Lecture Notes in Computer Science 12740

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

Frédéric Loulergue · Franz Wotawa (Eds.)

Tests and Proofs

15th International Conference, TAP 2021
Held as Part of STAF 2021
Virtual Event, June 21–22, 2021
Proceedings

Springer

Editors
Frédéric Loulergue
Université d'Orléans
Orléans, France

Franz Wotawa
Graz University of Technology
Graz, Austria

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-030-79378-4 ISBN 978-3-030-79379-1 (eBook)
https://doi.org/10.1007/978-3-030-79379-1

LNCS Sublibrary: SL2 – Programming and Software Engineering

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

Preface

This volume contains the papers accepted for the 15th International Conference on Tests and Proofs (TAP 2021), originally planned to be held during June 21–22, 2021, in Bergen, Norway, as part of Software Technologies: Applications and Foundations (STAF), a federation of some of Europe's leading conferences on software technologies. Due to the COVID-19 pandemic, STAF and TAP were held online, and the TAP 2021 conference featured presentations of papers accepted at TAP 2020 and published in LNCS volume 12165, as well as the presentations of the papers published in this volume.

The TAP conference promotes research in verification and formal methods that targets the interplay of proofs and testing: the advancement of techniques of each kind and their combination, with the ultimate goal of improving software and system dependability.

Research in verification has seen a steady convergence of heterogeneous techniques and a synergy between the traditionally distinct areas of testing (and dynamic analysis) and of proving (and static analysis). Formal techniques for counter-example generation based on, for example, symbolic execution, SAT/SMT-solving or model checking, furnish evidence for the potential of a combination of test and proof. The combination of predicate abstraction with testing-like techniques based on exhaustive enumeration opens the perspective for novel techniques of proving correctness. On the practical side, testing offers cost-effective debugging techniques of specifications or crucial parts of program proofs (such as invariants). Last but not least, testing is indispensable when it comes to the validation of the underlying assumptions of complex system models involving hardware or system environments. Over the years, there is growing acceptance in research communities that testing and proving are complementary rather than mutually exclusive techniques.

TAP 2021 received 13 abstracts that led to 10 submissions out of which we accepted 6 papers after review and discussion with the Program Committee (PC) members. The submissions came from authors in the following countries (in alphabetical order): France, Germany, India, Japan, New Zealand, Russia, Singapore, Sweden, the UK, and the USA. We thank the PC members and reviewers for doing an excellent job!

For the third time, TAP featured an artifact evaluation (AE) and three papers were awarded with AE badges. We thank the AE chairs, Daniel Dietsch (University of Freiburg, Germany) and Marie-Christine Jakobs (TU Darmstadt, Germany), for organizing artifact submission and evaluation, and the AE Committee members for thoroughly evaluating all artifacts.

This volume also contains two short abstracts: an abstract of the talk given by our invited speaker, Mohammad Mousavi (University of Leicester, UK), on "Learning About the Change: An Adaptive Approach to Automata Learning", and an abstract of our invited tutorial on Runtime Verification led by Martin Leucker (University of Lübeck, Germany).

We thank the organizing team of STAF in Bergen, in particular Adrian Rutle who had to deal with a very difficult situation. We also thank the publication team at Springer for their support. We hope that you will enjoy reading the volume.

May 2021

Frédéric Loulergue
Franz Wotawa

Organization

Program Committee Chairs

Frédéric Loulergue	Université d'Orléans, France
Franz Wotawa	Graz University of Technology, Austria

Program Committee

Wolfgang Ahrendt	Chalmers University of Technology, Sweden
Bernhard K. Aichernig	TU Graz, Austria
Dirk Beyer	LMU Munich, Germany
Achim D. Brucker	University of Exeter, UK
Simon Cruanes	Imandra, USA
Catherine Dubois	ENSIIE, France
Cédric Eichler	INSA Centre Val de Loire, France
Gordon Fraser	University of Passau, Germany
Alex Groce	Northern Arizona University, USA
Klaus Havelund	Jet Propulsion Laboratory, USA
Chantal Keller	Université Paris-Sud, France
Nikolai Kosmatov	CEA List, France
Martin Leucker	University of Lübeck, Germany
Karl Meinke	KTH Royal Institute of Technology, Sweden
Stephan Merz	Inria, France
Corina Pasareanu	NASA, USA
François Pessaux	ENSTA Paris, France
Ingo Pill	Silicon Austria Labs, Austria
Heike Wehrheim	University of Paderborn, Germany
Burkhart Wolff	Université Paris-Sud, France
Nina Yevtushenko	Ivannikov Institute for System Programming of RAS, Russia

Artifact Evaluation Committee Chairs

Marie-Christine Jakobs	TU Darmstadt, Germany
Danie Dietsch	University of Freiburg, Germany

Artifact Evaluation Committee

Patrice Clemente	INSA Centre Val de Loire, France
Simon Dierl	TU Dortmund, Germany
Mathias Fleury	Johannes Kepler University Linz, Austria
Michael Foster	University of Sheffield, UK

Ákos Hajdu	Budapest University of Technology and Economics, Hungary
Jan Haltermann	Paderborn University, Germany
Marcel Hark	RWTH Aachen University, Germany
Sean Kauffman	Aalborg University, Denmark
Sven Linker	Lancaster University (Leipzig), Germany
Cyrus Liu	Stevens Institute of Technology, USA
Marco Muniz	Aalborg University, Denmark
Yakoub Nemouchi	University of York, UK
Virgile Robles	CEA List, France
Martin Sachenbacher	University of Lübeck, Germany
Christian Schilling	University of Konstanz, Germany
Martin Tappler	Graz University of Technology, Austria
Nico Weise	Ludwig-Maximilians-Universität München, Germany

Steering Committee

Bernhardt K. Aichernig	TU Graz, Austria
Jasmin Blanchette	Vrije Universiteit Amsterdam, Netherlands
Achim D. Brucker	University of Sheffield, UK
Catherine Dubois (Chair)	ENSIIE, France
Martin Gogolla	University of Bremen, Germany
Nikolai Kosmatov	CEA, France
Burkhart Wolff	LRI, France

Additional Reviewers

Eduard Kamburjan
Karam Kharraz
Ashfaq Hussain Farooqui
Delphine Longuet

Abstracts of Invited Events

Learning About the Change: An Adaptive Approach to Automata Learning

Mohammad Reza Mousavi

University of Leicester, UK

Automata learning is a technique to learn behavioural models from black-box systems. Variability and evolution are inherent to much of the modern autonomous systems and hence, new sorts of automata learning techniques are needed to learn about variability-intensive and evolving systems. In this talk, we first present the basic principles of automata learning and then report on two novel techniques for learning variability-annotated models as well as efficient learning for evolving systems by identifying the commonalities and differences in the learning process.

This talk is based on joint work with several people, and in particular, with Diego Damasceno and Adenilso Simao.

Testing, Runtime Verification and Automata Learning

Martin Leucker

University of Lübeck, Germany

Testing and runtime verification are both verification techniques for checking whether a system is correct. The essential artefacts for checking whether the system is correct are actual executions of the system, formally words. Such a set of words should be representative for the systems behavior.

In the field of automata learning (or grammatical inference) a formal model of a system is derived based on exemplifying behavior. In other words, the question is addressed what model fits to a given set of words.

In testing, typically, the system under test is examined on a finite set of test cases, formally words, which may be derived manually or automatically. Oracle-based testing is a form of testing in which an oracle, typically a manually developed piece of code, is attached to the system under test and employed for checking whether a given set of test cases passes or fails.

In runtime verification, typically, a formal specification of the correct behavior is given from which a so-called monitor is synthesised and used for examining whether the behavior of the system under test, or generally the system to monitor, adheres to such a specification. In a sense, the monitor acts as a test oracle, when employed in testing.

From the discussion above we see that testing, runtime verification, and learning automata share similarities but also differences. The main artefacts used for the different methods are formal specifications, models like automata, but especially sets of words, on which the different system descriptions are compared, to eventually obtain a verdict whether the system under test is correct or not.

In this tutorial we recall the basic ideas of testing, oracle-based testing, model-based testing, conformance testing, automata learning and runtime verification and elaborate on a coherent picture with the above mentioned artefacts as ingredients. We mostly refrain from technical details but concentrate on the big picture of those verification techniques.

Contents