Lecture Notes in Computer Science **10028**

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

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

Roderick Bloem · Eli Arbel (Eds.)

Hardware and Software: Verification and Testing

12th International
Haifa Verification Conference, HVC 2016
Haifa, Israel, November 14–17, 2016
Proceedings

Springer

Editors
Roderick Bloem
IAIK
Graz University of Technology
Graz
Austria

Eli Arbel
IBM Research Labs
Haifa
Israel

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-319-49051-9 ISBN 978-3-319-49052-6 (eBook)
DOI 10.1007/978-3-319-49052-6

Library of Congress Control Number: 2016956611

LNCS Sublibrary: SL2 – Programming and Software Engineering

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 proceedings of the 12th Haifa Verification Conference (HVC 2016). The conference was hosted by IBM Research Haifa Laboratory and took place during November 14–17, 2016. It was the 12th event in this series of annual conferences dedicated to advancing the state of the art and state of the practice in verification and testing. The conference provided a forum for researchers and practitioners from academia and industry to share their work, exchange ideas, and discuss the future directions of testing and verification for hardware, software, and complex hybrid systems.

Overall, HVC 2016 attracted 26 submissions in response to the call for papers. Each submission was assigned to at least three members of the Program Committee and in some cases additional reviews were solicited from external experts. The Program Committee selected 13 papers for presentation. In addition to the contributed papers, the program included four invited talks, by Swarat Chaudhuri (Rice University), Markulf Kohlweiss (Microsoft Research), Rajeev Ranjan (Cadence), and Andreas Veneris (University of Toronto). On the last day of the conference, the HVC award was presented to Marta Kwiatkowska (University of Oxford), Gethin Norman (University of Glasgow), and Dave Parker (University of Birmingham), for the invention, development and maintenance of the PRISM probabilistic model checker. A special session about verification and testing challenges of autonomous systems was held on the first day of the conference. Thanks to Yoav Hollander (Foretellix LTD) for presenting in this session. On November 13, one day before the conference, we held a tutorial day with tutorials by Sanjit A. Seshia (University of California, Berkeley) on formal inductive synthesis, by Hari Mony (IBM) on sequential equivalence checking for hardware design and verification, by Amir Rahat (Optima Design Automation) on design reliability, and by Cristian Cadar (Imperial College) on dynamic symbolic execution and the KLEE infrastructure.

We would like to extend our appreciation and sincere thanks to the local organization team from IBM Research Haifa Laboratory: Tali Rabetti, the publicity chair, Revivit Yankovich, the local coordinator, Yair Harry, the Web master, and the Organizing Committee, which consisted of Laurent Fournier, Sharon Keidar-Barner, Moshe Levinger, Michael Vinov, Karen Yorav, and Avi Ziv. We would also like to thank the tutorial chair Natasha Sharygina (University of Lugano), and the HVC Award Committee, consisting of Armin Biere (Johannes Kepler University), Hana Chockler (King's College London), Kerstin Eder (University of Bristol), Andrey Rybalchenko (Microsoft Research), Ofer Strichman (Technion), and particularly its energetic chair, Leonardo Mariani (University of Milano Bicocca).

HVC 2016 received sponsorships from IBM, Cadence Design Systems, Mellanox Technologies, Mentor Graphics, Qualcomm, and Intel. (Thanks!)

Submission and evaluation of papers, as well as the preparation of this proceedings volume, were handled by the EasyChair conference management system. (Thanks, Andrei!)

It was a pleasure to organize this conference with so many old friends!

Graz
September 2016

Eli Arbel
Roderick Bloem

Organization

Program Committee

Eli Arbel	IBM Research, Israel
Domagoj Babic	Google, USA
Aviv Barkai	Intel Corporation, Israel
Nikolaj Bjorner	Microsoft Research, USA
Roderick Bloem	Graz University of Technology, Austria
Hana Chockler	King's College London, UK
Rayna Dimitrova	MPI-SWS, Germany
Adrian Evans	iRoC Technologies, France
Franco Fummi	University of Verona, Italy
Raviv Gal	IBM Research, Israel
Warren Hunt	University of Texas, USA
Barbara Jobstmann	EPFL and Cadence Design Systems, Switzerland
Laura Kovacs	Vienna University of Technology, Austria
João Lourenço	NOVA LINCS – Universidade Nova de Lisboa, Portugal
Annalisa Massini	Sapienza University of Rome, Italy
Hari Mony	IBM Corporation, USA
Nir Piterman	University of Leicester, UK
Pavithra Prabhakar	Kansas State University, USA
Sandip Ray	NXP Semiconductors, USA
Orna Raz	HRL, IBM Research, Israel
Martina Seidl	Johannes Kepler University Linz, Asutria
Sanjit A. Seshia	UC Berkeley, USA
A. Prasad Sistla	University of Illinois at Chicago, USA
Ufuk Topcu	University of Texas at Austin, USA
Eran Yahav	Technion, Israel

Additional Reviewers

Arechiga, Nikos
Dreossi, Tommaso
Fremont, Daniel J.
Gao, Sicun
Junges, Sebastian
Krakovski, Roi
Lal, Ratan
Mari, Federico
Rabe, Markus N.
Rabetti, Tali
Sadigh, Dorsa
Salvo, Ivano
Soto, Miriam Garcia
Veneris, Andreas

Abstracts

Current Trends and Future Direction in Eco-system of Hardware Formal Verification: A Technical and Business Perspective

Rajeev K. Ranjan

Cadence, San Jose, USA

Hardware formal verification is increasingly being adopted in the modern SoC design and verification flow for architectural specification and verification through RTL development and debugging through SoC integration – all the way up to post-silicon debugging. The productivity and quality benefits of adopting this technology for a gamut of verification tasks are well established. In this talk, we will cover the current trends and future directions in this area that is shaped by the technical feasibility of the solutions and the business RoI seen by different stakeholders V- chip companies, design/verification engineers, formal EDA vendors, and formal solution development engineers.

Guiding Formal Methods with Discovered Knowledge

Swarat Chaudhuri

Rice University, Houston, USA

Systems for automated formal reasoning about programs depend on human specification at multiple levels. Users of such a system must write full specifications of the tasks that they want performed. The designer of the system is expected to formalize the domain-specific language in which tasks are described, and specify the domain-dependent heuristics that guide automated reasoning. The assumption of specifications reflects a common expectation in formal methods research: that humans hold deep knowledge about problem domains and instances. In practice, this expectation can be violated and lead to hard-to-use or brittle tools. In this talk, we describe a new class of formal methods that address this difficulty through automatic discovery of knowledge from corpora of pre-existing code, execution traces, and proofs.

The starting point of this work is the observation that a human who describes or solves a reasoning task does not do so in a vacuum, but using insights established through prior experiences of their own or others. The thesis is that such insights can be algorithmically learned from datasets of existing formal artifacts, and can lead to systems for automated reasoning that demand less human intervention than traditional tools. The talk will describe multiple instantiations of this thesis, including a statistical notion of program correctness, a program synthesis algorithm guided by a "neural" model of program syntax and semantics, and an approach to program verification that uses pre-existing formal proofs.

Bug Wars: Automation Awakens

Andreas Veneris

Department of Electrical and Computer Engineering, and Department of Computer Science, University of Toronto, Toronto, Canada

Verification is the undisputed bottleneck in the design cycle consuming two thirds of the total chip development effort. This is in part because of the complexity of modern designs, the impact of geographical dispersed teams integrating new components with third-party/legacy IP under tight time-to-market goals, the evolving role of verification engineers to not only discover bugs but also aid correct them and the ever-evolving nature of the task itself. Today verification has stretched itself beyond its traditional boundaries into validation as most of silicon re-spins are not due to physical defects but because of functional errors not discovered or fixed earlier in the design cycle. Although parts of verification have been automated the core issue driving this gap remains debugging as it consumes more than half of the overall effort being a predominantly manual task.

In this talk, we revisit automation efforts in functional debug from late 1980s, when it was first introduced, to recent formal advances placing it into context as we recount new directions. In more detail, we will first outline early methodologies stemming from test and fault diagnosis to more recent techniques based on Boolean satisfiability. We will examine different angles of the debug problem and respective solutions for its various manifestations in the verification cycle. This will allow us to appraise theoretical and practical parallels in the foundations of those two tasks. As we evaluate the progress in debug automation, we will point out emerging deficiencies of existing methodologies more notably during regression verification. To that end, we will present novel techniques in debugging triage where statistical solutions, a radical departure from existing debug efforts, need complement traditional simulation/formal methods to not only take into account the design response but also the human factor. We will conclude with a mantra that research in debugging in the past 30 years points to a direction where verification and test prove once again to be fields deeply intertwined, and we will provide guidance for methodologies in silicon debug rooted on existing functional debug procedures.

miTLS: Can Cryptography, Formal Methods, and Applied Security be Friends?

Markulf Kohlweiss

Microsoft Research, Cambridge, UK

TLS was designed as a transparent channel abstraction to allow developers with no cryptographic expertise to protect their application against attackers that may control some clients, some servers, and may have the capability to tamper with network connections. However, the security guarantees of TLS fall short of those of a secure channel, leading to a variety of attacks. The Triple Handshake attack exploits combinations of RSA and Diffie-Hellman key exchange, session resumption, and renegotiation to bypass many recent countermeasures.

At the same time we study the provable security of TLS, as it is implemented and deployed. To capture the details of the standard and its main extensions, we rely on miTLS, a verified reference implementation of the protocol. miTLS inter-operates with mainstream browsers and servers for many protocol versions, configurations, and ciphersuites; and it provides application-level, provable security for some. This leads to the strange case of how something provable secure can be insecure.

In this talk I will play Dr Jekyll and Mr Hyde by playing off our CRYPTO proof and our S&P attack against each other.

Contents