Lecture Notes in Computer Science **11187**

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

Editorial Board

David Hutchison
Lancaster University, Lancaster, UK
Takeo Kanade
Carnegie Mellon University, Pittsburgh, PA, USA
Josef Kittler
University of Surrey, Guildford, UK
Jon M. Kleinberg
Cornell University, Ithaca, NY, USA
Friedemann Mattern
ETH Zurich, Zurich, Switzerland
John C. Mitchell
Stanford University, Stanford, CA, USA
Moni Naor
Weizmann Institute of Science, Rehovot, Israel
C. Pandu Rangan
Indian Institute of Technology Madras, Chennai, India
Bernhard Steffen
TU Dortmund University, Dortmund, Germany
Demetri Terzopoulos
University of California, Los Angeles, CA, USA
Doug Tygar
University of California, Berkeley, CA, USA
Gerhard Weikum
Max Planck Institute for Informatics, Saarbrücken, Germany

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

Bernd Fischer · Tarmo Uustalu (Eds.)

Theoretical Aspects of Computing – ICTAC 2018

15th International Colloquium
Stellenbosch, South Africa, October 16–19, 2018
Proceedings

Springer

Editors
Bernd Fischer
Stellenbosch University
Stellenbosch, South Africa

Tarmo Uustalu
Reykjavík University
Reykjavik, Iceland

and

Tallinn University of Technology
Tallinn, Estonia

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-030-02507-6 ISBN 978-3-030-02508-3 (eBook)
https://doi.org/10.1007/978-3-030-02508-3

Library of Congress Control Number: 2018957486

LNCS Sublibrary: SL1 – Theoretical Computer Science and General Issues

© Springer Nature Switzerland AG 2018
Chapter "Layer Systems for Confluence—Formalized" is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/). For further details see license information in the chapter.
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, express 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

This volume is the proceedings of the 15th International Colloquium on Theoretical Aspects of Computing, ICTAC 2018, which was held in Stellenbosch, South Africa, during October 16–19, 2018, in colocation with the 14th African Conference on Research in Computer Science and Applied Mathematics, CARI 2018, October 14–16, and a jointly organized CARI/ICTAC spring school, October 12–15.

Established in 2004 by the International Institute for Software Technology of the United Nations University (UNU-IIST), the ICTAC conference series aims at bringing together researchers and practitioners from academia, industry, and government to present research and exchange ideas and experience addressing challenges in both theoretical aspects of computing and the exploitation of theory through methods and tools for system development. ICTAC also specifically aims to promote research cooperation between developing and industrial countries.

The topics of the conference include, but are not limited to, languages and automata; semantics of programming languages; logic in computer science; lambda calculus, type theory and category theory; domain-specific languages; theories of concurrency and mobility; theories of distributed, grid and cloud computing; models of objects and components; coordination models; models of software architectures; timed, hybrid, embedded, and cyber-physical systems; static analysis; software verification; software testing; program generation and transformation; model checking and automated theorem proving; interactive theorem proving; certified software, formalized programming theory.

Previous editions of ICTAC were held in Guiyang, China (2004), Hanoi, Vietnam (2005 and 2017), Tunis, Tunisia (2006), Macau (2007), Istanbul, Turkey (2008), Kuala Lumpur, Malaysia (2009), Natal, Brazil (2010), Johannesburg, South Africa (2011), Bangalore, India (2012), Shanghai, China (2013), Bucharest, Romania (2014), Cali, Colombia (2015), Taipei, Taiwan (2016). The proceedings of all these events were published in the LNCS series.

The program of ICTAC 2018 consisted of four invited talks and 25 contributed papers. We were proud to have as invited speakers Yves Bertot (Inria, France), Thomas Meyer (University of Cape Town, South Africa), Gennaro Parlato (University of Southampton, UK), and Peter Thiemann (Universität Freiburg, Germany). The talks of Meyer and Parlato are represented in this volume by abstracts, those by Bertot and Thiemann by an extended abstract and a paper.

The contributed papers were selected from among the 59 full submissions that we received in response to our call. Each of those was reviewed by at least three, and on average 3.4, Program Committee members or external reviewers. The Program Committee consisted of 28 researchers from academia and industry and from every continent.

The CARI/ICTAC spring school program consisted of seven half-day tutorials, taught by Yves Bertot, Vincent Cheval (Inria, France), Martin Leucker (Universität zu Lübeck, Germany), Thomas Meyer, Ina Schaefer (Technische Universität Braunschweig, Germany) with Loek Cleophas (Technische Universiteit Eindhoven, The Netherlands), Peter Thiemann, and Willem Visser (Stellenbosch University, South Africa).

We are grateful to all our invited speakers, submission authors, Program Committee members, and external reviewers for their contributions to the program, to the Steering Committee and especially its chair, Ana Cavalcanti, for advice, to Easychair for the platform for Program Committee work, and to the LNCS editorial team for producing this volume and for donating the best paper award money. We are thankful to the Stellenbosch Institute for Advanced Study (STIAS) for lending the premises, and to Hayley Du Plessis and Andrew Collett for administrative and technical support. Stellenbosch University, IFIP TC6 and DEC, Inria, and their partnering French agencies provided financial support toward the costs of the invited speakers and tutorialists.

August 2018

Bernd Fischer
Tarmo Uustalu

Organization

Steering Committee

Ana Cavalcanti	University of York, UK
Martin Leucker	Universität zu Lübeck, Germany
Zhiming Liu	Southwest University, China
Tobias Nipkow	Technische Universität München, Germany
Augusto Sampaio	Universidade Federal de Pernambuco, Brazil
Natarajan Shankar	SRI International, USA

General Chair

Bernd Fischer	Stellenbosch University, South Africa

Program Chairs

Bernd Fischer	Stellenbosch University, South Africa
Tarmo Uustalu	Reykjavík University, Iceland, and Tallinn University of Technology, Estonia

Program Committee

June Andronick	Data61, Australia
Éric Badouel	IRISA, France
Eduardo Bonelli	Universidad Nacional de Quilmes, Argentina
Ana Cavalcanti	University of York, UK
Dang Van Hung	VNU University of Engineering and Technology, Vietnam
Uli Fahrenberg	LIX, France
Anna Lisa Ferrara	University of Southampton, UK
Adrian Francalanza	University of Malta, Malta
Edward Hermann Haeusler	Pontifícia Universidade Católica do Rio de Janeiro, Brazil
Ross Horne	Nanyang Technological University, Singapore
Atsushi Igarashi	Kyoto University, Japan
Jan Křetinský	Technische Universität München, Germany
Martin Leucker	Universität zu Lübeck, Germany
Zhiming Liu	Southwest University, China
Radu Mardare	Aalborg University, Denmark
Tobias Nipkow	Technische Universität München, Germany
Maciej Piróg	University of Wrocław, Poland
Sanjiva Prasad	IIT Delhi, India

Murali Krishna Ramanathan	Uber Technologies, USA
Camilo Rueda	Pontificia Universidad Javeriana Cali, Colombia
Augusto Sampaio	Universidade Federal de Pernambuco, Brazil
Ina Schaefer	Technische Universität Braunschweig, Germany
Natarajan Shankar	SRI International, USA
Georg Struth	University of Sheffield, UK
Cong Tian	Xidian University, China
Lynette van Zijl	Stellenbosch University, South Africa

Additional Reviewers

Abdulrazaq Abba
Antonis Achilleos
Leonardo Aniello
Jaime Arias
S. Arun-Kumar
Pranav Ashok
Duncan Attard
Mauricio Ayala-Rincón
Giorgio Bacci
Giovanni Bacci
Joffroy Beauquier
Giovanni Bernardi
Silvio Capobianco
Ian Cassar
Sheng Chen
Lukas Convent
Alejandro Díaz-Caro
Eric Fabre
Nathanaël Fijalkow
Robert Furber
Ning Ge
Jeremy Gibbons
Stéphane Graham-Lengrand
Reiko Heckel
Willem Heijltjes
Wu Hengyang
Bengt-Ove Holländer
Juliano Iyoda
Mauro Jaskelioff
Yu Jiang
Christian Johansen
Dejan Jovanovic
Karam Kharraz
Hélène Kirchner
Alexander Knüppel
Jérémy Ledent
Karoliina Lehtinen
Benjamin Martin
Tobias Meggendorfer
Carroll Morgan
Madhavan Mukund
Kedar Namjoshi
Michael Nieke
Sidney C. Nogueira
Carlos Olarte
Marcel Vinicius Medeiros Oliveira
Hugo Paquet
Mathias Ruggaard
André Pedro
Gustavo Petri
Mathias Preiner
Adrian Puerto Aubel
Karin Quaas
Andrew Reynolds
Pedro Ribeiro
James Riely
Camilo Rocha
Nelson Rosa
Martin Sachenbacher
Gerardo M. Sarria M.
Torben Scheffel
Alexander Schlie
Malte Schmitz
Sven Schuster

Thomas Sewell
René Thiemann
Daniel Thoma
Thomas Thüm
Ashish Tiwari
Hazem Torfah
Szymon Toruńczyk
Dmitriy Traytel
Christian Urban
Frank Valencia
Maximilian Weininger
Pengfei Yang
Hengjun Zhao

Organizing Committee

Bernd Fischer	Stellenbosch University, South Africa
Katarina Britz	Stellenbosch University, South Africa
Hayley Du Plessis	Stellenbosch University, South Africa

Host Institution

Stellenbosch University Computer Science Division

Sponsors

Stellenbosch University
Springer
IFIP Technical Committee 6 and Digital Equity Committee
Inria
Agence universitaire de la Francophonie (AUF)
Centre de coopération internationale en recherche agronomique pour le développement (CIRAD)
Institut de recherche pour le développement (IRD)

Invited Talks (Abstracts)

What Is Knowledge Representation and Reasoning?

Thomas Meyer

Department of Computer Science and Centre for Artificial Intelligence Research, University of Cape Town, Private Bag X3, Rondebosch 7701, South Africa
tmeyer@cs.uct.ac.za

Artificial Intelligence (AI) is receiving lots of attention at the moment, with all kinds of wild speculation in the media about its potential benefits. The excitement is mostly about recent successes in the subarea of AI known as Machine Learning. The current hype is reminiscent of the scenario about 20 years ago when logic-based AI, and more specifically, the subarea known as Knowledge Representation, had everyone in a state of euphoria about the future of AI.

My focus in this talk is on Knowledge Representation. I first provide an overview of the field as a whole, followed up by a more detailed presentation about some of the successful Knowledge Representation techniques and tools. The presentation is augmented with a discussion on the strengths and limitations of the Knowledge Representation approach to AI. Finally, I offer some thoughts on the recently revitalised suggestion that a combination of Knowledge Representation and Machine Learning techniques can lead to further advances in AI.

Finding Rare Concurrent Programming Bugs: An Automatic, Symbolic, Randomized, and Parallelizable Approach

Gennaro Parlato

School of Electronics and Computer Science,
University of Southampton, Highfield, Southampton SO17 1BJ, UK
gennaro@ecs.soton.ac.uk

Developing correct, scalable and efficient concurrent programs is a complex and difficult task, due to the large number of possible concurrent executions that need to be taken into account. Modern multi-core processors with weak memory models and lock-free algorithms make this task even more difficult, as they introduce additional executions that confound the developers' reasoning. Because of these complex interactions, concurrent programs often contain bugs that are difficult to find, reproduce, and fix. Stress testing is known to be very ineffective in detecting rare concurrency bugs as all possible executions of the programs have to be explored explicitly. Consequently, testing by itself is often inadequate for concurrent programs and needs to be complemented by automated analysis tools that enable detection of bugs in a systematic and symbolic way.

In the first part of the talk, I provide an overview of Lazy-CSeq, a symbolic method based on Bounded Model Checking (BMC) and Sequentialization. Lazy-CSeq first translates a multi-threaded C program into a nondeterministic sequential C program that preserves reachability for all round-robin schedules with a given bound on the number of rounds. It then reuses existing high-performance BMC tools as backends for the sequential verification problem. This translation is carefully designed to introduce very small memory overheads and very few sources of nondeterminism, so that it produces tight SAT/SMT formulae, and is thus very effective in practice.

In the second part of the talk, I present Swarm-CSeq, which extends Lazy-CSeq with a swarm-based bug-finding method. The key idea is to generate a set of simpler program instances, each capturing a reduced set of the original programs interleavings. These instances can then be verified independently in parallel. Our approach is parametrizable and allows us to fine-tune the nondeterminism and randomness used for the analysis. In our experiments, by using parallel analysis, we show that this approach is able, even with a small number of cores, to find bugs in the hardest known concurrency benchmarks in a matter of minutes, whereas other dynamic and static tools fail to do so in hours.

Contents