Lecture Notes in Computer Science 8164

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

Ernie Cohen Andrey Rybalchenko (Eds.)

Verified Software: Theories, Tools, Experiments

5th International Conference, VSTTE 2013
Menlo Park, CA, USA, May 17-19, 2013
Revised Selected Papers

Springer

Volume Editors

Ernie Cohen
107 Hewett Road, Wyncote, PA 19095, USA
E-mail: erniecohen1@gmail.com

Andrey Rybalchenko
Microsoft Research Cambridge
21 Station Road, CB1 2FB Cambridge, UK
E-mail: rybal@microsoft.com
and
Technical University Munich
Boltzmannstr. 3, 85748 Munich, Germany
E-mail: rybal@in.tum.de

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-54107-0 e-ISBN 978-3-642-54108-7
DOI 10.1007/978-3-642-54108-7
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013957812

CR Subject Classification (1998): D.2.4, D.2, F.3, D.3, D.1, C.2, F.4

LNCS Sublibrary: SL 2 – Programming and Software Engineering

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

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

Preface

This volume contains the papers presented at the 5th International Conference on Verified Software: Theories, Tool and Experiments (VSTTE), which was held in Menlo Park, USA, during May 17–19, 2013. Historically, the conference originated from the Verified Software Initiative (VSI), a cooperative, international initiative directed at the scientific challenges of large-scale software verification. The inaugral VSTTE conference was held at ETH Zurich in October 2005. Starting in 2008, the conference became a biennial event, VSTTE 2008 was held in Toronto, VSTTE 2010 was held in Edinburgh, and VSTTE 2012 was held in Philadelphia, which changed this year.

The goal of the VSTTE conference is to advance the state of the art through the interaction of theory development, tool evolution, and experimental validation.

VSTTE 2013 is especially interested in submissions describing large-scale verification efforts that involve collaboration, theory unification, tool integration, and formalized domain knowledge. We welcome papers describing novel experiments and case studies evaluating verification techniques and technologies. Topics of interest include education, requirements modeling, specification languages, specification/verification case-studies, formal calculi, software design methods, automatic code generation, refinement methodologies, compositional analysis, verification tools (e.g., static analysis, dynamic analysis, model checking, theorem proving, satisfiability), tool integration, benchmarks, challenge problems, and integrated verification environments.

There were 35 submissions. Each submission was reviewed by at least two, and on average 2.7, Program Committee members. The committee decided to accept 17 papers. The program also includes three invited talks, by Alex Aiken (Stanford University), Nikhil Swamy (Microsoft Research), and Andre Platzer (CMU), as well as an invited tutorial by Sandrine Blazy (University of Rennes 1).

We would like to thank the invited speakers, all submitting authors, the Steering Committee, the conference chair, the publicity chair, the external reviewers, and especially the Program Committee, who put a lot of hard work into reviewing and selecting the papers that appear in this volume.

We thank Andrei Voronkov for the access to EasyChair and Springer.

VSTTE 2013 was supported in part by NSF funding CISE award 1033105.

November 2013

Ernie Cohen
Andrey Rybalchenko

Organization

Program Committee

Josh Berdine	Microsoft Research
Ahmed Bouajjani	University of Paris 7, France
Marsha Chechik	Toronto University, Canada
Ernie Cohen	Microsoft
Jean-Christophe Filliatre	CNRS, LRI, Inria, France
Silvio Ghilardi	University of Milan, Italy
Aarti Gupta	NEC Labs
Arie Gurfinkel	CMU SEI
Jifeng He	East China Normal University, China
Andrew Ireland	Heriot-Watt University, UK
Ranjit Jhala	UC San Diego, USA
Cliff Jones	Newcastle University, UK
Rajeev Joshi	NASA JPL, USA
Gerwin Klein	NICTA, Australia
Daniel Kroening	Oxford University, UK
Gary Leavens	University of Central Florida, USA
Xavier Leroy	Inria, France
Zhiming Liu	UNI-IIST
Pete Manolios	Northeastern University
Tiziana Margaria	University of Potsdam, Germany
David Monniaux	VERIMAG, France
Peter Mueller	ETHZ
David Naumann	Stevens Institute of Technology, USA
Aditya Nori	Microsoft Research
Peter O'Hearn	UCL, UK
Matthew Parkinson	Microsoft Research
Wolfgang Paul	University of Saarland, Germany
Andreas Podelski	University of Freiburg, Germany
Andrey Rybalchenko	TUM
Natarajan Shankar	SRI International, Singapore
Zhong Shao	Yale University, USA
Willem Visser	University of Stellenbosch, South Africa
Thomas Wies	NYU
Jim Woodcock	University of York, UK
Kwangkeun Yi	Seoul National University, South Korea
Pamela Zave	AT&T Labs
Lenore Zuck	University of Illinois at Chicago, USA

Additional Reviewers

Chamarthi, Harsh Raju
Chen, Zhenbang
Christ, Jürgen
David, Cristina
Faber, Johannes
Joshi, Saurabh
Majumdar, Rupak
Nipkow, Tobias
Papavasileiou, Vasilis
Popeea, Corneliu
Qamar, Nafees
Tautschnig, Michael

Invited Talks

Using Learning Techniques in Invariant Inference

Alex Aiken

Stanford University

Abstract. Arguably the hardest problem in automatic program verification is designing appropriate techniques for discovering loop invariants (or, more generally, recursive procedures). Certainly, if invariants are known, the rest of the verification problem becomes easier. This talk presents a family of invariant inference techniques based on using test cases to generate an underapproximation of program behavior and then using learning algorithms to generalize the underapproximation to an invariant. These techniques are simpler, much more efficient, and appear to be more robust than previous approaches to the problem. If time permits, some open problems will also be discussed.

F*: Certified Correctness for Higher-Order Stateful Programs

Nikhil Swamy

Microsoft Research

Abstract. Abstract: F* is an ML-like programming language being developed at Microsoft Research. It has a type system based on dependent types and a typechecker that makes use of an SMT solver to discharge proof obligations. The type system is expressive enough to express functional correctness properties of typical, higher-order stateful programs.
We have used F* in a variety of settings, including in the verification of security protocol implementations; as a source language for secure web-browser extensions; as an intermediate verification language for JavaScript code; to verify the correctness of compilers; as a relational logic for probabilistic programs; and as a proof assistant in which to carry out programming language metatheory. We have also used F* to program the core typechecker of F* itself and have verified that it is correct. By bootstrapping this process using the Coq proof assistant, we obtain a theorem that guarantees the existence of a proof certificate for typechecked programs.
I will present a brief overview of the F* project, drawing on the examples just mentioned to illustrate the features of the F* language and certification system.
For more about F*, visit `http://research.microsoft.com/fstar`.

How to Explain Cyber-Physical Systems to Your Verifier

André Platzer

CMU

Abstract. Despite the theoretical undecidability of program verification, practical verification tools have made impressive advances. How can we take verification to the next level and use it to verify programs in cyber-physical systems (CPSs), which combine computer programs with the dynamics of physical processes. Cars, aircraft, and robots are prime examples where this matters, because they move physically in space in a way that is determined by discrete computerized control algorithms. Because of their direct impact on humans, verification for CPSs is even more important than it already is for programs.

This talk describes how formal verification can be lifted to one of the most prominent models of CPS called hybrid systems, i.e. systems with interacting discrete and continuous dynamics. It presents the theoretical and practical foundations of hybrid systems verification. The talk shows a systematic approach that is based on differential dynamic logic comes with a compositional proof technique for hybrid systems and differential equations. This approach is implemented in the verification tool KeYmaera and has been used successfully for verifying properties of aircraft, railway, car control, autonomous robotics, and surgical robotics applications.

A Tutorial on the CompCert Verified Compiler

Sandrine Blazy

University of Rennes 1

Abstract. Compilers are complicated pieces of software that sometimes contain bugs causing wrong executable code to be silently generated from correct source programs. In turn, this possibility of compiler-introduced bugs diminishes the assurance that can be obtained by applying formal methods to source code. This talk gives an overview of the CompCert project: an ongoing experiment in developing and formally proving correct a realistic, moderately-optimizing compiler from a large subset of C to popular assembly languages. The correctness proof, mechanized using the Coq proof assistant, establishes that the generated assembly code behaves exactly as prescribed by the semantic of the C source, eliminating all possibilities of compiler-introduced bugs and generating unprecedented confidence in this compiler. For more about CompCert, please visit `http://compcert.inria.fr`.

Table of Contents