Communications in Computer and Information Science 694

Commenced Publication in 2007
Founding and Former Series Editors:
Alfredo Cuzzocrea, Dominik Ślęzak, and Xiaokang Yang

More information about this series at http://www.springer.com/series/7899

Cyrille Artho · Peter Csaba Ölveczky (Eds.)

Formal Techniques for Safety-Critical Systems

5th International Workshop, FTSCS 2016
Tokyo, Japan, November 14, 2016
Revised Selected Papers

Springer

Editors
Cyrille Artho
KTH Royal Institute of Technology
Stockholm
Sweden

Peter Csaba Ölveczky
Department of Informatics
University of Oslo
Oslo
Norway

ISSN 1865-0929 ISSN 1865-0937 (electronic)
Communications in Computer and Information Science
ISBN 978-3-319-53945-4 ISBN 978-3-319-53946-1 (eBook)
DOI 10.1007/978-3-319-53946-1

Library of Congress Control Number: 2017931545

Printed on acid-free paper

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

Preface

This volume contains the proceedings of the Fifth International Workshop on Formal Techniques for Safety-Critical Systems (FTSCS 2016), held in Tokyo on November 14, 2016, as a satellite event of the ICFEM conference.

The aim of this workshop is to bring together researchers and engineers who are interested in the application of formal and semi-formal methods to improve the quality of safety-critical computer systems. FTSCS strives to promote research and development of formal methods and tools for industrial applications, and is particularly interested in industrial applications of formal methods. Specific topics include, but are not limited to:

- case studies and experience reports on the use of formal methods for analyzing safety-critical systems, including avionics, automotive, railway, medical, and other kinds of safety-critical and QoS-critical systems;
- methods, techniques, and tools to support automated analysis, certification, debugging, etc., of complex safety/QoS-critical systems;
- analysis methods that address the limitations of formal methods in industry (usability, scalability, etc.);
- formal analysis support for modeling languages used in industry, such as AADL, Ptolemy, SysML, SCADE, Modelica, etc.; and
- code generation from validated models.

The workshop received 23 regular paper submissions. Each submission was reviewed by at least three referees. Based on the reviews and extensive discussions, the program committee selected nine papers for presentation at the workshop and inclusion in this volume. Another highlight of the workshop was an invited talk by Naoki Kobayashi.

Many colleagues and friends have contributed to FTSCS 2016. We thank Naoki Kobayashi for giving an excellent invited talk and the authors who submitted their work to FTSCS 2016 and who, through their contributions, made the workshop an interesting event. We are particularly grateful that so many well-known researchers agreed to serve on the program committee, and that they provided timely, insightful, and detailed reviews. We also thank the editors of *Communications in Computer and Information Science* for agreeing to publish the proceedings of FTSCS 2016 as a volume in their series, and Shaoying Liu and Shin Nakajima for their help with the local arrangements.

December 2016

Cyrille Artho
Peter Csaba Ölveczky

Organization

Program Chairs

Cyrille Artho	KTH Royal Institute of Technology, Sweden
Peter Csaba Ölveczky	University of Oslo, Norway

Program Committee

Étienne André	University Paris 13, France
Toshiaki Aoki	JAIST, Japan
Cyrille Artho	KTH Royal Institute of Technology, Sweden
Kyungmin Bae	Pohang University of Science and Technology, Korea
Eun-Hye Choi	AIST, Japan
Alessandro Fantechi	University of Florence and ISTI-CNR, Pisa, Italy
Bernd Fischer	Stellenbosch University, South Africa
Osman Hasan	National University of Sciences & Technology, Pakistan
Klaus Havelund	NASA JPL, USA
Jérôme Hugues	Institute for Space and Aeronautics Engineering, France
Marieke Huisman	University of Twente, The Netherlands
Ralf Huuck	Synopsys, Australia
Fuyuki Ishikawa	National Institute of Informatics, Japan
Takashi Kitamura	AIST, Japan
Alexander Knapp	Augsburg University, Germany
Thierry Lecomte	ClearSy System Engineering, France
Yang Liu	Nanyang Technological University, Singapore
Robi Malik	University of Waikato, New Zealand
Frédéric Mallet	Université Nice Sophia Antipolis, France
Roberto Nardone	University of Naples Federico II, Italy
Vivek Nigam	Federal University of Paraíba, Brazil
Thomas Noll	RWTH Aachen University, Germany
Kazuhiro Ogata	JAIST, Japan
Peter Csaba Ölveczky	University of Oslo, Norway
Charles Pecheur	Université catholique de Louvain, Belgium
Markus Roggenbach	Swansea University, UK
Ralf Sasse	ETH Zürich, Switzerland
Martina Seidl	Johannes Kepler University Linz, Austria
Oleg Sokolsky	University of Pennsylvania, USA
Sofiène Tahar	Concordia University, Canada
Carolyn Talcott	SRI International, USA
Tatsuhiro Tsuchiya	Osaka University, Japan

András Vörös	Budapest University of Technology and Economics, Hungary
Chen-Wei Wang	State University of New York (SUNY), Korea
Mike Whalen	University of Minnesota, USA
Huibiao Zhu	East China Normal University, China

Additional Reviewers

Beillahi, Sidi Mohamed
Bukhari, Syed Ali Asadullah
Du, Xiaoning
Fang, Huixing
Gentile, Ugo
Gillard, Xavier
Oortwijn, Wytse
Qasim, Muhammad
Sardar, Muhammad Usama
Van Zijl, Lynette

On Two Higher-Order Extensions of Model Checking (Invited Talk)

Naoki Kobayashi

The University of Tokyo, Bunkyō, Japan
koba@is.s.u-tokyo.ac.jp

Inspired by the success of finite state model checking [2] in system verification, two kinds of its higher-order extensions have been studied since around 2000. One is model checking of higher-order recursion schemes (HORS) [3, 13], where the language for describing systems to be verified is extended to higher-order, and the other is higher-order modal fixpoint logic (HFL) model checking of finite-state systems [18], where the logic for specifying properties to be verified is extended to higher-order. Table 1 summarizes those extensions. In general, HORS model checking can be used for precisely modeling and verifying a certain class of *infinite* state systems, and HFL model checking can be used for checking *non-regular* properties of systems. HORS model checking has been successfully applied to automated verification of higher-order programs [5, 6, 8, 9, 10, 12, 14, 16, 17, 19], whereas HFL model checking has been studied for verification of concurrent systems [11, 18]. Although both HORS and HFL model checking problems are k-EXPTIME complete for the order-k fragments (where the order is the largest type-theoretic order of functions used in HORS and HFL respectively), practical model checking algorithms have been developed, which do not always suffer from the k-EXPTIME bottleneck [1, 4, 15]. We provide a brief introduction to the HORS and HFL model checking problems, their applications, and the state-of-the-art of higher-order model checkers and tools built on top of them. We also touch upon our recent result on the relationship between HORS and HFL model checking [7].

Table 1. Finite state model checking and its higher-order extensions

	Models	Logic
Finite state model checking	Finite state systems	Modal μ-calculus (or, LTL/CTL/CTL*)
HORS model checking	Higher-order recursion schemes (HORS)	Modal μ-calculus (or, tree automata)
HFL model checking	Finite state systems	Higher-order modal fixpoint logic (HFL)

References

1. Broadbent, C.H., Kobayashi, N.: Saturation-based model checking of higher-order recursion schemes. In: Proceedings of CSL 2013. LIPIcs, vol. 23, pp. 129–148 (2013)
2. Clarke, E.M., Grumberg, O., Peled, D.A.: Model Checking. The MIT Press (1999)
3. Knapik, T., Niwinski, D., Urzyczyn, P.: Higher-order pushdown trees are easy. In: Nielsen, M., Engberg, U. (eds.) FoSSaCS 2002. LNCS, vol. 2303, pp. 205–222. Springer, Heidelberg (2002)
4. Kobayashi, N.: Model-checking higher-order functions. In: Proceedings of PPDP 2009, pp. 25–36. ACM Press (2009)
5. Kobayashi, N.: Types and higher-order recursion schemes for verification of higher-order programs. In: Proceedings of POPL, pp. 416–428. ACM Press (2009)
6. Kobayashi, N.: Model checking higher-order programs. J. ACM **60**(3) (2013)
7. Kobayashi, N., Étienne Lozes, Bruse, F.: On the relationship between higher-order recursion schemes and higher-order modal fixpoint logic. In: Proceedings of POPL 2017 (2017, to appear)
8. Kobayashi, N., Sato, R., Unno, H.: Predicate abstraction and CEGAR for higher-order model checking. In: Proceedings of PLDI, pp. 222–233. ACM Press (2011)
9. Kobayashi, N., Tabuchi, N., Unno, H.: Higher-order multi-parameter tree transducers and recursion schemes for program verification. In: Proceedings of POPL, pp. 495–508. ACM Press (2010)
10. Kuwahara, T., Sato, R., Unno, H., Kobayashi, N.: Predicate abstraction and CEGAR for disproving termination of higher-order functional programs. In: Kroening, D., C.S. Păsăreanu (eds.) Proceedings of CAV 2015. LNCS, vol. 9207, pp. 287–303. Springer, Switzerland (2015)
11. Lange, M., Lozes, É., Guzmán, M.V.: Model-checking process equivalences. Theor. Comput. Sci. **560**, 326–347 (2014)
12. Murase, A., Terauchi, T., Kobayashi, N., Sato, R., Unno, H.: Temporal verification of higher-order functional programs. In: Proceedings of POPL 2016 (2016, to appear)
13. Ong, C.H.L.: On model-checking trees generated by higher-order recursion schemes. In: LICS 2006, pp. 81–90. IEEE Computer Society Press (2006)
14. Ong, C.H.L., Ramsay, S.: Verifying higher-order programs with pattern-matching algebraic data types. In: Proceedings of POPL, pp. 587–598. ACM Press (2011)
15. Ramsay, S., Neatherway, R., Ong, C.H.L.: An abstraction refinement approach to higher-order model checking. In: Proceedings of POPL 2014, pp. 61–72. ACM (2014)
16. Sato, R., Unno, H., Kobayashi, N.: Towards a scalable software model checker for higher-order programs. In: Proceedings of PEPM 2013, pp. 53–62. ACM Press (2013)
17. Unno, H., Terauchi, T., Kobayashi, N.: Automating relatively complete verification of higher-order functional programs. In: The 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2013. pp. 75–86. ACM (2013)
18. Viswanathan, M., Viswanathan, R.: A higher order modal fixed point logic. In: Gardner, P., Yoshida, N. (eds.) CONCUR. LNCS, vol. 3170, pp. 512–528. Springer, Heidelberg (2004)
19. Watanabe, K., Sato, R., Tsukada, T., Kobayashi, N.: Automatically disproving fair termination of higher-order functional programs. In: Proceedings of ICFP 2016, pp. 243–255. ACM (2016)

Contents