relative completeness of DAL	193
P 3.4	Equational deductive power	194

T 3.3	Deductive power	194
T 3.4	Safety of tangential roundabout manoeuvre	199
P 3.5	External separation of roundabout manoeuvres	200
P 4.1	Conservative temporal extension	215
L 4.1	Trace relation	215
T 4.1	Soundness of dTL	221
T 4.2	Incompleteness of dTL	223
T 4.3	Relative completeness of dTL	224
L 4.2	Hybrid program trace rendition	225
L 4.3	dTL Expressibility	225
P 4.2	Local soundness for temporal quantifier alternation	229
P 6.1	Principle of differential induction	260
P 6.2	Differential saturation	262
P 6.3	Loop saturation	265
T 6.1	Soundness of fixed-point verification algorithm	269
L 7.1	Principle of separation by movement authorities	282
P 7.1	Controllability	288
P 7.2	RBC preserves train controllability	289
P 7.3	Reactivity of ETCS	290
P 7.4	Reactivity constraint	290
P 7.5	Safety of ETCS	291
P 7.6	Liveness of ETCS	293
T 7.1	Correctness of ETCS cooperation protocol	294
P 7.7	Controllability despite disturbance	296
P 7.8	Reactivity constraint despite disturbance	298
P 7.9	Safety despite disturbance	298
T 8.1	Safety property of flyable tangential roundabouts	331
T A.1	Soundness of FOL	356
T A.2	Completeness of FOL	356
T B.1	Existence theorem of Peano	363
T B.2	Uniqueness theorem of Picard-Lindelöf	364
P B.1	Continuation of solutions	365
P B.2	Linear systems with constant coefficients	365
P C.1	Hybrid automata embedding	371
T D.1	Tarski-Seidenberg principle	383
T D.2	Semialgebraic sets	383
T D.3	Hilbert's basis theorem	388
P D.1	Soundness of Gröbner basis rules	390
T D.4	Hilbert's Nullstellensatz	391
T D.5	Real Nullstellensatz for real-closed fields	392
T D.6	Positivstellensatz for real-closed fields	394