Lecture Notes in Computer Science 12345

Founding Editors

Gerhard Goos
Karlsruhe Institute of Technology, Karlsruhe, Germany
Juris Hartmanis
Cornell University, Ithaca, NY, USA

Editorial Board Members

Elisa Bertino
Purdue University, West Lafayette, IN, USA
Wen Gao
Peking University, Beijing, China
Bernhard Steffen
TU Dortmund University, Dortmund, Germany
Gerhard Woeginger
RWTH Aachen, Aachen, Germany
Moti Yung
Columbia University, New York, NY, USA

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

Wolfgang Ahrendt · Bernhard Beckert ·
Richard Bubel · Reiner Hähnle ·
Mattias Ulbrich (Eds.)

Deductive Software Verification: Future Perspectives

Reflections on the Occasion of 20 Years of KeY

Springer

Editors
Wolfgang Ahrendt
Chalmers University of Technology
Gothenburg, Sweden

Bernhard Beckert
Karlsruhe Institute of Technology
Karlsruhe, Germany

Richard Bubel
Technical University of Darmstadt
Darmstadt, Germany

Reiner Hähnle
Technical University of Darmstadt
Darmstadt, Germany

Mattias Ulbrich
Karlsruhe Institute of Technology
Karlsruhe, Germany

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-030-64353-9 ISBN 978-3-030-64354-6 (eBook)
https://doi.org/10.1007/978-3-030-64354-6

LNCS Sublibrary: SL2 – Programming and Software Engineering

© Springer Nature Switzerland AG 2020
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

A little more than 20 years ago, a group of researchers started out with what became the KeY project. Back then, KeY was one of the early efforts in an emerging community, whose common agenda was to transform deductive verification methods from a largely theoretical endeavor, mostly applied to academic (class room) examples, to techniques and tools that enable effective verification of software developed in widely used programming languages and paradigms, for realistic applications. Indeed, the theoretical underpinnings that had evolved in the 30 years before that, since the late 60s, were vital for the success for this new movement. But the new focus also came with new scientific challenges.

Today, we can say that the more applied focus of the last two decades in deductive verification required not only strong efforts in efficiency and usability of verification tools, but also extensive further development on the theory side. In short, to become more practical, we needed more theory, and continue to do so. To master the concepts in contemporary languages, paradigms, and scenarios (aliasing, exceptions, concurrency, communication, reuse, extendability, dynamic configuration, to name a few), the community had to find new ways to precisely model and efficiently process these phenomena and artifacts, with a focus on applicability and usability. For a more comprehensive exposition of these developments, we refer to "Deductive Software Verification: From Pen-and-Paper Proofs to Industrial Tools" by Reiner Hähnle and Marieke Huisman.[1]

During the deductive verification journey in the past 20 years, the KeY project was only one "means of transport", but a steady one, with strong links to other actors in the community. Therefore, we took the 20 year anniversary of KeY as an opportunity to invite researchers, inside and outside of the project, to contribute to a book capturing some state-of-the-art developments in the field. We feel very lucky that so many of our peers responded. The result is a book whose chapters identify and address some of the latest challenges of deductive software verification. It captures aspects of verification tool development, the integration of different verification techniques, efficiency and usability of verification methods, novel contracts for specification and verification, and history.

Part I, "History," contains a single chapter: "A Short History of KeY." Peter H. Schmitt gives an overview of the beginning and first decade of the project. This is not merely interesting for readers with a special interest in KeY. Instead, it is a rare glimpse into and a reflection on the turns, successes, and also failures, which a long term research project experienced, when coming into being and growing from there, in particular in a (then) new landscape.

[1] In *Computing and Software Science: State of the Art and Perspectives*, Bernhard Steffen and Gerard Woeginger (Eds.), LNCS 10000, Springer 2019.

Part II, "Verification Tools," starts with a chapter[2] which also takes a historical viewpoint. It offers a retrospective on the development of the KeYmaera family of provers for hybrid system verification. The second chapter[3] of this part focuses on performance bottlenecks, and their improvement, in the context of the VerCors tool for the verification of parallel programs. Altogether, this part demonstrates the importance of implementation considerations for the impact of verification methods.

Part III focuses on a central technique of deductive verification and, therefore, of this book: *Contracts*. This concept is the cornerstone of modularity, and thus essential in rendering deductive verification scalable. The chapters introduce behavioral contracts for cooperative scheduling[4], propagate the usage of abstract contracts for feature evolution and feature interaction[5], and discuss constraint-based contract inference[6]. Another chapter addresses a long-standing issue in the verification of concurrent applications, by moving from explicit to implicit dynamic frames.[7] The final chapter[8] in this part discusses deductive verification in a very new application area, called smart contracts. Even if smart contracts are rather programs than specification contracts, they realize – and enforce – a contract between the users who chose to engage with it.

It is mandatory for a book on contemporary deductive software verification to include feasibility and usability aspects, which Part IV focuses on. The field has grown out of the analysis of toy examples and is now looking at complex properties of real-world programs. The first chapter[9] provides a tutorial for the challenging verification of the linked list data structure in the Java Collection Framework. The second chapter[10] describes a long term, collaborative challenge named VerifyThis, which calls upon the verification community to verify aspects of realistic software over a longer period, with the ultimate goal to facilitate the application of formal analysis in the software development process. The third chapter[11] presents suggestions to improve usability and user guidance in interactive deductive verification.

Finally, Part V discusses two instances of integration of verification techniques. First, it presents an integration of static and dynamic analysis applied to the classical security problem of noninterference.[12] Second, it presents a novel integration of partial

---

[2] "A Retrospective on Developing Hybrid System Provers in the KeYmaera Family," by Mitsch and Platzer.

[3] "Improving Performance of the VerCors Program Verifier," by Mulder et al.

[4] "Behavioral Contracts for Cooperative Scheduling," by Kamburjan et al.

[5] "Using Abstract Contracts for Verifying Evolving Features and Their Interactions," by Knüppel et al.

[6] "Constraint-based Contract Inference for Deductive Verification," by Alshnakat et al.

[7] "From Explicit to Implicit Dynamic Frames in Concurrent Reasoning for Java," by Mostowski.

[8] "Formal Analysis of Smart Contracts: Applying the KeY System," by Ahrendt et al.

[9] "A Tutorial on Verifying LinkedList using KeY," by Hiep et al.

[10] "VerifyThis! – The Collaborative Long-term Challenge," by Huisman et al.

[11] "Usability Recommendations for User Guidance in Deductive Program Verification," by Grebing and Ulbrich.

[12] "Integration of Static and Dynamic Analysis Techniques for Checking Noninterference," by Beckert et al.

order reduction (an optimization principle hugely successful in model checking) with symbolic execution (which lies at the heart of KeY style deductive verification).[13]

We thank all authors for accepting our invitation and putting a lot of effort into producing the high-quality content we are proud to present here. With this book, we hope to provide the community with insights into recent and latest developments in important subareas of contemporary deductive software verification.

For many years Alfred Hofmann, who retired in 2019, handled the KeY-related LNCS volumes at Springer, including the present one. The editors and the whole KeY team would like to express their gratitude for his constant support, always smooth communication, and many constructive suggestions.

July 2020

Wolfgang Ahrendt
Bernhard Beckert
Richard Bubel
Reiner Hähnle
Mattias Ulbrich

[13] "SymPaths: Symbolic Execution Meets Partial Order Reduction," by De Boer et al.

Organization

Editors

Wolfgang Ahrendt	Chalmers University of Technology, Sweden
Bernhard Beckert	Karlsruhe Institute of Technology (KIT), Germany
Richard Bubel	Technische Universität Darmstadt, Germany
Reiner Hähnle	Technische Universität Darmstadt, Germany
Mattias Ulbrich	Karlsruhe Institute of Technology (KIT), Germany

Reviewers

Marcello Bonsangue	Leiden University, The Netherlands
Frank S. de Boer	Centrum Wiskunde & Informatica, The Netherlands
Stijn de Gouw	The Open University, The Netherlands
Crystal Chang Din	University of Oslo, Norway
Sarah Grebing	Karlsruhe Institute of Technology (KIT), Germany
Dilian Gurov	KTH Royal Institute of Technology, Sweden
Mihai Herda	Karlsruhe Institute of Technology (KIT), Germany
Hans-Dieter A. Hiep	Centrum Wiskunde & Informatica, The Netherlands
Marieke Huisman	University of Twente, The Netherlands
Einar Broch Johnsen	University of Oslo, Norway
Sebastiaan Joosten	Dartmouth College, USA
Eduard Kamburjan	Technische Universität Darmstadt, Germany
Michael Kirsten	Karlsruhe Institute of Technology (KIT), Germany
Alexander Knüppel	Technische Universität Braunschweig, Germany
Stefan Mitsch	Carnegie Mellon University, USA
Wojciech Mostowski	Halmstad University, Sweden
Henk Mulder	University of Twente, The Netherlands
André Platzer	Carnegie Mellon University, USA
Violet Ka I Pun	Western Norway University of Applied Sciences, Norway
Philipp Rümmer	Uppsala University, Sweden
Ina Schaefer	Technische Universität Braunschweig, Germany
Jonas Schiffl	Karlsruhe Institute of Technology (KIT), Germany
Peter Schmitt	Karlsruhe Institute of Technology (KIT), Germany
Silvia Lizeth Tapia Tarifa	University of Oslo, Norway
Thomas Thüm	Ulm University, Germany
Lars Tveito	University of Oslo, Norway

Contents