

*Commenced Publication in 1973*

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

## Editorial Board

David Hutchison, UK

Josef Kittler, UK

Friedemann Mattern, Switzerland

Moni Naor, Israel

Bernhard Steffen, Germany

Doug Tygar, USA

Takeo Kanade, USA

Jon M. Kleinberg, USA

John C. Mitchell, USA

C. Pandu Rangan, India

Demetri Terzopoulos, USA

Gerhard Weikum, Germany

## Formal Methods

Subline of Lectures Notes in Computer Science

## Subline Series Editors

Ana Cavalcanti, *University of York, UK*

Marie-Claude Gaudel, *Université de Paris-Sud, France*

## Subline Advisory Board

Manfred Broy, *TU Munich, Germany*

Annabelle McIver, *Macquarie University, Sydney, NSW, Australia*

Peter Müller, *ETH Zurich, Switzerland*

Erik de Vink, *Eindhoven University of Technology, The Netherlands*

Pamela Zave, *AT&T Laboratories Research, Bedminster, NJ, USA*


More information about this series at <http://www.springer.com/series/7408>


Nadia Polikarpova · Steve Schneider (Eds.)

# Integrated Formal Methods

13th International Conference, IFM 2017  
Turin, Italy, September 20–22, 2017  
Proceedings

*Editors*

Nadia Polikarpova   
Massachusetts Institute of Technology  
Cambridge, MA  
USA

Steve Schneider   
University of Surrey  
Guildford  
UK

ISSN 0302-9743                      ISSN 1611-3349 (electronic)  
Lecture Notes in Computer Science  
ISBN 978-3-319-66844-4              ISBN 978-3-319-66845-1 (eBook)  
DOI 10.1007/978-3-319-66845-1

Library of Congress Control Number: 2017952382

LNCS Sublibrary: SL2 – Programming and Software Engineering

© Springer International Publishing AG 2017

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 Springer Nature  
The registered company is Springer International Publishing AG  
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

# Preface

Applying formal methods may involve the usage of different formalisms and different analysis techniques to validate a system, either because individual components are most amenable to one formalism or technique, because one is interested in different properties of the system, or simply to cope with the sheer complexity of the system. The iFM conference series seeks to further research into hybrid approaches to formal modeling and analysis; i.e., the combination of (formal and semi-formal) methods for system development, regarding both modeling and analysis. The conference covers all aspects from language design through verification and analysis techniques to tools and their integration into software engineering practice.

These proceedings document the outcome of the 13th International Conference on Integrated Formal Methods, iFM 2017, on recent developments toward this goal. The conference was held in Turin, Italy, on September 20–22, 2017, hosted by the University of Turin. Previous editions of iFM were held in York, UK (1999), Schloss Dagstuhl, Germany (2000), Turku, Finland (2002), Kent, UK (2004), Eindhoven, The Netherlands (2005), Oxford, UK (2007), Düsseldorf, Germany (2009), Nancy, France (2010), Pisa, Italy (2012), Turku, Finland (2013), Bertinoro, Italy (2014), and Reykjavik, Iceland (2016).

The conference received 61 submissions from authors in 24 countries. Papers were submitted in four categories: research papers, case study papers, regular tool papers, and tool demonstration papers. All papers were reviewed by at least three members of the Program Committee. After careful deliberation, the Program Committee selected 28 papers for presentation.

Among these papers, the Program Chairs, in consultation with the Program Committee, have selected winners for two awards. The contribution “Triggerless Happy: Intermediate Verification with a First-Order Prover” by YuTing Chen and Carlo A. Furia received the *Best Paper Award*. The contribution “Complexity Analysis for Java with AProVE” by Florian Frohn and Jürgen Giesl received the *Best Tool Paper Award*. Each award was accompanied by a EUR 500 prize, generously provided by Springer.

In addition to the 28 peer-reviewed papers, this volume contains contributions from each of the three invited keynote speakers:

- Jane Hillston (University of Edinburgh, UK): “Integrating Inference with Stochastic Process Algebra Models”
- André Platzer (Carnegie Mellon University, USA): “Logic & Proofs for Cyber-Physical Systems with KeYmaera X”
- Martin Vechev (ETH Zurich, Switzerland): “Machine Learning for Programming”

Invited presentations are always the highlights of a conference; these contributions are therefore gratefully acknowledged.

iFM was accompanied by a PhD Symposium, organized by the symposium chairs, Erika Ábrahám (RWTH Aachen University, Germany) and S. Lizeth Tapia Tarifa

(University of Oslo, Norway), as well as the following satellite events, managed by the workshop chairs, Wolfgang Ahrendt (Chalmers University of Technology, Sweden) and Michael Lienhardt (University of Turin, Italy):

- International Workshop on Formal Methods for Industrial Critical Systems and Automated Verification of Critical Systems (FMICS-AVoCS)
- Workshop on Architectures, Languages and Paradigms for IoT (ALP4IoT)
- Workshop on Actors and Active Objects (WAO)
- Workshop on Formal Verification of Autonomous Vehicles (FVAV)
- Second International Workshop on Pre- and Post-Deployment Verification Techniques (PrePost)
- Second International Workshop on Verification and Validation of Cyber-Physical Systems (V2CPS)

The conference would not have been possible without the enthusiasm and dedication of the iFM general chair, Ferruccio Damiani, and the support of the Computer Science Department at the University of Turin, Italy. The EasyChair conference management system was invaluable for conducting the peer review process and preparing the proceedings. Conferences like iFM rely on the willingness of experts to serve on the Program Committee; their professionalism and their helpfulness was exemplary. Finally, we would like to thank all the authors for their submissions, their willingness to continue improving their papers, and their presentations!

July 2017

Nadia Polikarpova  
Steve Schneider

# Organization

## General Chair

Ferruccio Damiani      University of Turin, Italy

## Program Chairs

Nadia Polikarpova      MIT, USA  
Steve Schneider      University of Surrey, UK

## Steering Committee

Erika Ábrahám      RWTH Aachen University, Germany  
Elvira Albert      Complutense University of Madrid, Spain  
John Derrick      University of Sheffield, UK  
Marieke Huisman      University of Twente, Netherlands  
Einar Broch Johnsen      University of Oslo, Norway  
Dominique Mery      Université de Lorraine, LORIA, France  
Luigia Petre      Åbo Akademi University, Finland  
Steve Schneider      University of Surrey, UK  
Emil Sekerinski      McMaster University, Canada  
Marjan Sirjani      University of Reykjavik, Iceland  
Helen Treharne      University of Surrey, UK  
Heike Wehrheim      University of Paderborn, Germany

## Program Committee

Erika Ábrahám      RWTH Aachen University, Germany  
Elvira Albert      Complutense University of Madrid, Spain  
Oana Andrei      University of Glasgow, UK  
Borzoo Bonakdarpour      McMaster University, Canada  
Barbora Buhnova      Masaryk University, Czech Republic  
David Cok      GrammaTech, USA  
John Derrick      University of Sheffield, UK  
Yliès Falcone      Univ. Grenoble Alpes, Inria, France  
Leo Freitas      Newcastle University, UK  
Carlo A. Furia      Chalmers University of Technology, Sweden  
Jan Friso Groote      Eindhoven University of Technology, Netherlands  
Reiner Hähnle      Technical University of Darmstadt, Germany  
Ian J. Hayes      University of Queensland, Australia  
Marieke Huisman      University of Twente, Netherlands  
Rajeev Joshi      NASA Jet Propulsion Laboratory, USA

Laura Kovács	Vienna University of Technology, Austria
Juliana Küster Filipe Bowles	University of St. Andrews, UK
Axel Legay	IRISA/Inria Rennes, France
K. Rustan M. Leino	Microsoft Research, USA
Gerald Lüttgen	University of Bamberg, Germany
Dominique Mery	Université de Lorraine, LORIA, France
Stefan Mitsch	Carnegie Mellon University, USA
Rosemary Monahan	Maynooth University, Ireland
Luigia Petre	Åbo Akademi University, Finland
Adrian Riesco	Universidad Complutense de Madrid, Spain
Gerhard Schellhorn	Universität Augsburg, Germany
Gerardo Schneider	Chalmers University of Technology, University of Gothenburg, Sweden
Emil Sekerinski	McMaster University, Canada
Graeme Smith	University of Queensland, Australia
Martin Steffen	University of Oslo, Norway
Armando Tacchella	Università di Genova, Italy
Helen Treharne	University of Surrey, UK
Mark Utting	University of the Sunshine Coast, Australia
Frits Vaandrager	Radboud University Nijmegen, Netherlands
Heike Wehrheim	University of Paderborn, Germany
Kirsten Winter	University of Queensland, Australia

## Additional Reviewers

Bodenmueller, Stefan	Flores-Montoya, Antonio	Pinisetty, Srinivas
Brett, Noel	Hallgren, Per	Schlaipfer, Matthias
Bubel, Richard	Isabel, Miguel	Siddique, Umair
Burton, Eden	Jakse, Raphaël	Suda, Martin
Camilleri, John J.	Kamburjan, Eduard	Talebi, Mahmoud
Caminati, Marco B.	Kragl, Bernhard	Töws, Manuel
Colvin, Robert	Modesti, Paolo	Traonouez, Louis-Marie
Do, Quoc Huy	Mueller, Andreas	Travkin, Oleg
Doménech, Jesús J.	Nazarpour, Hosein	de Vink, Erik
El-Hokayem, Antoine	Neele, Thomas	Yang, Fei
Enescu, Mike	Pagnin, Elena	Zanardini, Damiano
Fendrich, Sascha	Pfähler, Jörg	Zantema, Hans



# **Invited Talks**

# Integrating Inference into Stochastic Process Algebra Models

Jane Hillston

LFCS, School of Informatics, University of Edinburgh  
jane.hillston@ed.ac.uk

Stochastic process algebras emerged in the early 1990s as a quantitative formal method. By incorporating information about probabilities and timing into a classical process algebra, it was possible to build models which allowed quantitative aspects of behaviour such as performance, reliability and availability to be evaluated in addition to qualitative aspects such as liveness and safety. Thus it became possible to answer questions such as the expected time until a failure in the system, or the proportion messages that are successfully delivered within 10 seconds. The language is equipped with a structured operational semantics giving rise to a labelled transition system that can be interpreted as a continuous time Markov chain. This class of stochastic processes is widely used in quantitative modelling and many efficient analysis techniques are available. Moreover the formality and structure of the process algebra has allowed new decompositions and approximations to be defined at the language level and automatically applied.

However one of the drawbacks of the stochastic process algebra approach is that the quantitative analysis of the model is dependent on the accuracy of the parameters used to capture the timings and probabilities that influence behaviour within the system. In some application domains this data can be obtained from monitoring or logging software, systems specifications etc. But in others, such as systems biology, not all aspects of behaviour are accessible to measurement and it can be very difficult to arrive at accurate parameters for the models.

Thus in recent years we have developed a stochastic process algebra, ProPPA, which allows parameters within the model to be left uncertain, specified by a distribution rather than a concrete value. Thus a ProPPA model describes not a single model, but a family of models, each associated with a probability that it is a good representation of the system. Moreover when evidence about the behaviour of the system is available, the language supports inference techniques from machine learning, which allow us to refine the uncertainty and generate a new family of models with different probabilities. The range of possible quantitative behaviours can be derived from the family of models together with an estimate of their likelihood.

Thus ProPPA, Probabilistic Programming Process Algebra, is a stochastic process algebra that combines elements of the data-driven modelling approach adopted in machine learning, with a more mechanistic modelling style from formal methods. Since

different inference techniques are suited to different model characteristics, the ProPPA tool suite offers a modular approach with a number of different inference techniques which can be used to refine the estimate of the parameters of the model and therefore the possible quantitative behaviours that may be exhibited.

# Logic & Proofs for Cyber-Physical Systems with KeYmaera X

André Platzer

Computer Science Department, Carnegie Mellon University, Pittsburgh, USA  
aplatzer@cs.cmu.edu

## 1 Abstract of Invited Talk

Cyber-physical systems (CPS) combine cyber aspects such as communication and computer control with physical aspects such as movement in space, which arise frequently in many safety-critical application domains, including aviation, automotive, railway, and robotics [1, 2, 4–6, 8, 11, 16, 17, 24–28, 40, 42–44]. But how can we ensure that these systems are guaranteed to meet their design goals, e.g., that an aircraft will not crash into another one?

Borrowing from an invited paper at IJCAR [36] to which we refer for more detail, this talk will highlight some of the most fascinating aspects of cyber-physical systems and their dynamical systems models, such as hybrid systems that combine discrete transitions and continuous evolution along differential equations. Because of the impact that they can have on the real world, CPSs deserve proof as safety evidence.

Multi-dynamical systems understand complex systems as a combination of multiple elementary dynamical aspects [33], which makes them natural mathematical models for CPS, since they tame their complexity by compositionality. The family of differential dynamic logics [28–35, 37] achieves this compositionality by providing compositional logics, programming languages, and reasoning principles for CPS. Differential dynamic logics, as implemented in the theorem prover KeYmaera X [7], have been instrumental in verifying many applications, including the Airborne Collision Avoidance System ACAS X [9], the European Train Control System ETCS [39], automotive systems [13, 14, 20], aircraft roundabout maneuvers [38], mobile robot navigation [18, 19], and a surgical robot system for skull-base surgery [10].

In addition to serving as a basis for additional formal verification results in different CPS application domains, each of those case studies are chosen to demonstrate how characteristically new features can be verified in practice. Safety, controllability, reactivity, and liveness properties for the double integrator dynamics interacting with different discrete components are the basis for ETCS verification [39]. Combinations with distributed systems and communication systems are emphasized elsewhere

---

This talk is based on an overview of logic and proofs for cyber-physical systems from IJCAR [36] to which we refer for more details. The talk is augmented with more detail on the new theorem prover KeYmaera X, which is at <http://keymaeraX.org/>. This material is based upon work supported by the National Science Foundation under NSF CAREER Award CNS-1054246.

[13, 14, 20]. How safety properties of CPS with unsolvable dynamics can be verified rigorously is showcased for aircraft with fixed ground speed [38] and for mobile ground robot navigation with acceleration/braking [18, 19]. High precision results in the safe handling of data structures for an unbounded number of obstacles are showcased in medical robotics [10]. Systems whose decisions are based on table lookups from a machine-learned value table are studied in the context of elaborate characterizations of the safe region of the high-level vertical motion of aircraft [9]. The ACAS X results are also of interest for characterizations of last-resort safety, i.e., to restrict intervention to when the last chance for a corrective safety action has come.

The KeYmaera X prover implements a uniform substitution calculus for differential dynamic logic  $d\mathcal{L}$  [35], which enables a prover with a very small soundness-critical core of just about 1 700 LOC of Scala [7]. To achieve high levels of confidence, this uniform substitution calculus has been cross-verified both in the Isabelle/HOL and in the Coq theorem provers [3]. Verification results about CPS models transfer to CPS implementations when generating provably correct runtime monitors with the ModelPlex approach [21], which is also implemented as a proof tactic in KeYmaera X. That approach makes it possible to rigorously develop correct CPS controllers for CPS models with a provable link to the safety monitors in the system implementation. The use of components for hybrid systems has been explored as well [15, 22, 23], which make it possible to benefit from safety proofs about components and inherit safety proofs for a compound system for free (under certain compatibility conditions). While differential dynamic logics are already inherently compositional for each of their composition operators, component notions add additional structuring principles for bigger pieces and provide simple safety notions for components. In order to bootstrap such a component approach without having to enlarge the small soundness-critical core of KeYmaera X, the safety of the composite is proved automatically by a KeYmaera X tactic from correctness proofs about its components [23].

More technical overviews are available in the literature [29, 33, 36, 41].

## References

1. Alur, R.: Principles of Cyber-Physical Systems. MIT Press (2015)
2. Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T.A., Ho, P.H., Nicollin, X., Olivero, A., Sifakis, J., Yovine, S.: The algorithmic analysis of hybrid systems. *Theor. Comput. Sci.* **138**(1), 3–34 (1995)
3. Bohrer, B., Rahli, V., Vukotic, I., Völpl, M., Platzer, A.: Formally verified differential dynamic logic. In: Bertot, Y., Vafeiadis, V. (eds.) *Certified Programs and Proofs - 6th ACM SIGPLAN Conference. CPP 2017*, Paris, France, January 16–17, 2017, pp. 208–221. ACM (2017)
4. Clarke, E.M., Emerson, E.A., Sifakis, J.: Model checking: algorithmic verification and debugging. *Commun. ACM* **52**(11), 74–84 (2009)
5. Davoren, J.M., Nerode, A.: Logics for hybrid systems. *IEEE* **88**(7), 985–1010 (2000)
6. Doyen, L., Frehse, G., Pappas, G.J., Platzer, A.: Verification of hybrid systems. In: Clarke, E. M., Henzinger, T.A., Veith, H. (eds.) *Handbook of Model Checking*. Springer (2017)

7. Fulton, N., Mitsch, S., Quesel, J.D., Völpl, M., Platzer, A.: KeYmaera X: An axiomatic tactical theorem prover for hybrid systems. In: Felty, A.P., Middeldorp, A. (eds.) CADE-25. LNAI, vol. 9195, pp. 527–538. Springer, Switzerland (2015)
8. Henzinger, T.A., Sifakis, J.: The discipline of embedded systems design. *Computer* **40**(10), 32–40 (2007)
9. Jeannin, J., Ghorbal, K., Kouskoulas, Y., Schmidt, A., Gardner, R., Mitsch, S., Platzer, A.: A formally verified hybrid system for safe advisories in the next-generation airborne collision avoidance system. *STTT* (2016)
10. Kouskoulas, Y., Renshaw, D.W., Platzer, A., Kazanzides, P.: Certifying the safe design of a virtual fixture control algorithm for a surgical robot. In: Belta, C., Ivancic, F. (eds.) HSCC, pp. 263–272. ACM (2013)
11. Larsen, K.G.: Verification and performance analysis for embedded systems. In: Chin, W., Qin, S. (eds.) Third IEEE International Symposium on Theoretical Aspects of Software Engineering. TASE 2009, 29–31 July 2009, Tianjin, China, pp. 3–4. IEEE Computer Society (2009)
12. Proceedings of the 27th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2012, Dubrovnik, Croatia, June 25–28, 2012. IEEE (2012)
13. Loos, S.M., Platzer, A., Nistor, L.: Adaptive cruise control: Hybrid, distributed, and now formally verified. In: Butler, M., Schulte, W. (eds.) FM 2011. LNCS, vol. 6664, pp. 42–56. Springer, Heidelberg (2011)
14. Loos, S.M., Witmer, D., Steenkiste, P., Platzer, A.: Efficiency analysis of formally verified adaptive cruise controllers. In: Hegyi, A., Schutter, B.D. (eds.) ITSC, pp. 1565–1570 (2013)
15. Lunel, S., Boyer, B., Talpin, J.P.: Compositional proofs in differential dynamic logic. In: ACSD (2017)
16. Lunze, J., Lamnabhi-Lagarrigue, F. (eds.): Handbook of Hybrid Systems Control: Theory, Tools, Applications. Cambridge University Press (2009)
17. Maler, O.: Control from computer science. *Ann. Rev. Control* **26**(2), 175–187 (2002)
18. Mitsch, S., Ghorbal, K., Platzer, A.: On provably safe obstacle avoidance for autonomous robotic ground vehicles. In: Newman, P., Fox, D., Hsu, D. (eds.) Robotics: Science and Systems (2013)
19. Mitsch, S., Ghorbal, K., Vogelbacher, D., Platzer, A.: Formal verification of obstacle avoidance and navigation of ground robots (2016). [CoRR abs/1605.00604](https://arxiv.org/abs/1605.00604)
20. Mitsch, S., Loos, S.M., Platzer, A.: Towards formal verification of freeway traffic control. In: Lu, C. (ed.) ICCPS, pp. 171–180. IEEE (2012)
21. Mitsch, S., Platzer, A.: ModelPlex: Verified runtime validation of verified cyber-physical system models. *Form. Methods Syst. Des.* **49**(1), 33–74 (2016). Special issue of selected papers from RV’14
22. Müller, A., Mitsch, S., Retschitzegger, W., Schwinger, W., Platzer, A.: A component-based approach to hybrid systems safety verification. In: Ábrahám, E., Huisman, M. (eds.) IFM 2016. LNCS, vol. 9681, pp. 441–456. Springer, Switzerland (2016)
23. Müller, A., Mitsch, S., Retschitzegger, W., Schwinger, W., Platzer, A.: Change and delay contracts for hybrid system component verification. In: Huisman, M., Rubin, J. (eds.) FASE 2017. LNCS, vol. 10202, pp. 134–151. Springer, Germany (2017)
24. Nerode, A.: Logic and control. In: Cooper, S.B., Löwe, B., Sorbi, A. (eds.) CiE 2007. LNCS, vol. 4497, pp. 585–597. Springer, Heidelberg (2007)
25. Nerode, A., Kohn, W.: Models for hybrid systems: Automata, topologies, controllability, observability. In: Grossman, R.L., Nerode, A., Ravn, A.P., Rischel, H. (eds.) Hybrid Systems. LNCS, vol. 736, pp. 317–356. Springer (1992)
26. NITRD CPS Senior Steering Group: CPS vision statement. NITRD (2012)

27. Pappas, G.J.: Wireless control networks: modeling, synthesis, robustness, security. In: Caccamo, M., Frazzoli, E., Grosu, R. (eds.) *Proceedings of the 14th ACM International Conference on Hybrid Systems: Computation and Control, HSCC 2011, Chicago, IL, USA, April 12–14, 2011*, pp. 1–2. ACM (2011)
28. Platzer, A.: Differential dynamic logic for hybrid systems. *J. Autom. Reas.* **41**(2), 143–189 (2008)
29. Platzer, A.: *Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics*. Springer, Heidelberg (2010)
30. Platzer, A.: Stochastic differential dynamic logic for stochastic hybrid programs. In: Bjørner, N., Sofronie-Stokkermans, V. (eds.) *CADE 2011. LNCS*, vol. 6803, pp. 431–445. Springer, Heidelberg (2011)
31. Platzer, A.: A complete axiomatization of quantified differential dynamic logic for distributed hybrid systems. *Log. Meth. Comput. Sci.* **8**(4), 1–44 (2012). Special issue for selected papers from CSL’10
32. Platzer, A.: The complete proof theory of hybrid systems. In: *LICS 2012*, pp. 541–550 (2012)
33. Platzer, A.: Logics of dynamical systems. In: *LICS 2012*, pp. 13–24 (2012)
34. Platzer, A.: Differential game logic. *ACM Trans. Comput. Log.* **17**(1), 1:1–1:51 (2015)
35. Platzer, A.: A complete uniform substitution calculus for differential dynamic logic. *J. Autom. Reas.* (2016)
36. Platzer, A.: Logic & proofs for cyber-physical systems. In: Olivetti, N., Tiwari, A. (eds.) *IJCAR 2016. LNAI*, vol. 9706, pp. 15–21. Springer, Switzerland (2016)
37. Platzer, A.: Differential hybrid games. *ACM Trans. Comput. Log.* **18**(3) (2017)
38. Platzer, A., Clarke, E.M.: Formal verification of curved flight collision avoidance maneuvers: A case study. In: Cavalcanti, A., Dams, D. (eds.) *FM 2009. LNCS*, vol. 5850, pp. 547–562. Springer, Heidelberg (2009)
39. Platzer, A., Quesel, J.D.: European Train Control System: A case study in formal verification. In: Breitman, K., Cavalcanti, A. (eds.) *ICFEM 2009. LNCS*, vol. 5885, pp. 246–265. Springer, Heidelberg (2009)
40. President’s Council of Advisors on Science and Technology: *Leadership under challenge: Information technology R&D in a competitive world. An Assessment of the Federal Networking and Information Technology R&D Program*, August 2007
41. Quesel, J.D., Mitsch, S., Loos, S., Aréchiga, N., Platzer, A.: How to model and prove hybrid systems with KeYmaera: A tutorial on safety. *STTT* **18**(1), 67–91 (2016)
42. Tabuada, P.: *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer (2009)
43. Tiwari, A.: Logic in software, dynamical and biological systems. In: *LICS*, pp. 9–10. IEEE Computer Society (2011)
44. Wing, J.M.: Five deep questions in computing. *Commun. ACM* **51**(1), 58–60 (2008)

# Machine Learning for Programming

Martin Vechev

Department of Computer Science, ETH Zurich, Switzerland  
`martin.vechev@inf.ethz.ch`

In this talk I will discuss some of our latest research on creating probabilistic programming tools based on machine learning. These tools leverage the massive effort already spent by thousands of programmers and make useful predictions about new, unseen programs, helping solve difficult and important software tasks. I will illustrate several such probabilistic systems including statistical code synthesis and deobfuscation. Two of these de-obfuscation systems ([jsnice.org](http://jsnice.org) and [apk-deguard.com](http://apk-deguard.com)) are freely available, used daily and have more than 200,000 users from every country worldwide. I will also present new methods for creating probabilistic models that some of our systems are based on. These methods are more precise than neural networks and have applications to other domains, beyond code (e.g., to modeling natural language). Finally, I will conclude with what I believe are some of the more interesting, open problems in this area.



# Contents

## Cyber-Physical Systems

An Active Learning Approach to the Falsification of Black Box Cyber-Physical Systems . . . . .	3
<i>Simone Silveti, Alberto Policriti, and Luca Bortolussi</i>	
Modelling and Verification of Timed Robotic Controllers . . . . .	18
<i>Pedro Ribeiro, Alvaro Miyazawa, Wei Li, Ana Cavalcanti, and Jon Timmis</i>	
Spatial Reasoning About Motorway Traffic Safety with Isabelle/HOL . . . . .	34
<i>Sven Linker</i>	
Formalising and Monitoring Traffic Rules for Autonomous Vehicles in Isabelle/HOL . . . . .	50
<i>Albert Rizaldi, Jonas Keinholz, Monika Huber, Jochen Feldle, Fabian Immler, Matthias Althoff, Eric Hilgendorf, and Tobias Nipkow</i>	

## Software Verification Tools

Making Whiley Boogie! . . . . .	69
<i>Mark Utting, David J. Pearce, and Lindsay Groves</i>	
Complexity Analysis for Java with AProVE . . . . .	85
<i>Florian Frohn and Jürgen Giesl</i>	
The VerCors Tool Set: Verification of Parallel and Concurrent Software . . . .	102
<i>Stefan Blom, Saeed Darabi, Marieke Huisman, and Wytse Oortwijn</i>	
An Extension of the ABS Toolchain with a Mechanism for Type Checking SPLs . . . . .	111
<i>Ferruccio Damiani, Michael Lienhardt, Radu Muschevici, and Ina Schaefer</i>	

## Safety-Critical Systems

Generalised Test Tables: A Practical Specification Language for Reactive Systems . . . . .	129
<i>Bernhard Beckert, Suhyun Cha, Mattias Ulbrich, Birgit Vogel-Heuser, and Alexander Weigl</i>	

Transient and Steady-State Statistical Analysis for Discrete Event Simulators. . . . .	145
<i>Stephen Gilmore, Daniël Reijbergen, and Andrea Vandin</i>	
Algebraic Compilation of Safety-Critical Java Bytecode. . . . .	161
<i>James Baxter and Ana Cavalcanti</i>	
Task-Node Mapping in an Arbitrary Computer Network Using SMT Solver. . .	177
<i>Andrii Kovalov, Elisabeth Lobe, Andreas Gerndt, and Daniel Lüdtk</i>	
<b>Concurrency and Distributed Systems</b>	
Analysis of Synchronisations in Stateful Active Objects. . . . .	195
<i>Ludovic Henrio, Cosimo Laneve, and Vincenzo Mastandrea</i>	
BTS: A Tool for Formal Component-Based Development . . . . .	211
<i>Dalay Israel de Almeida Pereira, Marcel Vinicius Medeiros Oliveira, Madiel S. Conserva Filho, and Sarah Raquel Da Rocha Silva</i>	
Testing and Verifying Chain Repair Methods for CORFU Using Stateless Model Checking . . . . .	227
<i>Stavros Aronis, Scott Lystig Fritchie, and Konstantinos Sagonas</i>	
Synthesizing Coalitions for Multi-agent Games. . . . .	243
<i>Wei Ji, Farn Wang, and Peng Wu</i>	
<b>Program Verification Techniques</b>	
Hoare-Style Reasoning from Multiple Contracts . . . . .	263
<i>Olaf Owe, Toktam Ramezanifarkhani, and Elahe Fazeldehkhordi</i>	
A New Invariant Rule for the Analysis of Loops with Non-standard Control Flows. . . . .	279
<i>Dominic Steinhöfel and Nathan Wasser</i>	
Triggerless Happy: Intermediate Verification with a First-Order Prover . . . . .	295
<i>YuTing Chen and Carlo A. Furia</i>	
SEMSLICE: Exploiting Relational Verification for Automatic Program Slicing . . . . .	312
<i>Bernhard Beckert, Thorsten Bormer, Stephan Gocht, Mihai Herda, Daniel Lentzsch, and Mattias Ulbrich</i>	
<b>Formal Modeling</b>	
VBPMN: Automated Verification of BPMN Processes (Tool Paper) . . . . .	323
<i>Ajay Krishna, Pascal Poizat, and Gwen Salaün</i>	

How Well Can I Secure My System? . . . . .	332
<i>Barbara Kordy and Wojciech Widel</i>	
MaxUSE: A Tool for Finding Achievable Constraints and Conflicts for Inconsistent UML Class Diagrams . . . . .	348
<i>Hao Wu</i>	
Formal Verification of CNL Health Recommendations. . . . .	357
<i>Fahrurrozi Rahman and Juliana Küster Filipe Bowles</i>	
<b>Verified Software</b>	
Modular Verification of Order-Preserving Write-Back Caches. . . . .	375
<i>Jörg Pfähler, Gidon Ernst, Stefan Bodenmüller, Gerhard Schellhorn, and Wolfgang Reif</i>	
Formal Verification of ARP (Address Resolution Protocol) Through SMT-Based Model Checking - A Case Study - . . . . .	391
<i>Danilo Bruschi, Andrea Di Pasquale, Silvio Ghilardi, Andrea Lanzi, and Elena Pagani</i>	
Certified Password Quality: A Case Study Using Coq and Linux Pluggable Authentication Modules . . . . .	407
<i>João F. Ferreira, Saul A. Johnson, Alexandra Mendes, and Phillip J. Brooke</i>	
Verification of STAR-Vote and Evaluation of FDR and ProVerif . . . . .	422
<i>Murat Moran and Dan S. Wallach</i>	
<b>Author Index</b> . . . . .	437