Lecture Notes in Computer Science 9807

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

Jasmin Christian Blanchette · Stephan Merz (Eds.)

Interactive Theorem Proving

7th International Conference, ITP 2016
Nancy, France, August 22–25, 2016
Proceedings

Springer

Editors
Jasmin Christian Blanchette
Inria Nancy – Grand Est
Villers-lès-Nancy
France

Stephan Merz
Inria Nancy – Grand Est
Villers-lès-Nancy
France

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-319-43143-7 ISBN 978-3-319-43144-4 (eBook)
DOI 10.1007/978-3-319-43144-4

Library of Congress Control Number: 2016945777

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 Switzerland

Preface

The International Conference on Interactive Theorem Proving (ITP) is the premier venue for publishing research in the area of logical frameworks and interactive proof assistants, ranging from theoretical foundations, technology, and implementation aspects to their applications in areas such as verifying algorithms and programs, ensuring their safety and security, or formalizing significant mathematical theories. ITP grew out of the TPHOLs conferences and ACL2 workshops organized since the early 1990s.

Previous editions of ITP took place in Edinburgh, Nijmegen, Princeton, Rennes, Vienna, and Nanjing. The seventh edition (ITP 2016) was organized by the Inria research center Nancy – Grand Est in Nancy, France, during August 22–25, 2016. In all, 55 submissions were received for ITP 2016. Each submitted paper was reviewed by at least three members of the Program Committee or external reviewers, and the Program Committee decided to accept 27 regular contributions and five rough diamonds. Viktor Kuncak, Grant Olney Passmore, and Nikhil Swamy were invited to present keynote talks at the conference. The main conference was followed by workshops dedicated to the Coq and Isabelle systems, as well as to the Mathematical Components library.

The present volume collects the scientific contributions accepted for publication at ITP 2016. It also contains abstracts of the keynote presentations.

We are very grateful to the members of the ITP Steering Committee for their guidance and advice. Our colleagues in the Program Committee and the external reviewers did an excellent job in preparing timely and helpful reviews as a basis for selecting the accepted contributions. We extend our thanks to the authors of all submitted papers and the ITP community at large, without which the conference would not exist.

The Inria research center Nancy – Grand Est, and in particular the delegate for colloquia Anne-Lise Charbonnier, provided professional support for the organization of ITP 2016. We gratefully acknowledge financial support by Aesthetic Integration, Communauté Urbaine du Grand Nancy, Microsoft Research, Région Alsace Champagne-Ardenne Lorraine, and Springer. As in previous years, Springer accepted to publish the proceedings of ITP 2016 as a volume in the LNCS series, and we would like to thank the editorial team for the very smooth interaction.

June 2016

Jasmin Christian Blanchette
Stephan Merz

Organization

Steering Committee

David Basin	ETH Zürich, Switzerland
Jasmin Christian Blanchette	Inria Nancy – Grand Est, France
Amy Felty	University of Ottawa, Canada
Panagiotis Manolios	Northeastern University, USA
César Muñoz	NASA Langley Research Center, USA
Christine Paulin-Mohring	Université Paris Sud, France
Lawrence Paulson	University of Cambridge, UK
Michael Norrish	NICTA, Australia
Tobias Nipkow	Technische Universität München, Germany
Sofiène Tahar	Concordia University, Canada
Christian Urban	King's College London, UK

Program Committee

Andrea Asperti	University of Bologna, Italy
Jeremy Avigad	Carnegie Mellon University, USA
Yves Bertot	Inria Sophia Antipolis – Méditerranée, France
Lars Birkedal	Aarhus University, Denmark
Jasmin Christian Blanchette	Inria Nancy – Grand Est, France
Adam Chlipala	MIT, USA
Nils Anders Danielsson	University of Gothenburg and Chalmers University of Technology, Sweden
Amy Felty	University of Ottawa, Canada
Herman Geuvers	Radboud University Nijmegen, The Netherlands
Georges Gonthier	Microsoft Research, UK
John Harrison	Intel, USA
Hugo Herbelin	Inria Paris, France
Cătălin Hriţcu	Inria Paris, France
Cezary Kaliszyk	University of Innsbruck, Austria
Matt Kaufmann	University of Texas at Austin, USA
Gerwin Klein	NICTA and UNSW, Australia
Xavier Leroy	Inria Paris, France
Andreas Lochbihler	ETH Zürich, Switzerland
Frédéric Loulergue	Université d'Orleans and LIFO, France
Assia Mahboubi	Inria Saclay – Île-de-France, France
Panagiotis Manolios	Northeastern University, USA
Stephan Merz	Inria Nancy – Grand Est, France

Magnus O. Myreen	Chalmers University of Technology, Sweden
Adam Naumowicz	University of Białystok, Poland
Tobias Nipkow	Technische Universität München, Germany
Michael Norrish	NICTA, Australia
Sam Owre	SRI International, USA
Christine Paulin-Mohring	Université Paris-Sud and LRI, France
Lawrence C. Paulson	University of Cambridge, UK
Andrei Popescu	University of Middlesex, UK
Gert Smolka	Saarland University, Germany
Matthieu Sozeau	Inria Paris, France
René Thiemann	University of Innsbruck, Austria
Laurent Théry	Inria Sophia Antipolis – Méditerranée, France
Andrew Tolmach	Portland State University, USA
Christian Urban	King's College London, UK
Viktor Vafeiadis	MPI-SWS, Germany

Additional Reviewers

Bahr, Patrick	Chau, Cuong Kim
Cohen, Cyril	Contejean, Evelyne
Doczkal, Christian	Felgenhauer, Bertram
Färber, Michael	Giero, Mariusz
Goel, Shilpi	Heule, Marijn J.H.
Hölzl, Johannes	Imine, Abdessamad
Immler, Fabian	Jimenez, Raul Pardo
Joosten, Sebastiaan	Krebbers, Robbert
Lammich, Peter	Li, Wenda
Mahmoud, Mohamed Yousri	Maric, Ognjen
Mehta, Mihir	Narboux, Julien
Oster, Gérald	Schäfer, Steven
Selfridge, Benjamin	Sibut-Pinote, Thomas
Spitters, Bas	Sternagel, Christian
Struth, Georg	Tan, Yong Kiam
Tuerk, Thomas	Wenzel, Makarius
Wiedijk, Freek	Yamada, Akihisa

Sponsors

Aesthetic Integration
Communauté Urbaine du Grand Nancy
Microsoft Research
Région Alsace Champagne-Ardenne Lorraine
Springer International Publishing AG

Abstracts of Keynote Presentations

Propositions as Programs, Proofs as Programs

Viktor Kuncak

École Polytechnique Fédérale de Lausanne (EPFL)

Leon is a system that (among other features) enables writing verified programs and their properties in a purely functional subset of Scala. The key specification statement in Leon is that a function satisfies its contract for all inputs. Leon proves properties and finds counterexamples using SMT solvers and an unfolding strategy for recursive functions. A newly developed link with Isabelle provides an additional safety net for soundness of the approach.

Due to Leon's unfolding mechanism, it is possible to write additional, semantically redundant, expressions that help Leon prove a theorem. We attempt to formalize this "accidental" feature of Leon. In our view, propositions, as well as proofs, are just terminating programs. This makes Leon statements and proofs (syntactically) accessible to the half a million of Scala developers. We explain some limitations of this approach in writing proof tactics and controlling the space of assumptions, suggesting that a form of reflection would provide benefits of Turing-complete tactic language without ever leaving the paradise of purely functional Scala programs.

Formal Verification of Financial Algorithms, Progress and Prospects

Grant Olney Passmore

Aesthetic Integration and University of Cambridge

Many deep issues plaguing today's financial markets are symptoms of a fundamental problem: The complexity of algorithms underlying modern finance has significantly outpaced the power of traditional tools used to design and regulate them. At Aesthetic Integration, we've pioneered the use of formal verification for analysing the safety and fairness of financial algorithms. With a focus on financial infrastructure (e.g., the matching logics of exchanges and dark pools), we'll describe the landscape, and illustrate our Imandra formal verification system on a number of real-world examples. We'll sketch many open problems and future directions along the way.

Dijkstra Monads for Free: A Framework for Deriving and Extending F*'s Effectful Semantics

Nikhil Swamy

Microsoft Research

F* is a higher-order effectful language with dependent types. It aims to provide equal support for general purpose programming (as in the ML family of languages) as well as for developing formal proofs (like other type-theory based proof assistants, e.g., Coq, Agda or Lean). By making use of an SMT solver while type-checking, F* provides automation for many routine proofs.

At the heart of F* is the manner in which effects and dependent types are combined: this presents several well-known difficulties. Our basic approach to solving these difficulties is not surprising: effectful computations are encapsulated within monad-like structures. More specifically, F* interprets effectful computations using monads of predicate transformers, so called "Dijkstra monads" that compute weakest pre-conditions for arbitrary post-conditions. These Dijkstra monads are arranged in a lattice of effect inclusions, e.g., pure computations are included within stateful ones.

In this talk, I will describe a new technique for deriving F*'s Dijkstra monad lattice by CPS'ing (with result type Prop) purely functional definitions of monads corresponding to F*'s effects. Several benefits ensue:

1. For starters, programmers are able to customize F*'s effect lattice using familiar Haskell-style monadic definitions, while gaining for each such monad a weakest pre-condition calculus suitable for Hoare-style verification of programs.
2. Next, several useful properties, e.g., monotonicity of predicate transformers, are guaranteed by the derivation, reducing the proof obligations for adding an effect to F*.
3. Third, our technique supports a mechanism to break the abstraction of a monadic effect in a controlled manner, reifying an effectful computation as its pure counterpart, and reflecting pure reasoning on the reified program back on to the effectful code.

I will also provide a general introduction to F* and its applications, notably its use in Everest, a new project to build and deploy a verified, secure implementation of HTTPS, including Transport Layer Security, TLS-1.3.

F* is open source and developed on github by researchers at Microsoft Research and Inria. For more information, visit `https://fstar-lang.org`.

Contents

Regular Contributions

Rough Diamonds