Lecture Notes in Computer Science 9583

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

Advanced Research in Computing and Software Science
Subline of Lecture Notes in Computer Science

Subline Series Editors

Giorgio Ausiello, University of Rome ‘La Sapienza’, Italy
Vladimiro Sassone, University of Southampton, UK

Subline Advisory Board

Susanne Albers, TU Munich, Germany
Benjamin C. Pierce, University of Pennsylvania, USA
Bernhard Steffen, University of Dortmund, Germany
Deng Xiaotie, City University of Hong Kong
Jeannette M. Wing, Microsoft Research, Redmond, WA, USA
More information about this series at http://www.springer.com/series/7407
Verification, Model Checking, and Abstract Interpretation

17th International Conference, VMCAI 2016
St. Petersburg, FL, USA, January 17–19, 2016
Proceedings

Springer
Preface


VMCAI provides a forum for researchers from the communities of verification, model checking, and abstract interpretation, facilitating interaction, cross-fertilization, and advancement of hybrid methods that combine these and related areas. VMCAI topics include: program verification, model checking, abstract interpretation and abstract domains, program synthesis, static analysis, type systems, deductive methods, program certification, debugging techniques, program transformation, optimization, hybrid and cyber-physical systems.

This year the conference attracted 89 abstract submission leading to 67 full-paper submissions. Each submission was reviewed by at least three Program Committee members. The committee decided to accept 24 papers. The principal selection criteria were relevance, quality, and originality. We are glad to include in the proceedings the contributions of three invited keynote speakers: Peter Müller on “Viper — A Verification Infrastructure for Permission-based Reasoning,” Bryan Parno on “Ironclad — Full Verification of Complex Systems,” and Thomas Reps on “Automating Abstract Interpretation.” We would like to thank them for sharing their insights with us through their talks and articles contributed to the proceedings.

We thank our wonderful Program Committee members and reviewers for their reviews and discussions. Our gratitude goes to the Steering Committee members for their helpful advice and support, in particular to Lenore Zuck and Dave Schmidt for their assistance and invaluable experience with the organization of VMCAI. We would like to thank Annabel Satin for the great help in coordinating the events co-located with POPL 2016. We are indebted to EasyChair for providing us with an excellent conference management system. Finally, we thank our sponsors, Facebook and Microsoft Research, as well as NSF for providing travel grants for students.

November 2015

Barbara Jobstmann
K. Rustan M. Leino
Organization

Program Committee

Bor-Yuh Evan Chang  University of Colorado Boulder, USA
Hana Chockler  King’s College London, UK
Eva Darulova  MPI for Software Systems, Germany
Rayna Dimitrova  MPI for Software Systems, Germany
Javier Esparza  Technical University of Munich, Germany
Aarti Gupta  Princeton University, USA
Arie Gurfinkel  Software Engineering Institute, CMU, USA
Barbara Jobstmann  EPFL, Switzerland
Rustan Leino  Microsoft Research, USA
Francesco Logozzo  Facebook, USA
Madhavan Mukund  Chennai Mathematical Institute, India
Peter Müller  ETH Zürich, Switzerland
David Parker  University of Birmingham, UK
Andreas Podelski  University of Freiburg, Germany
Nadia Polikarpova  MIT CSAIL, USA
Philipp Rümmer  Uppsala University, Sweden
Roopsha Samanta  Institute of Science and Technology, Austria
Martina Seidl  Johannes Kepler University Linz, Austria
Sanjit A. Seshia  UC Berkeley, USA
Sharon Shoham  The Academic College of Tel Aviv Yaffo, Israel
Tachio Terauchi  Japan Advanced Institute of Science and Technology, Japan
Caterina Urban  ETH Zürich, Switzerland
Thomas Wies  New York University, USA
Lenore Zuck  University of Illinois in Chicago, USA

Steering Committee

Agostino Cortesi  Ca Foscari University of Venice, Italy
Patrick Cousot  CNRS and ENS and Inria, France and New York University, USA
E. Allen Emerson  University of Texas at Austin, USA
Andreas Podelski  University of Freiburg, Germany
Thomas W. Reps  University of Wisconsin at Madison, USA
David Schmidt  Kansas State University, USA
Lenore Zuck  University of Illinois at Chicago, USA
Additional Reviewers

Avni, Guy
Backeman, Peter
Bucur, Stefan
Cai, Xiaojuan
Chistikov, Dmitry
Christakis, Maria
Daca, Przemyslaw
Darais, David
Davies, Jessica
Donzé, Alexandre
Ehlers, Rüdiger
Feret, Jerome
Ferrara, Pietro
Ferrere, Thomas
Fremont, Daniel J.
Furia, Carlo A.
Garoche, Pierre-Loic
Gjomemo, Rigel
Hahn, Ernst Moritz
Hoffmann, Philipp
Itzhaky, Shachar
Kim, Eric
Kincaid, Zachary
Komuravelli, Anvesh
Krishna, Siddharth
Kupriyanov, Andrey
Lahav, Ori
Lammich, Peter
Meshman, Yuri
Meyer, Philipp J.
Milicevic, Aleksandar
Miné, Antoine

Mover, Sergio
Mukherjee, Suvam
Namjoshi, Kedar
Navas, Jorge A.
Ngo, Tuan Phong
Padon, Oded
Pavlinovic, Zvonimir
Pavlogiannis, Andreas
Prasad, Sanjiva
Rabe, Markus N.
Raghothaman, Mukund
Reynolds, Andrew
Rinetzky, Noam
Sadigh, Dorsa
Schilling, Christian
Schwerhoff, Malte
Schäf, Martin
Shenoy R., Gautham
Shoukry, Yasser
Sickert, Salomon
Sinha, Rohit
Srivathsan, B.
Summers, Alexander J.
Suresh, S.P.
Suwimonteerabuth, Dejvuth
Tarrach, Thorsten
Totla, Nishant
Unno, Hiroshi
Van Horn, David
Zeljić, Aleksandar
Zhai, Ennan
Ironclad: Full Verification of Complex Systems
(Invited Talk)

Bryan Parno
Microsoft Research

The Ironclad project at Microsoft Research is using a set of new and modified tools based on automated theorem proving to build Ironclad services. An Ironclad service guarantees to remote parties that every CPU instruction the service executes adheres to a high-level specification, convincing clients that the service will be worthy of their trust. To provide such end-to-end guarantees, we built a full stack of verified software. That software includes a verified kernel; verified drivers; verified system and cryptography libraries including SHA, HMAC, and RSA; and four Ironclad Apps [1]. As a concrete example, our Ironclad database provably provides differential privacy to its data contributors. In other words, if a client encrypts her personal data with the database’s public key, then it can only be decrypted by software that guarantees, down to the assembly level, that it preserves differential privacy when releasing aggregate statistics about the data.

We’ve also recently expanded the scope of our verification efforts to distributed systems, which are notorious for harboring subtle bugs. We have developed IronFleet [2], a methodology for building practical and provably correct distributed systems. We demonstrated the methodology on a complex implementation of a Paxos-based replicated state machine library and a lease-based sharded key-value store. We proved that each obeys a concise safety specification, as well as desirable liveness requirements. Each implementation achieves performance competitive with a reference system.

In this talk, we describe our methodology, formal results, and lessons we learned from building large stacks of verified systems software. In pushing automated verification tools to new scales (over 70K lines of code and proof so far), our team has both benefited from automated verification techniques and uncovered new challenges in using them.

By continuing to push verification tools to larger and more complex systems, Ironclad ultimately aims to raise the standard for security- and reliability-critical systems from “tested” to “correct”.

References

## Contents

### Invited Talks

- Automating Abstract Interpretation ........................................... 3  
  *Thomas Reps and Aditya Thakur*
- Viper: A Verification Infrastructure for Permission-Based Reasoning .... 41  
  *Peter Müller, Malte Schwerhoff, and Alexander J. Summers*

### Abstract Interpretation

- Predicate Abstraction for Linked Data Structures ......................... 65  
  *Alexander Bakst and Ranjit Jhala*
- An Abstract Domain of Uninterpreted Functions .............................. 85  
  *Graeme Gange, Jorge A. Navas, Peter Schachte, Harald Søndergaard, and Peter J. Stuckey*
- Property Directed Abstract Interpretation .................................. 104  
  *Noam Rinetzky and Sharon Shoham*

### Abstraction

- Program Analysis with Local Policy Iteration ............................... 127  
  *Egor George Karpenkov, David Monniaux, and Philipp Wendler*
- Lazy Constrained Monotonic Abstraction .................................... 147  
  *Zeinab Ganjei, Ahmed Rezine, Petru Eles, and Zebo Peng*
- Polyhedral Approximation of Multivariate Polynomials Using Handelman’s Theorem ................................................................. 166  
  *Alexandre Maréchal, Alexis Fouilhè, Tim King, David Monniaux, and Michael Périn*
- $D^3$: Data-Driven Disjunctive Abstraction .................................. 185  
  *Hila Peleg, Sharon Shoham, and Eran Yahav*
- Exact Heap Summaries for Symbolic Execution ............................... 206  
  *Benjamin Hillery, Eric Mercer, Neha Rungta, and Suzette Person*
Hybrid and Timed Systems

Abstract Interpretation with Infinitesimals: Towards Scalability
in Nonstandard Static Analysis ........................................... 229
Kengo Kido, Swarat Chaudhuri, and Ichiro Hasuo

Lipschitz Robustness of Timed I/O Systems ............................. 250
Thomas A. Henzinger, Jan Otop, and Roopsha Samanta

A Method for Invariant Generation for Polynomial Continuous Systems .... 268
Andrew Sogokon, Khalil Ghorbal, Paul B. Jackson, and André Platzer

Dynamic and Static Verification

Hybrid Analysis for Partial Order Reduction of Programs with Arrays ...... 291
Pavel Parízek

Cloud-Based Verification of Concurrent Software ........................... 311
Gerard J. Holzmann

Abstraction-driven Concolic Testing ........................................ 328
Przemysław Daca, Ashutosh Gupta, and Thomas A. Henzinger

Probabilistic Systems

Reward-Bounded Reachability Probability for Uncertain Weighted MDPs ... 351
Vahid Hashemi, Holger Hermanns, and Lei Song

Parameter Synthesis for Parametric Interval Markov Chains ............... 372
Benoît Delahaye, Didier Lime, and Laure Petrucci

Concurrent Programs

Pointer Race Freedom .............................................................. 393
Frédéric Haziza, Lukáš Holík, Roland Meyer, and Sebastian Wolff

A Program Logic for C11 Memory Fences ................................... 413
Marko Doko and Viktor Vafeiadís

From Low-Level Pointers to High-Level Containers ......................... 431
Kamil Dudka, Lukáš Holík, Petr Peringer, Marek Trtík, and Tomáš Vojnar
Parameterized and Component-Based Systems

Regular Symmetry Patterns ............................................. 455
Anthony W. Lin, Truong Khanh Nguyen, Philipp Rümmer, and Jun Sun

Tight Cutoffs for Guarded Protocols with Fairness ....................... 476
Simon Außerelechner, Swen Jacobs, and Ayrat Khalimov

A General Modular Synthesis Problem for Pushdown Systems .......... 495
Ilaria De Crescenzo and Salvatore La Torre

Solver Improvements

Model Checking with Multi-threaded IC3 Portfolios ..................... 517
Sagar Chaki and Derrick Karimi

Automatic Generation of Propagation Complete SAT Encodings ........ 536
Martin Brain, Liana Hadarean, Daniel Kroening, and Ruben Martins

Author Index ............................................................. 557