Lecture Notes in Computer Science 10145

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Advanced Research in Computing and Software Science

Subline of Lecture Notes in Computer Science

More information about this series at http://www.springer.com/series/7407

Ahmed Bouajjani · David Monniaux (Eds.)

Verification, Model Checking, and Abstract Interpretation

18th International Conference, VMCAI 2017
Paris, France, January 15–17, 2017
Proceedings

Springer

Editors
Ahmed Bouajjani
IRIF, Université Paris Diderot
Paris
France

David Monniaux
VERIMAG, CNRS & Université
Grenoble Alpes
Grenoble
France

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-319-52233-3 ISBN 978-3-319-52234-0 (eBook)
DOI 10.1007/978-3-319-52234-0

Library of Congress Control Number: 2016963156

LNCS Sublibrary: SL1 – Theoretical Computer Science and General Issues

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

This volume contains the papers presented at VMCAI 2017, the 18th International Conference on Verification, Model Checking, and Abstract Interpretation, held during January 15–17, 2017, in Paris, France, co-located with POPL 2017 (the annual ACM SIGPLAN/SIGACT Symposium on Principles of Programming Languages). Previous meetings were held in Port Jefferson (1997), Pisa (1998), Venice (2002), New York (2003), Venice (2004), Paris (2005), Charleston (2006), Nice (2007), San Francisco (2008), Savannah (2009), Madrid (2010), Austin (2011), Philadelphia (2012), Rome (2013), San Diego (2014), Mumbai (2015), and St. Petersburg, Florida (2016).

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 60 submissions. Each submission was reviewed by at least three Program Committee members. The committee decided to accept 27 papers. The principal selection criteria were relevance, quality, and originality. We are pleased to include in the proceedings the contributions of three invited keynote speakers: Ernie Cohen (Amazon Web Services), Pascal Cuoq (Trust in Soft), and Jasmin Fisher (Microsoft Research). We warmly thank them for their participation and for their contributions.

We would like also to thank the members of the Program Committee and the external reviewers for their excellent work. We also thanks the members of the Steering Committee, and in particular Andreas Podelski and Lenore Zuck, for their helpful advice, assistance, and support. We also thank Laure Gonnord for her invaluable help in all aspects related to the organization of the conference. We thank Annabel Satin for the help in coordinating the events co-located with POPL 2017, and we thank the POPL 2017 Organizing Committee for providing all the logistics for organizing VMCAI. We are also indebted to EasyChair for providing us with an excellent conference management system.

Finally, we would like to thank our generous sponsors: AdaCore, Amazon Web Services, Facebook, and Microsoft Research.

December 2016

Ahmed Bouajjani
David Monniaux

Organization

Program Committee

Erika Abraham	RWTH Aachen University, Germany
Mohamed Faouzi Atig	Uppsala University, Sweden
Roderick Bloem	Graz University of Technology, Austria
Ahmed Bouajjani	IRIF, Paris Diderot University, France
Wei-Ngan Chin	National University of Singapore, Singapore
Deepak D'Souza	Indian Institute of Science, Bangalore, India
Cezara Drăgoi	Inria, ENS, France
Roberto Giacobazzi	University of Verona, Italy
Laure Gonnord	University of Lyon/LIP, France
Orna Grumberg	Technion - Israel Institute of Technology, Israel
Dejan Jovanović	SRI International, USA
Konstantin Korovin	Manchester University, UK
Laura Kovacs	Vienna University of Technology, Austria
Shuvendu Lahiri	Microsoft Research, USA
Akash Lal	Microsoft Research, India
Rupak Majumdar	MPI-SWS, Germany
David Monniaux	VERIMAG, CNRS & Université Grenoble Alpes, France
Madhavan Mukund	Chennai Mathematical Institute, India
Corina Pasareanu	CMU/NASA Ames Research Center, USA
Andreas Podelski	University of Freiburg, Germany
Jean-Francois Raskin	Université Libre de Bruxelles, Belgium
Sriram Sankaranarayanan	University of Colorado, Boulder, USA
Armando Solar-Lezama	MIT, USA
Marielle Stoelinga	University of Twente, The Netherlands
Boris Yakobowski	CEA, LIST, France

Additional Reviewers

Basso-Blandin, Adrien
Ben-Amram, Amir
Blom, Stefan
Bobot, François
Brain, Martin
Braud-Santoni, Nicolas
Cai, Zhouhong
Castellan, Simon
Chakarov, Aleksandar
Costea, Andreea
Coti, Camille
Darabi, Saeed
Dehnert, Christian
Demange, Delphine
Enea, Constantin
Feret, Jerome
Forget, Julien
Frenkel, Hadar

Garg, Pranav
Ghilardi, Silvio
Girault, Alain
Gleiss, Bernhard
Habermehl, Peter
Hadarean, Liana
Halbwachs, Nicolas
He, Shaobo
Heußner, Alexander
Ho, Hsi-Ming
Iusupov, Rinat
Jansen, Nils
Jaroschek, Maximilian
Jecker, Ismaël
Khalimov, Ayrat
Koenighofer, Bettina
Konnov, Igor
Korovina, Margarita
Kremer, Gereon
Kretinsky, Jan
Lange, Tim
Le Roux, Stephane
Le, Quang Loc
Le, Ton Chanh
Lee, Benedict
Mastroeni, Isabella
Matteplackel, Raj Mohan
Merz, Stephan
Mukherjee, Suvam
Muoi, Tran Duc
Narayan Kumar, K.
Navas, Jorge A.
Ngo, Tuan Phong
Niksic, Filip
Petri, Gustavo
Rakamaric, Zvonimir
Rasin, Dan
Rensink, Arend
Rezine, Othmane
Rodriguez, Cesar
Roeck, Franz
Rothenberg, Bat-Chen
Sangnier, Arnaud
Scherer, Gabriel
Schilling, Christian
Shi, Jinghao
Sofronie-Stokkermans, Viorica
Suda, Martin
Tiwari, Ashish
Urban, Caterina
van Glabbeek, Rob
Vedrine, Franck
Verdoolaege, Sven
Widder, Josef

Abstracts of Invited Talks

Bringing LTL Model Checking to Biologists

Zara Ahmed[1], David Benque[2], Sergey Berezin[3],
Anna Caroline E. Dahl[4], Jasmin Fisher[1,5], Benjamin A. Hall[6],
Samin Ishtiaq[1], Jay Nanavati[1], Nir Piterman[7],
Maik Riechert[1], and Nikita Skoblov[3]

[1] Microsoft Research, Cambridge, UK
`jasmin.fisher@microsoft.com`
[2] Royal College of Art, London, UK
[3] Moscow State University, Moscow, Russia
[4] Center for Technology in Medicine and Health,
KTH Royal Institute of Technology, Huddinge, Sweden
[5] Department of Biochemistry, University of Cambridge, Cambridge, UK
[6] MRC Cancer Unit, University of Cambridge, Cambridge, UK
[7] University of Leicester, Leicester, UK

Abstract. The BioModelAnalyzer (BMA) is a web based tool for the development of discrete models of biological systems. Through a graphical user interface, it allows rapid development of complex models of gene and protein interaction networks and stability analysis without requiring users to be proficient computer programmers. Whilst stability is a useful specification for testing many systems, testing temporal specifications in BMA presently requires the user to perform simulations. Here we describe the LTL module, which includes a graphical and natural language interfaces to testing LTL queries. The graphical interface allows for graphical construction of the queries and presents results visually in keeping with the current style of BMA. The Natural language interface complements the graphical interface by allowing a gentler introduction to formal logic and exposing educational resources.

Verified Concurrent Code: Tricks of the Trade

Ernie Cohen

Amazon Web Services, Wyncote, USA
ecohen@amazon.com

Abstract. Modular code verification, suitably extended with shared atomic objects, supports a number of useful verification idioms and semantic models, without further logical extension.

Keywords: Real-time · Hybrid systems · Probability · Stopping failures · Weak memory · Cryptography · Ownership · Permissions · Simulation · Knowledge · Behavioral polymorphism · Device drivers · Concurrent data structures · Transactions · Linearizability · Deductive verification · VCC

Detecting Strict Aliasing Violations in the Wild

Pascal Cuoq[1], Loïc Runarvot[1], and Alexander Cherepanov[2,3]

[1] TrustInSoft, Paris, France
cuoq@trust-in-soft.com
[2] Openwall, Moscow, Russia
[3] National Research University Higher School of Economics,
Moscow, Russia

Abstract. Type-based alias analyses allow C compilers to infer that memory locations of distinct types do not alias. Idiomatic reliance on pointers on the one hand, and separate compilation on the other hand, together make it impossible to get this aliasing information any other way. As a consequence, most modern optimizing C compilers implement some sort of type-based alias analysis. Unfortunately, pointer conversions, another pervasive idiom to achieve code reuse in C, can interact badly with type-based alias analyses. This article investigate the fine line between the allowable uses of low-level constructs (pointer conversions, unions) that should never cause the predictions of a standard-compliant type-based alias analysis to be wrong, and the dangerous uses that can result in bugs in the generated binary. A sound and precise analyzer for strict aliasing violations is briefly described.

Contents