

*Commenced Publication in 1973*

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

## Editorial Board Members

David Hutchison, UK

Josef Kittler, UK

Friedemann Mattern, Switzerland

Moni Naor, Israel

Bernhard Steffen, Germany

Doug Tygar, USA

Takeo Kanade, USA

Jon M. Kleinberg, USA

John C. Mitchell, USA

C. Pandu Rangan, India

Demetri Terzopoulos, USA

## Formal Methods

Subline of Lectures Notes in Computer Science

## Subline Series Editors

Ana Cavalcanti, *University of York, UK*

Marie-Claude Gaudel, *Université de Paris-Sud, France*

## Subline Advisory Board

Manfred Broy, *TU Munich, Germany*

Annabelle McIver, *Macquarie University, Sydney, NSW, Australia*

Peter Müller, *ETH Zurich, Switzerland*

Erik de Vink, *Eindhoven University of Technology, The Netherlands*

Pamela Zave, *AT&T Laboratories Research, Bedminster, NJ, USA*


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


Maurice H. ter Beek · Annabelle McIver ·  
José N. Oliveira (Eds.)


# Formal Methods – The Next 30 Years

Third World Congress, FM 2019  
Porto, Portugal, October 7–11, 2019  
Proceedings

*Editors*

Maurice H. ter Beek   
Consiglio Nazionale delle Ricerche  
Pisa, Italy

Annabelle McIver   
Macquarie University  
Sydney, NSW, Australia

José N. Oliveira   
University of Minho  
Braga, Portugal

ISSN 0302-9743                      ISSN 1611-3349 (electronic)  
Lecture Notes in Computer Science  
ISBN 978-3-030-30941-1              ISBN 978-3-030-30942-8 (eBook)  
<https://doi.org/10.1007/978-3-030-30942-8>

LNCS Sublibrary: SL2 – Programming and Software Engineering

© Springer Nature Switzerland AG 2019

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

This volume contains the papers presented at the 23rd Symposium on Formal Methods (FM 2019), held in Porto, Portugal, in the form of the Third World Congress on Formal Methods, during October 7–11, 2019. These proceedings also contain five papers selected by the Program Committee (PC) of the Industry Day (I-Day).

FM 2019 was organized under the auspices of Formal Methods Europe (FME), an independent association whose aim is to stimulate the use of, and research on, formal methods for software development. It has been more than 30 years since the first VDM symposium in 1987 brought together researchers with the common goal of creating methods to produce high-quality software based on rigor and reason. Since then the diversity and complexity of computer technology has changed enormously and the formal methods community has stepped up to the challenges those changes brought by adapting, generalizing, and improving the models and analysis techniques that were the focus of that first symposium. The theme for FM 2019, “The Next 30 Years,” was a reflection on how far the community has come and the lessons we can learn for understanding and developing the best software for future technologies.

To reflect the fact that it has been 20 years since FM 1999 in Toulouse and 10 years since FM 2009 in Eindhoven, FM 2019 was organized as a World Congress, and we composed a PC of renowned scientists from 42 different countries spread across all continents except for Antarctica. We originally received a stunning total of 185 abstract submissions, which unfortunately resulted in ‘only’ 129 paper submissions from 36 different countries. Each submission went through a rigorous review process in which 95% of the papers were reviewed by four PC members. Following an in-depth discussion phase lasting two weeks, we selected 37 full papers and 2 short tool papers, an acceptance rate of 30%, for presentation during the symposium and inclusion in these proceedings. The symposium featured keynotes by Shriram Krishnamurthi (Brown University, USA), Erik Poll (Radboud University, The Netherlands), and June Andronick (CSIRO-Data61 and UNSW, Australia). We hereby thank these invited speakers for having accepted our invitation. The program also featured a Lucas Award and FME Fellowship Award Ceremony.

We are grateful to all involved in FM 2019. In particular the PC members and subreviewers for their accurate and timely reviewing, all authors for their submissions, and all attendees of the symposium for their participation. We also thank all the other committees (I-Day, Doctoral Symposium, Journal First Track, Workshops, and Tutorials), itemized on the following pages, and particularly the excellent local organization and publicity teams. In addition to FM 2019 they also managed the FM week consisting of another 8 conferences, 17 workshops, and 7 tutorials, as well as ‘X’, the secret project of a colloquium in honor of Stefania Gnesi based on a Festschrift to celebrate her 65th birthday.

We are very grateful to our platinum sponsors: Amazon Web Services (AWS), Google, and Sony; our gold sponsors: Springer, Semmler, ASML, and PT-FLAD Chair

in Smart Cities & Smart Governance; our silver sponsors: Oracle Labs, Runtime Verification Inc., Standard Chartered, GMV, United Technologies Research Center (UTRC), and Efacec; our bronze sponsors i2S, Foundations of Perspicuous Software Systems Collaborative Research Center, and the Mathematical research center of the University of Porto (CMUP); and our basic sponsors: Natixis and Neadvance.

Finally, we thank Springer for publishing these proceedings in their FM subline and we acknowledge the support from EasyChair in assisting us in managing the complete process from submissions to these proceedings to the program.

August 2019

Maurice H. ter Beek  
Annabelle McIver  
José N. Oliveira

# Organization

## General Chair

José N. Oliveira                      University of Minho and INESC TEC, Portugal

## FM Program Chairs

Maurice H. ter Beek                      ISTI–CNR, Italy  
Annabelle McIver                      Macquarie University, Australia

## Industry Day Chairs

Joe Kiniry                      Galois Inc., USA  
Thierry Lecomte                      ClearSy, France

## Doctoral Symposium Chairs

Alexandra Silva                      University College London, UK  
Antónia Lopes                      University of Lisbon, Portugal

## Journal First Track Chair

Augusto Sampaio                      Federal University of Pernambuco, Brazil

## Workshop and Tutorial Chairs

Emil Sekerinski                      McMaster University, Canada  
Nelma Moreira                      University of Porto, Portugal

## FM Program Committee

Bernhard Aichernig	TU Graz, Austria
Elvira Albert	Complutense University of Madrid, Spain
María Alpuente	Polytechnic University of Valencia, Spain
Dalal Alrajeh	Imperial College, UK
Mário S. Alvim	Federal University of Minas Gerais, Brazil
June Andronick	CSIRO-Data61, Australia
Christel Baier	TU Dresden, Germany
Luís Barbosa	University of Minho and UN University, Portugal
Gilles Barthe	IMDEA Software Institute, Spain
Marcello Bersani	Polytechnic University of Milan, Italy
Gustavo Betarte	Tilsor SA and University of the Republic, Uruguay

Nikolaj Bjørner	Microsoft Research, USA
Frank de Boer	CWI, The Netherlands
Sergiy Bogomolov	Australian National University, Australia
Julien Brunel	ONERA, France
Néstor Cataño	Universidad del Norte, Colombia
Ana Cavalcanti	University of York, UK
Antonio Cerone	Nazarbayev University, Kazakhstan
Marsha Chechik	University of Toronto, Canada
David Chemouil	ONERA, France
Alessandro Cimatti	FBK-IRST, Italy
Alcino Cunha	University of Minho and INESC TEC, Portugal
Michael Dierkes	Rockwell Collins, France
Alessandro Fantechi	University of Florence, Italy
Carla Ferreira	New University of Lisbon, Portugal
João Ferreira	Teesside University, UK
José L. Fiadeiro	Royal Holloway University of London, UK
Marcelo Frias	Buenos Aires Institute of Technology, Argentina
Fatemeh Ghassemi	University of Tehran, Iran
Silvia Ghilezan	University of Novi Sad, Serbia
Stefania Gnesi	ISTI-CNR, Italy
Reiner Hähnle	TU Darmstadt, Germany
Osman Hasan	University of Sciences and Technology, Pakistan
Klaus Havelund	NASA Jet Propulsion Laboratory, USA
Anne Haxthausen	TU Denmark, Denmark
Ian Hayes	University of Queensland, Australia
Constance Heitmeyer	Naval Research Laboratory, USA
Jane Hillston	University of Edinburgh, UK
Thai Son Hoang	University of Southampton, UK
Zhenjiang Hu	National Institute of Informatics, Japan
Dang Van Hung	Vietnam National University, Vietnam
Atsushi Igarashi	Kyoto University, Japan
Suman Jana	Columbia University, USA
Ali Jaoua	Qatar University, Qatar
Einar Broch Johnsen	University of Oslo, Norway
Joost-Pieter Katoen	RWTH Aachen University, Germany
Laura Kovács	TU Vienna, Austria
Axel Legay	UCLouvain, Belgium
Gabriele Lenzini	University of Luxembourg, Luxembourg
Yang Liu	Nanyang Technical University, Singapore
Alberto Lluch Lafuente	TU Denmark, Denmark
Malte Lochau	TU Darmstadt, Germany
Michele Loreti	University of Camerino, Italy
Anastasia Mavridou	NASA Ames, USA
Hernán Melgratti	University of Buenos Aires, Argentina
Sun Meng	Peking University, China
Dominique Méry	LORIA and University of Lorraine, France



Rosemary Monahan	Maynooth University, Ireland
Olfa Mosbahi	University of Carthage, Tunisia
Mohammad Mousavi	University of Leicester, UK
César Muñoz	NASA Langley, USA
Tim Nelson	Brown University, USA
Gethin Norman	University of Glasgow, UK
Colin O'Halloran	D-RisQ Software Systems, UK
Federico Olmedo	University of Chile, Chile
Gordon Pace	University of Malta, Malta
Jan Peleska	University of Bremen, Germany
Marielle Petit-Doche	Systerel, France
Alexandre Petrenko	Computer Research Institute of Montréal, Canada
Anna Philippou	University of Cyprus, Cyprus
Jorge Sousa Pinto	University of Minho and INESC TEC, Portugal
André Platzer	Carnegie Mellon University, USA
Jaco van de Pol	Aarhus University, Denmark
Tahiry Rabehaja	Macquarie University, Australia
Steve Reeves	University of Waikato, New Zealand
Matteo Rossi	Polytechnic University of Milan, Italy
Augusto Sampaio	Federal University of Pernambuco, Brazil
Gerardo Schneider	Chalmers University of Gothenburg, Sweden
Daniel Schwartz Narbonne	Amazon Web Services, USA
Natasha Sharygina	University of Lugano, Switzerland
Nikolay Shilov	Innopolis University, Russia
Ana Sokolova	University of Salzburg, Austria
Marielle Stoelinga	University of Twente, The Netherlands
Jun Sun	University of Technology and Design, Singapore
Helen Treharne	University of Surrey, UK
Elena Troubitsyna	Åbo Akademi University, Finland
Tarmo Uustalu	Reykjavik University, Iceland
Andrea Vandin	TU Denmark, Denmark
R. Venkatesh	TCS Research, India
Erik de Vink	TU Eindhoven and CWI, The Netherlands
Willem Visser	Stellenbosch University, South Africa
Farn Wang	National Taiwan University, Taiwan
Bruce Watson	Stellenbosch University, South Africa
Tim Willemse	TU Eindhoven, The Netherlands
Kirsten Winter	University of Queensland, Australia
Jim Woodcock	University of York, UK
Lijun Zhang	Chinese Academy of Sciences, China

## Additional Reviewers

Rui Abreu	Gloria Gori	Paolo Masci
Arthur Américo	Friedrich Gretz	Mieke Massink
Hugo Araujo	Jerry den Hartog	Franco Mazzanti
Myla Archer	Raju Halder	Larissa Meinicke
Sepideh Asadi	Hossein Hojjat	Alexandra Mendes
Florent Avellaneda	Karel Horak	Stephan Merz
Eduard Baranov	Zhe Hou	Ravindra Metta
Davide Basile	Thomas Hujsa	Andrea Micheli
Cláudio Belo Lourenço	Andreas Humenberger	Stefan Mitsch
Philipp Berger	Antti Hyvarinen	Alvaro Miyazawa
František Blahoudek	Peter Häfner	Carroll Morgan
Martin Blichá	Fabian Immler	Mariano Moscato
Jean-Paul Bodeveix	Miguel Isabel	Toby Murray
Brandon Bohrer	Shaista Jabeen	David Müller
Ioana Boureanu	Phillip James	Koji Nakazawa
Laura Bozzelli	Seema Jehan	Pham Ngoc Hung
Daniel Britten	Saul Johnson	Omer Nguena-Timo
James Brotherston	Violet Ka I Pun	Hans de Nivelles
Richard Bubel	Eduard Kamburjan	Quentin Peyras
Doina Bucur	Minh-Thang Khuu	Paul Piho
Juan Diego Campo	Sascha Klüppelholz	Danny Bøgsted Poulsen
Laura Carnevali	Dimitrios Kouzapas	James Power
Gustavo Carvalho	Robbert Krebbers	Tim Quatmann
Davide Cavezza	Shrawan Kumar	Jean-Baptiste Raclet
Xiaohong Chen	Luca Laurenti	Markus Roggenbach
Yu-Ting Chen	Maurice Laveaux	Guillermo Román-Díez
Robert Colvin	Corey Lewis	Jurriaan Rot
Jesús Correas Fernández	Jianlin Li	Albert Rubio
Silvano Dal Zilio	Yi Li	Enno Ruijters
Carlos Diego Damasceno	Yong Li	Sebastian Ruland
Quoc Huy Do	Ai Liu	David Sanan
Sebastian Ehmes	Wanwei Liu	Julia Sapiña
Santiago Escobar	Martin Lukac	Andy Schürr
Marco Faella	Carlos Luna	Ramy Shahin
Paul Fiterau Brostean	Lars Luthmann	Neeraj Singh
Simon Foster	Joshua Moerman	Andrew Sogokon
Maria João Frade	Hendrik Maarand	B. Srivathsan
Maciej Gazda	Kumar Madhukar	Dominic Steinhöfel
Lorenzo Gheri	Shahar Maoz	Ivan Stojic
Eduardo Giménez	Matteo Marescotti	Sandro Stucki
Pablo Gordillo	Bojan Marinkovic	Martin Tappler

Laura Titolo  
 Andrea Turrini  
 Ben Tyler  
 Evangelia Vanezi  
 Alicia Villanueva

Inna Vistbakka  
 Matthias Volk  
 Jingyi Wang  
 Shuling Wang  
 Markus Weckesser

Stephan Wesemeyer  
 Pengfei Yang  
 Haodong Yao

## **I-Day Program Committee**

M. Antony Aiello  
 Flemming Andersen  
 Stylianos Basagianni  
 Roderick Chapman  
 David Cok  
 Alessandro Fantechi  
 Chris Hawblitzel  
 Peter Gorm Larsen  
 Michael Leuschel  
 Yannick Moy  
 Jan Peleska  
 Etienne Prun  
 Kenji Taguchi  
 Stefano Tonetta  
 Daniel Zimmerman

AdaCore, USA  
 Galois Inc., USA  
 United Technologies Research Centre, Ireland  
 Protean Code Limited, UK  
 GrammaTech, USA  
 University of Florence, Italy  
 Microsoft, USA  
 Aarhus University, Denmark  
 University of Düsseldorf, Germany  
 AdaCore, France  
 Verified Systems International GmbH, Germany  
 ClearSy, France  
 CAV Technologies Co., Ltd., Japan  
 FBK-IRST, Italy  
 Galois Inc., USA

## **DS Program Committee**

Ana Cavalcanti  
 André Platzer  
 Alessandro Fantechi  
 Carlo A. Furia  
 Dalal Alrajeh  
 Einar Broch Johnson  
 Elvira Albert  
 Jaco van de Pol  
 Matteo Rossi  
 Stefania Gnesi  
 Stephan Merz

University of York, UK  
 Carnegie Mellon University, USA  
 University of Florence, Italy  
 USI, Switzerland  
 Imperial College, UK  
 University of Oslo, Norway  
 Complutense University of Madrid, Spain  
 Aarhus University, Denmark  
 Polytechnic University of Milan, Italy  
 ISTI-CNR, Italy  
 Inria, France

## **JFT Program Committee**

Cliff Jones  
 Manfred Broy

University of Newcastle, UK  
 TU Munich, Germany

## Organizing Committee

Luís Soares Barbosa	University of Minho and INESC TEC, Portugal
José Creissac Campos	University of Minho and INESC TEC, Portugal
João Pascoal Faria	University of Porto and INESC TEC, Portugal
Sara Fernandes	University of Minho and INESC TEC, Portugal
Luís Neves	Critical Software, Portugal
Ana Paiva	University of Porto and INESC TEC, Portugal

## Local Organizers

Catarina Fernandes	University of Minho and INESC TEC, Portugal
Paula Rodrigues	INESC TEC, Portugal
Ana Rita Costa	INESC TEC, Portugal

## Web Team

Francisco Neves	University of Minho and INESC TEC, Portugal
Rogério Pontes	University of Minho and INESC TEC, Portugal
Paula Rodrigues	INESC TEC, Portugal

## FME Board

Ana Cavalcanti	University of York, UK
Lars-Henrik Eriksson	Uppsala University, Sweden
Stefania Gnesi	ISTI-CNR, Italy
Einar Broch Johnsen	University of Oslo, Norway
Nico Plat	Thanos, The Netherlands

# Formal Methods for Security Functionality and for Secure Functionality (Invited Presentation)

Erik Poll

Digital Security group, Radboud University Nijmegen, The Netherlands  
erikpoll@cs.ru.nl

With cyber security becoming a growing concern, it has naturally attracted the attention of researchers in formal methods. One recent success story here is TLS: the development of the new TLS 1.3 specification has gone hand-in-hand with efforts to verify security properties of formal models [5] and the development of a fully verified implementation [3]. Earlier well-known success stories in using formal methods for security are the verifications of operating system kernels or hypervisors, namely seL4 [7] and Microsoft’s Hyper-V [10].

These examples – security protocols and OS kernels – are applications whose primary purpose is to provide security. It is natural to apply formal methods to such systems: they are by their very nature security-critical and they provide some security functionality that we can try to specify and verify.

However, we want *all* our systems to be secure, not just these security systems. There is an important difference between *secure* functionality and *security* functionality, or – given that most functionality and most security problems are down to software – between *software security* and *security software* [11]. Many, if not most, security problems arise in systems that have no specific security objective, say PDF viewers or video players, but which can still be hacked to provide attackers with unwanted functionality they can abuse.

Using formal methods to prove security is probably not on the cards of something as complex as a PDF viewer or video player. Just defining what it would mean for such a system to be secure is probably already infeasible. Still, formal methods can be useful, to prove the absence of certain types of security flaws or simply find security flaws. Successes here have been in the use of static analysis in source code analysers, e.g. tools like Fortify SCA that look for flaws in web applications and tools like Coverity that look for memory vulnerabilities in C(++) code. Another successful application of formal methods is the use of symbolic (or concolic) execution to generate test cases for security testing, as in SAGE [6] or, going one step further, not just automatically finding flaws but also automatically generating exploits, as in angr [16].

Downside of these approaches is that they are post-hoc and can only look for flaws in existing code. The *LangSec* paradigm [4, 9], on the other hand, provides ideas on how to prevent many security problems *by construction*. Key insights are that most security flaws occur in input handling and that there are several root causes in play here. Firstly, the input languages involved (e.g. file formats and network protocols) are complex, very expressive, and poorly, informally, specified. Secondly, there are *many*

of these input languages, sometimes nested or stacked. Finally, parsers for these languages are typically hand-written, with parsing code scattered throughout the application code in so-called shotgun parsers [12]. With clearer, formal specifications of input languages and generated parser code much security misery could be avoided. (Recent initiatives in tools for parser generation here include Hammer [1] and Nail [2].) Given that formal languages and parser generation are some of the most basic and established formal methods around, it is a bit of an embarrassment to us as formal methods community that sloppy language specifications and hand-coded parsers should cause so many security problems.

Some security flaws in input handling are not so much caused by *buggy* parsing of inputs, but rather by the *unexpected* parsing of input [13]. Classic examples of this are command injection, SQL injection, and Cross-Site Scripting (XSS). Tell-tale sign that unwanted parsing of input may be happening in unexpected places is the heavy use of strings as data types [14].

Information or data flow analysis can be used to detect such flaws; indeed, this is a standard technique used in the source code analysis tools mentioned above. These flaws can also be prevented by construction, namely by using type systems. A recent example of this is the ‘Trusted Types’ browser API [8] by Google, where different types are used to track different kinds of data and different trust level of data to prevent XSS vulnerabilities, esp. the DOM-based XSS vulnerabilities that have proved so difficult to root out.

To conclude, formal methods cannot only be used to *prove* security of security-critical applications and components – i.e. the security software –, but they can be much more widely used to *improve* security by ruling out of the root causes behind security flaws in input handling, and do so by construction, and hence improve software security in general. Moreover, some very basic and lightweight formal methods can be used for this: methods that we teach – or should be teaching – our students in the first years of their Bachelor degree, such as regular expressions, finite state machines, grammars, and types. Indeed, in my own research I have been surprised to see how useful the simple notion of finite state machine for describing input sequences is to discover security flaws [15].

That we have not been able to get these basic techniques into common use does not say much for our success in transferring formal methods to software engineering practice. Still, looking at the bright side, it does suggest opportunities for improvement.

## References

1. Anantharaman, P., Millian, M.C., Bratus, S., Patterson, M.L.: Input handling done right: building hardened parsers using language-theoretic security. In: Cybersecurity Development (SecDev), pp. 4–5. IEEE (2017)
2. Bangert, J., Zeldovich, N.: Nail: A practical tool for parsing and generating data formats. In: OSDI 2014, pp. 615–628. Usenix (2014)

3. Bhargavan, K., Blanchet, B., Kobeissi, N.: Verified models and reference implementations for the TLS 1.3 standard candidate. In: Security and Privacy (S&P 2017), pp. 483–502. IEEE (2017)
4. Bratus, S., Locasto, M.E., Patterson, M.L., Sassaman, L., Shubina, A.: Exploit programming: from buffer overflows to weird machines and theory of computation. *Login*, 13–21 (2011)
5. Cremers, C., Horvat, M., Hoyland, J., Scott, S., van der Merwe, T.: A comprehensive symbolic analysis of TLS 1.3. In: SIGSAC Conference on Computer and Communications Security (CCS 2017), pp. 1773–1788. ACM (2017)
6. Godefroid, P., Levin, M.Y., Molnar, D.: SAGE: Whitebox fuzzing for security testing. *Commun. ACM* **55**(3), 40–44 (2012)
7. Klein, G., et al.: seL4: Formal verification of an OS kernel. In: ACM SIGOPS, pp. 207–220. ACM (2009)
8. Kotowicz, K.: Trusted types help prevent cross-site scripting (2019). <https://developers.google.com/web/updates/2019/02/trusted-types>. blog
9. LangSec: Recognition, validation, and compositional correctness for real world security (2013). <http://langsec.org/bof-handout.pdf>. uSENIX Security BoF hand-out
10. Leinenbach, D., Santen, T.: Verifying the microsoft hyper-V hypervisor with VCC. In: Cavalcanti, A., Dams, D.R. (eds.) FM 2009, LNCS, vol. 5850, pp. 806–809. Springer, Heidelberg (2009). [https://doi.org/10.1007/978-3-642-05089-3\\_51](https://doi.org/10.1007/978-3-642-05089-3_51)
11. McGraw, G.: Software security. *IEEE Secur. Priv.* **2**(2), 80–83 (2004)
12. Momot, F., Bratus, S., Hallberg, S.M., Patterson, M.L.: The seven turrets of Babel: a taxonomy of LangSec errors and how to expunge them. In: Cybersecurity Development (SecDev 2016), pp. 45–52. IEEE (2016)
13. Poll, E.: LangSec revisited: input security flaws of the second kind. In: Workshop on Language-Theoretic Security (LangSec 2018). IEEE (2018)
14. Poll, E.: Strings considered harmful. *Login*, **43**(4), 21–26 (2018)
15. Poll, E., de Ruiter, J., Schubert, A.: Protocol state machines and session languages: specification, implementation, and security flaws. In: Workshop on Language-Theoretic Security (LangSec 2015), pp. 125–133. IEEE (2015)
16. Shoshitaishvili, Y., et al.: SoK:(state of) the art of war: offensive techniques in binary analysis. In: Symposium on Security and Privacy (SP 2016), pp. 138–157. IEEE (2016)

# Contents

## Invited Presentations

The Human in Formal Methods . . . . .	3
<i>Shriram Krishnamurthi and Tim Nelson</i>	
Successes in Deployed Verified Software (and Insights on Key Social Factors) . . . . .	11
<i>June Andronick</i>	

## Verification

Provably Correct Floating-Point Implementation of a Point-in-Polygon Algorithm . . . . .	21
<i>Mariano M. Moscato, Laura Titolo, Marco A. Feliú, and César A. Muñoz</i>	
Formally Verified Roundoff Errors Using SMT-based Certificates and Subdivisions . . . . .	38
<i>Joachim Bard, Heiko Becker, and Eva Darulova</i>	
Mechanically Verifying the Fundamental Liveness Property of the Chord Protocol . . . . .	45
<i>Jean-Paul Bodeveix, Julien Brunel, David Chemouil, and Mamoun Filali</i>	
On the Nature of Symbolic Execution . . . . .	64
<i>Frank S. de Boer and Marcello Bonsangue</i>	

## Synthesis Techniques

GR(1)*: GR(1) Specifications Extended with Existential Guarantees . . . . .	83
<i>Gal Amram, Shahar Maoz, and Or Pistiner</i>	
Counterexample-Driven Synthesis for Probabilistic Program Sketches . . . . .	101
<i>Milan Češka, Christian Hensel, Sebastian Junges, and Joost-Pieter Katoen</i>	
Synthesis of Railway Signaling Layout from Local Capacity Specifications . . . .	121
<i>Bjørnar Luteberget, Christian Johansen, and Martin Steffen</i>	
Pegasus: A Framework for Sound Continuous Invariant Generation. . . . .	138
<i>Andrew Sogokon, Stefan Mitsch, Yong Kiam Tan, Katherine Cordwell, and André Platzer</i>	



## Concurrency

A Parametric Rely-Guarantee Reasoning Framework for Concurrent Reactive Systems . . . . .	161
<i>Yongwang Zhao, David Sanán, Fuyuan Zhang, and Yang Liu</i>	
Verifying Correctness of Persistent Concurrent Data Structures . . . . .	179
<i>John Derrick, Simon Doherty, Brijesh Dongol, Gerhard Schellhorn, and Heike Wehrheim</i>	
Compositional Verification of Concurrent Systems by Combining Bisimulations. . . . .	196
<i>Frédéric Lang, Radu Mateescu, and Franco Mazzanti</i>	

## Model Checking Circus

Towards a Model-Checker for <i>Circus</i> . . . . .	217
<i>Artur Oliveira Gomes and Andrew Butterfield</i>	
<i>Circus2CSP</i> : A Tool for Model-Checking <i>Circus</i> Using FDR. . . . .	235
<i>Artur Oliveira Gomes and Andrew Butterfield</i>	

## Model Checking

How Hard Is Finding Shortest Counter-Example Lassos in Model Checking? . . . . .	245
<i>Rüdiger Ehlers</i>	
From LTL to Unambiguous Büchi Automata via Disambiguation of Alternating Automata. . . . .	262
<i>Simon Jantsch, David Müller, Christel Baier, and Joachim Klein</i>	
Generic Partition Refinement and Weighted Tree Automata . . . . .	280
<i>Hans-Peter Deifel, Stefan Milius, Lutz Schröder, and Thorsten Wißmann</i>	
Equilibria-Based Probabilistic Model Checking for Concurrent Stochastic Games . . . . .	298
<i>Marta Kwiatkowska, Gethin Norman, David Parker, and Gabriel Santos</i>	

## Analysis Techniques

Abstract Execution . . . . .	319
<i>Dominic Steinhöfel and Reiner Hähnle</i>	
Static Analysis for Detecting High-Level Races in RTOS Kernels. . . . .	337
<i>Abhishek Singh, Rekha Pai, Deepak D'Souza, and Meenakshi D'Souza</i>	

Parallel Composition and Modular Verification of Computer Controlled Systems in Differential Dynamic Logic . . . . .	354
<i>Simon Lunel, Stefan Mitsch, Benoit Boyer, and Jean-Pierre Talpin</i>	
An Axiomatic Approach to Liveness for Differential Equations. . . . .	371
<i>Yong Kiam Tan and André Platzer</i>	
Local Consistency Check in Synchronous Dataflow Models . . . . .	389
<i>Dina Irofti and Paul Dubrulle</i>	
Gray-Box Monitoring of Hyperproperties . . . . .	406
<i>Sandro Stucki, César Sánchez, Gerardo Schneider, and Borzoo Bonakdarpour</i>	
Quantitative Verification of Numerical Stability for Kalman Filters . . . . .	425
<i>Alexandros Evangelidis and David Parker</i>	
Concolic Testing Heap-Manipulating Programs . . . . .	442
<i>Long H. Pham, Quang Loc Le, Quoc-Sang Phan, and Jun Sun</i>	
<b>Specification Languages</b>	
Formal Semantics Extraction from Natural Language Specifications for ARM . . . . .	465
<i>Anh V. Vu and Mizuhito Ogawa</i>	
GOSPEL—Providing OCaml with a Formal Specification Language . . . . .	484
<i>Arthur Charguéraud, Jean-Christophe Filliâtre, Cláudio Lourenço, and Mário Pereira</i>	
Unification in Matching Logic . . . . .	502
<i>Andrei Arusoaie and Dorel Lucanu</i>	
Embedding High-Level Formal Specifications into Applications . . . . .	519
<i>Philipp Körner, Jens Bendisposto, Jannik Dunkelau, Sebastian Krings, and Michael Leuschel</i>	
<b>Reasoning Techniques</b>	
Value-Dependent Information-Flow Security on Weak Memory Models. . . . .	539
<i>Graeme Smith, Nicholas Coughlin, and Toby Murray</i>	
Reasoning Formally About Database Queries and Updates . . . . .	556
<i>Jon Haël Brenas, Rachid Echahed, and Martin Strecker</i>	
Abstraction and Subsumption in Modular Verification of C Programs . . . . .	573
<i>Lennart Beringer and Andrew W. Appel</i>	

## Modelling Languages

IELE: A Rigorously Designed Language and Tool Ecosystem for the Blockchain. . . . .	593
<i>Theodoros Kasampalis, Dwight Guth, Brandon Moore, Traian Florin Şerbănuţă, Yi Zhang, Daniele Filaretti, Virgil Şerbănuţă, Ralph Johnson, and Grigore Roşu</i>	
APML: An Architecture Proof Modeling Language . . . . .	611
<i>Diego Marmosoler and Genc Blakqori</i>	

## Learning-Based Techniques and Applications

Learning Deterministic Variable Automata over Infinite Alphabets . . . . .	633
<i>Sarai Sheinvald</i>	
$L^*$ -Based Learning of Markov Decision Processes . . . . .	651
<i>Martin Tappler, Bernhard K. Aichernig, Giovanni Bacci, Maria Eichlseder, and Kim G. Larsen</i>	
Star-Based Reachability Analysis of Deep Neural Networks . . . . .	670
<i>Hoang-Dung Tran, Diago Manzananas Lopez, Patrick Musau, Xiaodong Yang, Luan Viet Nguyen, Weiming Xiang, and Taylor T. Johnson</i>	

## Refactoring and Reprogramming

SOA and the Button Problem . . . . .	689
<i>Sung-Shik Jongmans, Arjan Lamers, and Marko van Eekelen</i>	
Controlling Large Boolean Networks with Temporary and Permanent Perturbations. . . . .	707
<i>Cui Su, Soumya Paul, and Jun Pang</i>	

## I-Day Presentations

Formal Methods Applicability on Space Applications Specification and Implementation Using MORA-TSP . . . . .	727
<i>Daniel Silveira, Andreas Jung, Marcel Verhoef, and Tiago Jorge</i>	
Industrial Application of Event-B to a Wayside Train Monitoring System: Formal Conceptual Data Analysis . . . . .	738
<i>Robert Eschbach</i>	
Property-Driven Software Analysis (Extended Abstract). . . . .	746
<i>Mathieu Comptier, David Déharbe, Paulin Fournier, and Julien Molinero-Perez</i>	

Practical Application of SPARK to OpenUxAS. . . . .	751
<i>M. Anthony Aiello, Claire Dross, Patrick Rogers, Laura Humphrey, and James Hamil</i>	
Adopting Formal Methods in an Industrial Setting: The Railways Case . . . . .	762
<i>Maurice H. ter Beek, Arne Borälv, Alessandro Fantechi, Alessio Ferrari, Stefania Gnesi, Christer Löfving, and Franco Mazzanti</i>	
Author Index . . . . .	773