Eyal Bin
Avi Ziv
Shmuel Ur (Eds.)

LNCS 4383

Hardware and Software, Verification and Testing

Second International Haifa Verification Conference, HVC 2006
Haifa, Israel, October 2006
Revised Selected Papers

Springer

Eyal Bin   Avi Ziv   Shmuel Ur (Eds.)

Hardware and Software, Verification and Testing

Second International Haifa Verification Conference, HVC 2006
Haifa, Israel, October 23-26, 2006
Revised Selected Papers

Springer

Volume Editors

Eyal Bin
Avi Ziv
Shmuel Ur
IBM Labs, Haifa University
Mount Carmel, Haifa 31905, Israel
E-mail: {bin,aziv,ur}@il.ibm.com

Library of Congress Control Number: 2007920490

CR Subject Classification (1998): D.2.4-5, D.2, D.3, F.3

LNCS Sublibrary: SL 2 – Programming and Software Engineering

ISSN 0302-9743
ISBN-10 3-540-70888-X Springer Berlin Heidelberg New York
ISBN-13 978-3-540-70888-9 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12019380 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 4383

Preface

The Haifa Verification Conference 2006 took place for the second year in a row at the IBM Haifa Research Lab and at the Haifa University in Israel during October 23–26, 2006. The verification conference was a three-day, single-track conference followed by a one-day tutorial on PSL.

This Haifa Verification Conference was established to bring together researchers from two different disciplines, hardware verification and software testing. The use of similar techniques among the two communities enabled the conference to help generate a unique synergy that fortifies both groups. This year, we had two traditional tracks, hardware verification and software testing, in addition to a new track dedicated to tools in these areas.

The conference emphasized applicability to real-world challenges, which was vital to the many attendees coming from industry. The conference hosted two internationally recognized individuals as keynote speakers. Randal E. Bryant, Dean and University Professor from the School of Computer Science at Carnegie Mellon University gave a talk on "System Modeling and Formal Verification with UCLID" and Michael Jackson from the University of Newcastle gave a talk on "Testing the Machine in the World." The numerous invited speakers presented topics of great interest to the audience. Just some of these outstanding speakers included Cindy Eisner in the hardware verification track, Alon Gluska and Andrew Piziali in the tools track, and Mauro Pezze and Nir Shavit in the software testing track. The prize for Best Paper was awarded to Stefan Staber, Gerschwin Fey, Roderick Bloem and Rolf Drechsler from Graz University of Technology and the University of Bremen, for their paper titled "Automatic Fault Localization for Property Checking."

Thirty-three papers from ten countries were submitted, including Israel, Finland, India, the Czech Republic, Germany, China, USA, Spain, France, and Switzerland. The papers were reviewed by the Program Committee and additional referees, with an average of 3.6 reviews per paper. Of the papers submitted, 15 were accepted. The acceptance was based on the score received, the reviewer's confidence and the final decisions of the Organizing Committee. The keynote speakers and the invited speakers were encouraged to submit papers as well. This volume is composed of the papers accepted by the committee and the invited papers. This volume also includes an abstract of the conference panel on the "Unpaved Road between Hardware Verification and Software Testing Techniques" moderated by Shmuel Ur.

This year's conference included a number of new initiatives. A Web application was adopted to enable the online submission and review of papers. A ten-minute multimedia clip was produced to provide an overview of the activities in the conference. The clip covered recent news highlights in verification from around the world and gave viewers a short virtual tour of Haifa through scenes from around the city. The conference also included a tool exhibition where leading EDA companies presented their products. The conference organizers initiated a 'speed networking' session; based on the original idea of speed dating, this activity helped foster introductions and collaboration among individuals attending the event.

Attendance at the conference was very high throughout the four conference days, with more than 250 participants from several different countries. The facilities provided by the IBM Haifa Research Labs and the Caesarea Edmond Benjamin de Rothschild Foundation Institute for Interdisciplinary Applications of Computer Science (C.R.I.) were remarked upon very favorably by the attendees, as was the proficiency of the administrative assistants.

We would like to thank our sponsors, IBM and CRI, the Organizing Committee, and the Program Committee. Our appreciation goes out to the administrative assistants, especially Vered Aharon from IBM and Rona Perkis from CRI. Special thanks to Shai Halevi, Iliya Kalderon, Ido Levy, and Valentin Mashiah for their important help with the submission and review Web application. We also wish to thank the communications team for their important role: Ettie Gilead, Chani Sacharen, Yair Harry, Tamar Dekel, Hanan Singer and Anne Lustig-Picus. Many thanks to Tsvi Kuflik for his vital help with the proceedings. We would also like to extend special thanks all the authors who contributed their work.

It is our hope that the enthusiasm and value generated by this conference will lead to many other interesting events in the growing fields addressed by the hardware verification and software testing communities.

We would like also to thank Dana Fisman for giving the tutorial on PSL.

October 2006 Eyal Bin

Organization

The Haifa Verification Conference 2006 was organized by:

General Chair and Program Chair

Eyal Bin (bin@il.ibm.com)

Verification Conference Organizing Committee

Eyal Bin (bin@il.ibm.com)
Gadiel Auerbach (gadiel@il.ibm.com)
Laurent Fournier (laurent@il.ibm.com)
Moshe Levinger (levinger@il.ibm.com)
Shmuel Ur (ur@il.ibm.com)
Yaniv Eytani (ieytani@cslx.haifa.ac.il)
Yaron Wolfsthal (wolfstal@il.ibm.com)
Karen Yorav (yorav@il.ibm.com)
Avi Ziv (aziv@il.ibm.com)

Verification Track Co-chairs

Laurent Fournier, IBM Haifa Labs, Israel (laurent@il.ibm.com)
Karen Yorav, IBM Haifa Labs, Israel (yorav@il.ibm.com)

Tools Track Co-chairs

Avi Ziv, IBM Haifa Labs, Israel (aziv@il.ibm.com)
Gadiel Auerbach, IBM Haifa Labs, Israel (gadiel@il.ibm.com)

Software Testing Track Chair

Shmuel Ur, IBM Haifa Labs, Israel (ur@il.ibm.com)

PSL Tutorial Track Chair

Gadiel Auerbach, IBM Haifa Labs, Israel (gadiel@il.ibm.com)

Program Committee

Aarti Gupta, NEC Labs America (agupta@nec-labs.com)
Abraham Kandel, University of South Florida, USA (kandel@cse.usf.edu)
Alessandro Cimatti, IRST - Istituto per la Ricerca Scientifica e Tecnologica, Italy (cimatti@itc.it)
Amos Noy, Cadence (amos@cadence.com)
Andrew Piziali, Cadence (andy@cadence.com)
Assaf Schuster, Technion Institute, Haifa, Israel (assaf@cs.technion.ac.il)
Avi Ziv, IBM Haifa Labs, Israel (aziv@il.ibm.com)
Bernd Finkbeiner, Universität des Saarlandes , Germany (finkbeiner@cs.uni-sb.de)
Cindy Eisner, IBM Haifa Labs, Israel (EISNER@il.ibm.com)
Daniel Kroening, Computer Systems Institute, ETH Zuerich (kroening@handshake.de)
Dominique Borrione, Laboratoire TIMA (Dominique.Borrione@imag.fr)
Eitan Farchi, IBM Haifa Labs, Israel (farchi@il.ibm.com)
Erich Marschner, Cadence (erichm@cadence.com)
Eyal Bin, IBM Haifa Labs, Israel (bin@il.ibm.com)
Fabio Somenzi, University of Colorado (fabio@Colorado.EDU)
Gadiel Auerbach, IBM Haifa Labs, Israel (GADIEL@il.ibm.com)
Geert Janssen, IBM T.J. Watson Research Center (geert@watson.ibm.com)
Holger Hermanns, Saarland University, Germany (hermann@cs.uni-sb.de)
Ilan Harris, University of California, Irvine (harris@ics.uci.edu)
Jason Baumgartner, IBM Austin (baumgarj@us.ibm.com)
Joao Lourenco, University Nova de Lisboa (Joao.Lourenco@di.fct.unl.pt)
Jong-Deok Choi, IBM Research, USA (jdchoi@us.ibm.com)
Karen Yorav, IBM Haifa Labs, Israel (YORAV@il.ibm.com)
Ken McMillan, Cadence (mcmillan@cadence.com)
Kerstin Eder, University of Bristol (eder@cs.bris.ac.uk)
Klaus Havelund, NASA's Jet Propulsion Labratory (Klaus.Havelund@jpl.nasa.gov)
Laurent Fournier, IBM Haifa Labs, Israel (LAURENT@il.ibm.com)
Lyes Benalycherif, STMicroelectronics (Lyes.Benalycherif@st.com)
Mark Last, Ben Gurion University, Israel (mlast@bgumail.bgu.ac.il)
Mauro Pezze, Universita degli Studi di Milano, Bicocca (pezze@disco.unimib.it)
Moshe Levinger, IBM Haifa Labs, Israel (LEVINGER@il.ibm.com)
Ofer Strichman, Technion, Israel (ofers@ie.technion.ac.il)
Orit Edelstein, IBM Haifa Labs, Israel (edelstein@il.ibm.com)
Orna Kupferman, Hebrew University, Israel (orna@cs.huji.ac.il)
Pablo P. Sanchez, University of Cantabria (sanchez@teisa.unican.es)
Paul Strooper, University of Queensland, Australia (pstroop@itee.uq.edu.au)
Roderick Bloem, Graz University of Technology (Roderick.Bloem@ist.TUGratz.at)
Scott Stoller, SUNY Stony Brook, USA (stoller@cs.sunysb.edu)
Serdar Tasiran, Koç University, Turkey (stasiran@ku.edu.tr)
Sharad Malik, Princeton University (sharad@princeton.edu)
Shmuel Ur, IBM Haifa Labs, Israel (UR@il.ibm.com)

Tao Xie, North Carolina State University, USA (taoxie@acm.org)
Tsvi Kuflik, University of Haifa, Israel (tsvikak@mis.hevra.haifa.ac.il)
Warren Hunt, University of Texas, Austin (hunt@cs.utexas.edu)
Willem Visser, NASA, USA (wvisser@email.arc.nasa.gov)
Wolfgang Roesner, IBM Austin, USA (wolfgang@us.ibm.com)
Yaron Wolfsthal, IBM Haifa Labs, Israel (wolfstal@il.ibm.com)
Ziyad Hanna, Intel Israel (ziyad.hanna@intel.com)

Additional Referees

Ali Bayazit
Allon Adir
Andreas Griesmayer
Benny Godlin
Calogero Zarba
Georg Weissenbacher
Hana Chockler
Jörn Guy Süß
Klaus Draeger
Marco Roveri
Margaret Wojcicki
Mark Moulin
Nicolas Blanc
Orna Raz
Philippe Georgelin
Rachel Tzoref
Stefan Staber
Yarden Nir-Buchbinder
Yoad Lustig
Zhaohui Fu

Lecture Notes in Computer Science

For information about Vols. 1–4291

please contact your bookseller or Springer

Vol. 4405: L. Padgham, F. Zambonelli (Eds.), Agent-Oriented Software Engineering VII. XII, 225 pages. 2007.

Vol. 4394: A. Gelbukh (Ed.), Computational Linguistics and Intelligent Text Processing. XVI, 648 pages. 2007.

Vol. 4393: W. Thomas, P. Weil (Eds.), STACS 2007. XVIII, 708 pages. 2007.

Vol. 4392: S.P. Vadhan (Ed.), Theory of Cryptography. XI, 595 pages. 2007.

Vol. 4390: S.O. Kuznetsov, S. Schmidt (Eds.), Formal Concept Analysis. X, 329 pages. 2007. (Sublibrary LNAI).

Vol. 4385: K. Coninx, K. Luyten, K.A. Schneider (Eds.), Task Models and Diagrams for Users Interface Design. XI, 355 pages. 2007.

Vol. 4384: T. Washio, K. Satoh, H. Takeda, A. Inokuchi (Eds.), New Frontiers in Artificial Intelligence. IX, 401 pages. 2007. (Sublibrary LNAI).

Vol. 4383: E. Bin, A. Ziv, S. Ur (Eds.), Hardware and Software, Verification and Testing. XII, 235 pages. 2007.

Vol. 4381: J. Akiyama, W.Y.C. Chen, M. Kano, X. Li, Q. Yu (Eds.), Discrete Geometry, Combinatorics and Graph Theory. XI, 289 pages. 2007.

Vol. 4380: S. Spaccapietra, P. Atzeni, F. Fages, M.-S. Hacid, M. Kifer, J. Mylopoulos, B. Pernici, P. Shvaiko, J. Trujillo, I. Zaihrayeu (Eds.), Journal on Data Semantics VIII. XV, 219 pages. 2007.

Vol. 4378: I. Virbitskaite, A. Voronkov (Eds.), Perspectives of Systems Informatics. XIV, 496 pages. 2007.

Vol. 4377: M. Abe (Ed.), Topics in Cryptology – CT-RSA 2007. XI, 403 pages. 2006.

Vol. 4373: K. Langendoen, T. Voigt (Eds.), Wireless Sensor Networks. XIII, 358 pages. 2007.

Vol. 4372: M. Kaufmann, D. Wagner (Eds.), Graph Drawing. XIV, 454 pages. 2007.

Vol. 4371: K. Inoue, K. Satoh, F. Toni (Eds.), Computational Logic in Multi-Agent Systems. X, 315 pages. 2007. (Sublibrary LNAI).

Vol. 4369: M. Umeda, A. Wolf, O. Bartenstein, U. Geske, D. Seipel, O. Takata (Eds.), Declarative Programming for Knowledge Management. X, 229 pages. 2006. (Sublibrary LNAI).

Vol. 4368: T. Erlebach, C. Kaklamanis (Eds.), Approximation and Online Algorithms. X, 345 pages. 2007.

Vol. 4367: K. De Bosschere, D. Kaeli, P. Stenström, D. Whalley, T. Ungerer (Eds.), High Performance Embedded Architectures and Compilers. XI, 307 pages. 2007.

Vol. 4364: T. Kühne (Ed.), Models in Software Engineering. XI, 332 pages. 2007.

Vol. 4362: J. van Leeuwen, G.F. Italiano, W. van der Hoek, C. Meinel, H. Sack, F. Plášil (Eds.), SOFSEM 2007: Theory and Practice of Computer Science. XXI, 937 pages. 2007.

Vol. 4361: H.J. Hoogeboom, G. Păun, G. Rozenberg, A. Salomaa (Eds.), Membrane Computing. IX, 555 pages. 2006.

Vol. 4360: W. Dubitzky, A. Schuster, P.M.A. Sloot, M. Schroeder, M. Romberg (Eds.), Distributed, High-Performance and Grid Computing in Computational Biology. X, 192 pages. 2007. (Sublibrary LNBI).

Vol. 4358: R. Vidal, A. Heyden, Y. Ma (Eds.), Dynamical Vision. IX, 329 pages. 2007.

Vol. 4357: L. Buttyán, V. Gligor, D. Westhoff (Eds.), Security and Privacy in Ad-Hoc and Sensor Networks. X, 193 pages. 2006.

Vol. 4355: J. Julliand, O. Kouchnarenko (Eds.), B 2007: Formal Specification and Development in B. XIII, 293 pages. 2006.

Vol. 4354: M. Hanus (Ed.), Practical Aspects of Declarative Languages. X, 335 pages. 2006.

Vol. 4353: T. Schwentick, D. Suciu (Eds.), Database Theory – ICDT 2007. XI, 419 pages. 2006.

Vol. 4352: T.-J. Cham, J. Cai, C. Dorai, D. Rajan, T.-S. Chua, L.-T. Chia (Eds.), Advances in Multimedia Modeling, Part II. XVIII, 743 pages. 2006.

Vol. 4351: T.-J. Cham, J. Cai, C. Dorai, D. Rajan, T.-S. Chua, L.-T. Chia (Eds.), Advances in Multimedia Modeling, Part I. XIX, 797 pages. 2006.

Vol. 4349: B. Cook, A. Podelski (Eds.), Verification, Model Checking, and Abstract Interpretation. XI, 395 pages. 2007.

Vol. 4348: S.T. Taft, R.A. Duff, R.L. Brukardt, E. Ploedereder, P. Leroy (Eds.), Ada 2005 Reference Manual. XXII, 765 pages. 2006.

Vol. 4347: J. Lopez (Ed.), Critical Information Infrastructures Security. X, 286 pages. 2006.

Vol. 4345: N. Maglaveras, I. Chouvarda, V. Koutkias, R. Brause (Eds.), Biological and Medical Data Analysis. XIII, 496 pages. 2006. (Sublibrary LNBI).

Vol. 4344: V. Gruhn, F. Oquendo (Eds.), Software Architecture. X, 245 pages. 2006.

Vol. 4342: H. de Swart, E. Orłowska, G. Schmidt, M. Roubens (Eds.), Theory and Applications of Relational Structures as Knowledge Instruments II. X, 373 pages. 2006. (Sublibrary LNAI).

Vol. 4341: P.Q. Nguyen (Ed.), Progress in Cryptology - VIETCRYPT 2006. XI, 385 pages. 2006.

Vol. 4340: R. Prodan, T. Fahringer, Grid Computing. XXIII, 317 pages. 2007.

Vol. 4339: E. Ayguadé, G. Baumgartner, J. Ramanujam, P. Sadayappan (Eds.), Languages and Compilers for Parallel Computing. XI, 476 pages. 2006.

Vol. 4338: P. Kalra, S. Peleg (Eds.), Computer Vision, Graphics and Image Processing. XV, 965 pages. 2006.

Vol. 4337: S. Arun-Kumar, N. Garg (Eds.), FSTTCS 2006: Foundations of Software Technology and Theoretical Computer Science. XIII, 430 pages. 2006.

Vol. 4335: S.A. Brueckner, S. Hassas, M. Jelasity, D. Yamins (Eds.), Engineering Self-Organising Systems. XII, 212 pages. 2007. (Sublibrary LNAI).

Vol. 4334: B. Beckert, R. Hähnle, P.H. Schmitt (Eds.), Verification of Object-Oriented Software. XXIX, 658 pages. 2007. (Sublibrary LNAI).

Vol. 4333: U. Reimer, D. Karagiannis (Eds.), Practical Aspects of Knowledge Management. XII, 338 pages. 2006. (Sublibrary LNAI).

Vol. 4332: A. Bagchi, V. Atluri (Eds.), Information Systems Security. XV, 382 pages. 2006.

Vol. 4331: G. Min, B. Di Martino, L.T. Yang, M. Guo, G. Ruenger (Eds.), Frontiers of High Performance Computing and Networking – ISPA 2006 Workshops. XXXVII, 1141 pages. 2006.

Vol. 4330: M. Guo, L.T. Yang, B. Di Martino, H.P. Zima, J. Dongarra, F. Tang (Eds.), Parallel and Distributed Processing and Applications. XVIII, 953 pages. 2006.

Vol. 4329: R. Barua, T. Lange (Eds.), Progress in Cryptology - INDOCRYPT 2006. X, 454 pages. 2006.

Vol. 4328: D. Penkler, M. Reitenspiess, F. Tam (Eds.), Service Availability. X, 289 pages. 2006.

Vol. 4327: M. Baldoni, U. Endriss (Eds.), Declarative Agent Languages and Technologies IV. VIII, 257 pages. 2006. (Sublibrary LNAI).

Vol. 4326: S. Göbel, R. Malkewitz, I. Iurgel (Eds.), Technologies for Interactive Digital Storytelling and Entertainment. X, 384 pages. 2006.

Vol. 4325: J. Cao, I. Stojmenovic, X. Jia, S.K. Das (Eds.), Mobile Ad-hoc and Sensor Networks. XIX, 887 pages. 2006.

Vol. 4323: G. Doherty, A. Blandford (Eds.), Interactive Systems. XI, 269 pages. 2007.

Vol. 4320: R. Gotzhein, R. Reed (Eds.), System Analysis and Modeling: Language Profiles. X, 229 pages. 2006.

Vol. 4319: L.-W. Chang, W.-N. Lie (Eds.), Advances in Image and Video Technology. XXVI, 1347 pages. 2006.

Vol. 4318: H. Lipmaa, M. Yung, D. Lin (Eds.), Information Security and Cryptology. XI, 305 pages. 2006.

Vol. 4317: S.K. Madria, K.T. Claypool, R. Kannan, P. Uppuluri, M.M. Gore (Eds.), Distributed Computing and Internet Technology. XIX, 466 pages. 2006.

Vol. 4316: M.M. Dalkilic, S. Kim, J. Yang (Eds.), Data Mining and Bioinformatics. VIII, 197 pages. 2006. (Sublibrary LNBI).

Vol. 4314: C. Freksa, M. Kohlhase, K. Schill (Eds.), KI 2006: Advances in Artificial Intelligence. XII, 458 pages. 2007. (Sublibrary LNAI).

Vol. 4313: T. Margaria, B. Steffen (Eds.), Leveraging Applications of Formal Methods. IX, 197 pages. 2006.

Vol. 4312: S. Sugimoto, J. Hunter, A. Rauber, A. Morishima (Eds.), Digital Libraries: Achievements, Challenges and Opportunities. XVIII, 571 pages. 2006.

Vol. 4311: K. Cho, P. Jacquet (Eds.), Technologies for Advanced Heterogeneous Networks II. XI, 253 pages. 2006.

Vol. 4309: P. Inverardi, M. Jazayeri (Eds.), Software Engineering Education in the Modern Age. VIII, 207 pages. 2006.

Vol. 4308: S. Chaudhuri, S.R. Das, H.S. Paul, S. Tirthapura (Eds.), Distributed Computing and Networking. XIX, 608 pages. 2006.

Vol. 4307: P. Ning, S. Qing, N. Li (Eds.), Information and Communications Security. XIV, 558 pages. 2006.

Vol. 4306: Y. Avrithis, Y. Kompatsiaris, S. Staab, N.E. O'Connor (Eds.), Semantic Multimedia. XII, 241 pages. 2006.

Vol. 4305: A.A. Shvartsman (Ed.), Principles of Distributed Systems. XIII, 441 pages. 2006.

Vol. 4304: A. Sattar, B.-H. Kang (Eds.), AI 2006: Advances in Artificial Intelligence. XXVII, 1303 pages. 2006. (Sublibrary LNAI).

Vol. 4303: A. Hoffmann, B.-H. Kang, D. Richards, S. Tsumoto (Eds.), Advances in Knowledge Acquisition and Management. XI, 259 pages. 2006. (Sublibrary LNAI).

Vol. 4302: J. Domingo-Ferrer, L. Franconi (Eds.), Privacy in Statistical Databases. XI, 383 pages. 2006.

Vol. 4301: D. Pointcheval, Y. Mu, K. Chen (Eds.), Cryptology and Network Security. XIII, 381 pages. 2006.

Vol. 4300: Y.Q. Shi (Ed.), Transactions on Data Hiding and Multimedia Security I. IX, 139 pages. 2006.

Vol. 4299: S. Renals, S. Bengio, J.G. Fiscus (Eds.), Machine Learning for Multimodal Interaction. XII, 470 pages. 2006.

Vol. 4297: Y. Robert, M. Parashar, R. Badrinath, V.K. Prasanna (Eds.), High Performance Computing - HiPC 2006. XXIV, 642 pages. 2006.

Vol. 4296: M.S. Rhee, B. Lee (Eds.), Information Security and Cryptology – ICISC 2006. XIII, 358 pages. 2006.

Vol. 4295: J.D. Carswell, T. Tezuka (Eds.), Web and Wireless Geographical Information Systems. XI, 269 pages. 2006.

Vol. 4294: A. Dan, W. Lamersdorf (Eds.), Service-Oriented Computing – ICSOC 2006. XIX, 653 pages. 2006.

Vol. 4293: A. Gelbukh, C.A. Reyes-Garcia (Eds.), MICAI 2006: Advances in Artificial Intelligence. XXVIII, 1232 pages. 2006. (Sublibrary LNAI).

Vol. 4292: G. Bebis, R. Boyle, B. Parvin, D. Koracin, P. Remagnino, A. Nefian, G. Meenakshisundaram, V. Pascucci, J. Zara, J. Molineros, H. Theisel, T. Malzbender (Eds.), Advances in Visual Computing, Part II. XXXII, 906 pages. 2006.

¥452.00元

Table of Contents

Hardware Verification Track

Tools Track

Software Testing Track

Model Checking PSL Using HOL and SMV

Thomas Tuerk[1,*], Klaus Schneider[1], and Mike Gordon[2]

[1] Reactive Systems Group
Department of Computer Science, University of Kaiserslautern
P.O. Box 3049, 67653 Kaiserslautern, Germany
http://rsg.informatik.uni-kl.de
[2] University of Cambridge Computer Laboratory
William Gates Building, JJ Thomson Avenue, Cambridge CB3 0FD, United Kingdom
http://www.cl.cam.ac.uk

Abstract. In our previous work, we formally validated the correctness of a translation from a subset of Accellera's Property Specification Language (PSL) to linear temporal logic (LTL) using the HOL theorem prover. We also built an interface from HOL to the SMV model checker based on a formal translation of LTL to ω-automata. In the present paper, we describe how this work has been extended and combined to produce a model checking infrastructure for a significant subset of PSL that works by translating model checking problems to equivalent checks for the existence of fair paths through a Kripke structure specified in higher order logic. This translation is done by theorem proving in HOL, so it is proven to be correct. The existence check is carried out using the interface from HOL to SMV. Moreover, we have applied our infrastructure to implement a tool for validating the soundness of a separate PSL model checker.

1 Introduction

The Property Specification Language (PSL) [1] is an industrial-strength temporal logic. It was developed by the Functional Verification Technical Committee of Accellera based on IBM's Sugar language [3] and has now become an IEEE standard. It is designed both for formal verification and for simulation and has been described as the most popular property specification language in industry [10].

The linear time subset of PSL is a complex language that includes many special cases with subtle semantics. It is well known how LTL can be translated to equivalent ω-automata [30,9,13,12,23], but PSL additionally provides a reset (abort) operator whose semantics has been the subject of debate. In order to study the impact of different kinds of abort operators on the complexity of the translation and verification, a logic RLTL [2] was introduced that extends LTL by a reset operator. It turned out that, in the worst case, Version 1.01 of PSL lead to a non-elementary blow-up in the translation to ω-automata. For this reason, the semantics of PSL's reset operator were changed in Version 1.1 (the current

[*] This work has been done while this author visited the University of Cambridge Computer Laboratory.

E. Bin, A. Ziv, and S. Ur (Eds.): HVC 2006, LNCS 4383, pp. 1–15, 2007.

version). Thus, a significant subset of PSL can now be translated to RLTL. A further translation from RLTL to LTL has already been presented in [2].

Because of the subtle semantics of PSL, it is non-trivial to ensure that implementations accurately reflect the official language standard. Thus, we feel that there is value in using automated formal methods to reason about the semantics of PSL in general, and to verify model checking algorithms for this logic. PSL has already been deeply embedded in HOL [15] and a translation from a significant subset of PSL to ω-automata via RLTL and LTL has been verified [26,25]. However, in this previous work only the correctness of these translations has been proved.

In this paper, we use revised versions of the correctness translation theorems to create PSL implementation infrastructure directly on top of the formalisation of the standard PSL semantics. Model checking problems for PSL can be handled fully automatically. We have used this infrastructure to build a specific tool to check the accuracy of an implementation of PSL used by IBM's RuleBase CTL model checker. We were able to detect an incorrectness (unknown to us, but known to IBM) in the implementation of clocked `aborts` (they are treated as synchronous but should have been asynchronous).

Our infrastructure includes formal translators, implemented by theorem-proving in HOL, from the linear time fragment of PSL to LTL and from LTL to ω-automata. Although these are based on previous work, they were largely rewritten so that they could be turned into a new automatic tool for translating PSL to automata. To check the existence of fair paths, we use a link from HOL to SMV. This is based on Schneider's earlier work, though we changed from a shallow to a deep embedding of LTL in HOL and modified many details. Model checking problems for PSL can be translated, using theorem proving, to equivalent checks for the existence of fair paths through a Kripke structure. A proof of the correctness of the emptiness check is created by these translation procedures. The resulting check is finally performed by SMV [19].

The rest of this paper is organised as follows. The formalisms we use are explained in the next section. We then briefly sketch translations between them. In Section 4, we describe the infrastructure and in Section 5, we outline its application to build a tool to validate the handling of PSL by RuleBase. Finally, we draw some conclusions and show directions for future work.

2 Basic Notions

Temporal logics like LTL, RLTL and PSL use propositional logic to describe (static) properties of the current point of time. The semantics of temporal properties is based on sequences of points of time called *paths*, which are usually defined by transition systems. Thus, we first define propositional logic, paths and transition systems in this section. Then, the logics LTL, RLTL, and PSL are presented. Finally, ω-automata are introduced.

Definition 1 (Propositional Logic). *Let $\mathcal{V}$ be a set of variables. Then, the* set of propositional formulas *over $\mathcal{V}$ (short $\mathsf{prop}_{\mathcal{V}}$) is recursively given as follows:*

- *each variable $v \in \mathcal{V}$ is a propositional formula*
- $\neg\varphi \in \mathsf{prop}_{\mathcal{V}}$, *if* $\varphi \in \mathsf{prop}_{\mathcal{V}}$
- $\varphi \wedge \psi \in \mathsf{prop}_{\mathcal{V}}$, *if* $\varphi, \psi \in \mathsf{prop}_{\mathcal{V}}$

An assignment *over $\mathcal{V}$ is a subset of $\mathcal{V}$. In our context, assignments are also called* states. *The set of all states over $\mathcal{V}$, which is the power set of $\mathcal{V}$, is denoted by $\mathcal{P}(\mathcal{V})$. The semantics of a propositional formula with respect to a state s is given by the relation $\models_{\mathsf{prop}}$ that is defined as follows:*

- $s \models_{\mathsf{prop}} v$ *iff* $v \in s$
- $s \models_{\mathsf{prop}} \neg\varphi$ *iff* $s \not\models_{\mathsf{prop}} \varphi$
- $s \models_{\mathsf{prop}} \varphi \wedge \psi$ *iff* $s \models_{\mathsf{prop}} \varphi$ *and* $s \models_{\mathsf{prop}} \psi$

If $s \models_{\mathsf{prop}} \varphi$ holds, then the assignment s is said to satisfy *the formula φ.*

We use the operators $\vee$, $\rightarrow$ and $\leftrightarrow$ and the constants true and false as syntactic sugar with their usual meaning.

A finite word v over a set Σ of length $|v| = n+1$ is a function $v : \{0, \ldots n\} \rightarrow \Sigma$. An infinite word v over Σ is a function $v : \mathbb{N} \rightarrow \Sigma$ and its length is denoted by $|v| = \infty$. The set Σ is called the *alphabet* and the elements of Σ are called *letters*. The finite word of length 0 is called the *empty word* (denoted by ε). For reasons of simplicity, $v(i)$ is often denoted by v^i for $i \in \mathbb{N}$. Using this notation, words are often given in the form $v^0 v^1 v^2 \ldots v^n$ or $v^0 v^1 \ldots$. The set of all finite and infinite words over Σ is denoted by Σ^* and Σ^ω, respectively.

Counting starts from zero, i. e. v^{i-1} refers to the i-th letter of v. Furthermore, $v^{i..}$ denotes the suffix of v starting at position i, i. e. $v^{i..} = v^i v^{i+1} \ldots$ for all $i < |v|$. The finite word $v^i v^{i+1} \ldots v^j$ is denoted by $v^{i..j}$. Notice that in case $j < i$, the expression $v^{i..j}$ evaluates to the empty word ε. For two words v_1, v_2 with $v_1 \in \Sigma^*$, we write $v_1 v_2$ for their concatenation. The union $v_1 \cup v_2$ of two words v_1, v_2 with $|v_1| = |v_2|$ over sets is defined as the word v with $|v| = |v_1| = |v_2|$ and $v^j = v_1^j \cup v_2^j$ for all $j < |v|$. Analogously, the intersection $v_1 \cap v_2$ of v_1 and v_2 is defined. We write l^ω for the infinite word v with $v^j = l$ for all j.

2.1 Kripke Structures

Systems used with model checking techniques are usually given as labelled transition systems that are often called Kripke structures. In this paper, we use symbolically represented Kripke structures as usual in symbolic model checking.

Definition 2 (Symbolically Represented Kripke Structures). *A symbolically represented Kripke structure $\mathcal{K}$ over a set of variables $\mathcal{V}$ is a tuple $\mathcal{K} = (\mathcal{I}, \mathcal{R})$ such that*

- *$\mathcal{I}$ is a propositional formula over $\mathcal{V}$*
- *$\mathcal{R}$ is a propositional formula over $\mathcal{V} \cup \{\mathtt{X}v \mid v \in \mathcal{V}\}$*

A path *p through $\mathcal{K} = (\mathcal{I}, \mathcal{R})$ is an infinite word over $\mathcal{V}$ such that for all i, the relation $p^i \cup \{\mathtt{X}v \mid v \in p^{i+1}\} \models_{\mathsf{prop}} \mathcal{R}$ holds. A path p is called* initial, *iff $p^0 \models_{\mathsf{prop}} \mathcal{I}$ holds. A path is called* fair *according to some propositional formula f, called the* fairness condition, *iff infinitely many letters of p satisfy the fairness condition, i. e. iff the set $\{i \mid p^i \models_{\mathsf{prop}} f\}$ is infinite. The set of all initial paths through $\mathcal{K}$ is denoted by $\mathsf{IPath}(\mathcal{K})$. The set of all initial paths that satisfy all fairness constraints in the set fc is denoted by $\mathsf{IPath}_{\mathsf{fair}}(\mathcal{K}, fc)$.*

According to this definition, the new variable $\mathtt{X}v$ is used to denote the value of the variable v at the next state. It is often convenient to evaluate a whole propositional formula instead of just one variable at the next state, so the $\mathbf{X}$ operator is introduced as a shorthand for replacing every occurrence of a variable v by $\mathtt{X}v$ in a propositional formula. Similarly, $\mathbf{X}$ is also used to replace every variable v in a set by $\mathtt{X}v$.

2.2 Linear Temporal Logic (LTL)

Linear Temporal Logic (LTL) has been proposed for the specification of reactive systems by Pnueli in [20]. LTL essentially consists of propositional logic enriched with the temporal operators X and $\underline{\mathsf{U}}$. The formula $\mathsf{X}\varphi$ means that the property φ holds at the next point of time, $\varphi \ \underline{\mathsf{U}}\ \psi$ means that φ holds until ψ holds and that ψ eventually holds.

Definition 3 (Syntax of Linear Temporal Logic (LTL)). *The set $\mathsf{ltl}_{\mathcal{V}}$ of* LTL *formulas over a given set of variables $\mathcal{V}$ is defined as follows:*

- $p \in \mathsf{ltl}_{\mathcal{V}}$ *for all* $p \in \mathsf{prop}_{\mathcal{V}}$
- $\neg\varphi$, $\varphi \wedge \psi \in \mathsf{ltl}_{\mathcal{V}}$, *if* $\varphi, \psi \in \mathsf{ltl}_{\mathcal{V}}$
- $\mathsf{X}\varphi$, $\varphi \ \underline{\mathsf{U}}\ \psi \in \mathsf{ltl}_{\mathcal{V}}$, *if* $\varphi, \psi \in \mathsf{ltl}_{\mathcal{V}}$

Further temporal operators can be defined as syntactic sugar, for example, $\mathsf{F}\varphi := (\mathsf{true}\ \underline{\mathsf{U}}\ \psi)$, $\mathsf{G}\varphi := \neg\mathsf{F}\neg\varphi$, $\varphi\ \mathsf{U}\ \psi := \varphi\ \underline{\mathsf{U}}\ \psi \vee \mathsf{G}\varphi$, and $\varphi\ \mathsf{B}\ \psi := \neg(\neg\varphi)\ \underline{\mathsf{U}}\ \psi$. LTL with the operators $\underline{\mathsf{U}}$ and X is, however, already expressively complete with respect to the first order theory of linear orders [23].

Definition 4 (Semantics of Linear Temporal Logic (LTL)). *For $b \in \mathsf{prop}_{\mathcal{V}}$ and $\varphi, \psi \in \mathsf{ltl}_{\mathcal{V}}$ the semantics of* LTL *with respect to an infinite word $v \in \mathcal{P}(\mathcal{V})^{\omega}$ and a point of time $t \in \mathbb{N}$ is given as follows:*

- $v \models_{\mathsf{ltl}}^{t} b$ *iff* $v^t \models_{\mathsf{prop}} b$
- $v \models_{\mathsf{ltl}}^{t} \neg\varphi$ *iff* $v \not\models_{\mathsf{ltl}}^{t} \varphi$
- $v \models_{\mathsf{ltl}}^{t} \varphi \wedge \psi$ *iff* $v \models_{\mathsf{ltl}}^{t} \varphi$ *and* $v \models_{\mathsf{ltl}}^{t} \psi$
- $v \models_{\mathsf{ltl}}^{t} \mathsf{X}\varphi$ *iff* $v \models_{\mathsf{ltl}}^{t+1} \varphi$
- $v \models_{\mathsf{ltl}}^{t} \varphi\ \underline{\mathsf{U}}\ \psi$ *iff* $\exists k.\ k \geq t \ \wedge\ v \models_{\mathsf{ltl}}^{k} \psi \ \wedge\ \forall j.\ t \leq j < k \rightarrow v \models_{\mathsf{ltl}}^{j} \varphi$

A word $v \in \mathcal{P}(\mathcal{V})^{\omega}$ satisfies *a* LTL *formula $\varphi \in \mathsf{ltl}_{\mathcal{V}}$ (written as $v \models_{\mathsf{ltl}} \varphi$) iff $v \models_{\mathsf{ltl}}^{0} \varphi$; a Kripke structure $\mathcal{K}$* satisfies *φ (denoted $\mathcal{K} \models_{\mathsf{ltl}} \varphi$) iff all paths $v \in \mathsf{IPath}(\mathcal{K})$ satisfy φ.*