Lecture Notes in Computer Science 12399

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

Jyotirmoy Deshmukh · Dejan Ničković (Eds.)

Runtime Verification

20th International Conference, RV 2020
Los Angeles, CA, USA, October 6–9, 2020
Proceedings

Springer

Editors
Jyotirmoy Deshmukh
University of Southern California
Los Angeles, CA, USA

Dejan Ničković
AIT Austrian Institute of Technology GmbH
Vienna, Austria

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-030-60507-0 ISBN 978-3-030-60508-7 (eBook)
https://doi.org/10.1007/978-3-030-60508-7

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 refereed proceedings of the 20th International Conference on Runtime Verification (RV 2020), held virtually October 6–9, 2020.

The RV series is a sequence of annual meetings that brings together scientists from both academia and industry interested in investigating novel lightweight formal methods to monitor, analyze, and guide the runtime behavior of software and hardware systems. Runtime verification techniques are crucial for system correctness, reliability, and robustness; they provide an additional level of rigor and effectiveness compared to conventional testing, and are generally more practical than exhaustive formal verification. Runtime verification can be used prior to deployment, for testing, verification, and debugging purposes, and after deployment for ensuring reliability, safety, and security, for providing fault containment and recovery, as well as online system repair.

RV started in 2001 as an annual workshop and turned into a conference in 2010. The workshops were organized as satellite events of established forums, including the Conference on Computer-Aided Verification and ETAPS. The proceedings of RV from 2001 to 2005 were published in the *Electronic Notes in Theoretical Computer Science*. Since 2006, the RV proceedings have been published in Springer's *Lecture Notes in Computer Science*. Previous RV conferences took place in Istanbul, Turkey (2012); Rennes, France (2013); Toronto, Canada (2014); Vienna, Austria (2015); Madrid, Spain (2016); Seattle, USA (2017); Limassol, Cyprus (2018); and Porto, Portugal (2019).

In 2020, RV celebrated its 20th edition, and to mark this occasion, the conference had a couple of new initiatives. The first initiative was to invite researchers from a special focus area to submit papers; the focus area for RV 2020 was "Runtime Verification for Autonomy." The second initiative was a panel discussion on RV for Autonomy, which invited selected prominent researchers from academia and practitioners from industry to serve as panelists. The panel focused on the role of runtime verification in the emerging field of autonomous systems, highlighting the theoretical and technical challenges and presenting potential opportunities.

This year we received 43 submissions, 27 as regular contributions, and 16 as short, tool, or benchmark papers. Each of these submissions went through a rigorous single-blind review process, as a result of which most papers received four reviews and all papers received at least three review reports. The committee selected 23 contributions, 14 regular and 9 short/tool/benchmark papers for presentation during the conference and inclusion in these proceedings. The evaluation and selection process involved thorough discussions among the members of the Program Committee and external reviewers through the EasyChair conference manager, before reaching a consensus on the final decisions.

The conference featured three keynote speakers:

– Katherine Driggs-Campbell, University of Illinois at Urbana-Champaign, USA
– Lane Desborough, Nudge BG, Inc., USA
– Thomas Henzinger, IST Austria, Austria

The conference included five tutorials on the first day, including one invited tutorial and four other tutorials selected to cover a variety of topics relevant to RV:

– Laura Nenzi, Ezio Bartocci, Luca Bortolussi, Michele Loreti, and Ennio Visconti presented the invited tutorial on "Monitoring Spatio-Temporal Properties"
– Yanhong A. Liu and Scott D. Stoller presented a tutorial on "Assurance of Distributed Algorithms and Systems: Runtime Checking of Safety and Liveness"
– Joshua Heneage Dawes, Marta Han, Omar Javed, Giles Reger, Giovanni Franzoni, and Andreas Pfeiffer presented a tutorial on "Analysing the Performance of Python-based Web Services with the VyPR Framework"
– Maximilian Schwenger presented a tutorial on "Monitoring Cyber-Physical Systems: From Design to Integration"
– Klaus Havelund and Doron Peled presented a tutorial on "BDDs for Representing Data in Runtime Verification"

The 2020 RV Test of Timed Award was given to Nicholas Nethercote and Julian Seward for their RV 2003 seminal paper "Valgrind: A Program Supervision Framework" on the dynamic analysis of programs.

RV 2020 is the result of the combined efforts of many individuals to whom we are deeply grateful. In particular, we thank the Program Committee members and sub-reviewers for their accurate and timely reviewing, all authors for their submissions, and all attendees of the conference for their participation. We thank Houssam Abbas for helping us organize the poster session. We are very grateful to RV sponsor Toyota Research Institute, USA, and Springer who provided an award for the best RV paper. We thank the RV Steering Committee for their support.

October 2020

Jyotirmoy Deshmukh
Dejan Ničković

Organization

RV Program Chairs

Jyotirmoy Deshmukh	University of Southern California, USA
Dejan Ničković	Austrian Institute of Technology (AIT), Austria

RV Poster and Demo Chair

Houssam Abbas	Oregon State University, USA

RV Program Committee

Houssam Abbas	Oregon State University, USA
Wolfgang Ahrendt	Chalmers University of Technology, Sweden
Ezio Bartocci	Vienna University of Technology, Austria
Nicolas Basset	Verimag, France
Domenico Bianculli	University of Luxembourg, Luxembourg
Borzoo Bonakdarpour	Michigan State University, USA
Chih-Hong Cheng	Denso, Germany
Katherine Driggs Campbell	University of Illinois at Urbana-Champaign, USA
Georgios Fainekos	Arizona State University, USA
Ylies Falcone	University of Grenoble Alpes, Inria, France
Chuchu Fan	Massachusetts Institute of Technology, USA
Lu Feng	University of Virginia, USA
Thomas Ferrère	Imagination Technologies, UK
Bernd Finkbeiner	Saarland University, Germany
Sebastian Fischmeister	University of Waterloo, Canada
Dana Fisman	Ben-Gurion University, Israel
Adrian Francalanza	University of Malta, Malta
Radu Grosu	Vienna University of Technology, Austria
Sylvain Hallè	Universitè du Quèbec à Chicoutimi, Canada
Klaus Havelund	NASA JPL, USA
Stefan Jakšić	Austrian Institute of Technology (AIT), Austria
Violet Ka I. Pun	Western Norway University of Applied Sciences, Norway
Jim Kapinski	Amazon, USA
Safraz Khurshid	The University of Texas at Austin, USA
Bettina Könighofer	TU Graz, Austria
Martin Leucker	University of Lübeck, Germany
Chung-Wei Lin	National Taiwan University, Taiwan

Additional Reviewers

RV Steering Committee

Insup Lee	University of Pennsylvania, USA
Martin Leucker	University of Lübeck, Germany
Giles Reger	The University of Manchester, UK
Grigore Rosu	University of Illinois at Urbana-Champaign, USA
Oleg Sokolsky	University of Pennsylvania, USA

Abstracts of Invited Presentations

Fantastic Failures and Where to Find Them: Designing Trustworthy Autonomy

Katherine Driggs-Campbell

University of Illinois at Urbana-Champaign, USA

Abstract. Autonomous robots are becoming tangible technologies that will soon impact the human experience. However, the desirable impacts of autonomy are only achievable if the underlying algorithms are robust to real-world conditions and are effective in (near) failure modes. This is often challenging in practice, as the scenarios in which general robots fail are often difficult to identify and characterize. In this talk, we'll discuss how to learn from failures to design robust interactive systems and how we can exploit structure in different applications to efficiently find and classify failures. We'll showcase both our failures and successes on autonomous vehicles and agricultural robots in real-world settings.

The Physical Side of Cyber-Physical Systems

Lane Desborough

Nudge BG, Inc., USA

Abstract. When our commercial reach exceeds our technical grasp, it is imperative that we advance our knowledge, that we embrace approaches to manage complexity, lest that complexity introduce undesired emergent properties. These complexity management approaches may seem new or novel, yet they rarely are. As science fiction author William Gibson is wont to say, "The future is already here, it just hasn't been evenly distributed yet."

Chemical engineering process control has afforded me a career spanning five continents and five industries. Although my current focus is the "artificial pancreas" – automated insulin delivery for people living with insulin-requiring diabetes – I have been privileged to be exposed to some of the most complex and challenging cyber-physical systems in the world; systems upon which society depends.

Most industries exist within their own bubble; exclusionary languages and pedagogy successfully defend their domains from new ideas. As one who has traversed many industries and worked on scores of industrial systems, a variety of personal, visceral experiences have allowed me to identify patterns and lessons applicable more broadly, perhaps even to your domain. Using examples drawn from petrochemical production, oil refining, power generation, industrial automation, and chronic disease management, I hope to demonstrate the need for, and value of, real-time verification.

Monitorability Under Assumptions

Thomas A. Henzinger and N. Ege Saraç

IST Austria, Klosterneuburg, Austria
{tah,ege.sarac}@ist.ac.at

Abstract. We introduce the monitoring of trace properties under assumptions. An assumption limits the space of possible traces that the monitor may encounter. An assumption may result from knowledge about the system that is being monitored, about the environment, or about another, connected monitor. We define monitorability under assumptions and study its theoretical properties. In particular, we show that for every assumption A, the boolean combinations of properties that are safe or co-safe relative to A are monitorable under A. We give several examples and constructions on how an assumption can make a non-monitorable property monitorable, and how an assumption can make a monitorable property monitorable with fewer resources, such as integer registers.

This research was supported in part by the Austrian Science Fund (FWF) under grant Z211-N23 (Wittgenstein Award).

Contents

Runtime Verification for Software

Runtime Verification with Temporal Logic Specifications