Doron A. Peled
Yih-Kuen Tsay (Eds.)

LNCS 3707

Automated Technology for Verification and Analysis

Third International Symposium, ATVA 2005
Taipei, Taiwan, October 2005
Proceedings

Springer

Doron A. Peled Yih-Kuen Tsay (Eds.)

Automated Technology for Verification and Analysis

Third International Symposium, ATVA 2005
Taipei, Taiwan, October 4-7, 2005
Proceedings

Springer

Volume Editors

Doron A. Peled
University of Warwick
Department of Computer Science
Coventry, CV4 7AL, UK
E-mail: doron@dcs.warwick.ac.uk

Yih-Kuen Tsay
National Taiwan University
Department of Information Management
No. 1, Sec. 4, Roosevelt Rd., Taipei 106, Taiwan (ROC)
E-mail: tsay@im.ntu.edu.tw

Library of Congress Control Number: 2005932760

CR Subject Classification (1998): B.1.2, B.2.2, B.5.2, B.6, B.7.2, C.2, C.3, D.2, D.3, F.3

ISSN	0302-9743
ISBN-10	3-540-29209-8 Springer Berlin Heidelberg New York
ISBN-13	978-3-540-29209-8 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11562948 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 3707

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

Editorial Board

Lecture Notes in Computer Science

For information about Vols. 1–3629

please contact your bookseller or Springer

Vol. 3728: V. Paliouras, J. Vounckx, D. Verkest (Eds.), Integrated Circuit and System Design. XV, 753 pages. 2005.

Vol. 3726: L.T. Yang, O.F. Rana, B. Di Martino, J. Dongarra (Eds.), High Performance Computing and Communcations. XXVI, 1116 pages. 2005.

Vol. 3725: D. Borrione, W. Paul (Eds.), Correct Hardware Design and Verification Methods. XII, 412 pages. 2005.

Vol. 3718: V.G. Ganzha, E.W. Mayr, E.V. Vorozhtsov (Eds.), Computer Algebra in Scientific Computing. XII, 502 pages. 2005.

Vol. 3717: B. Gramlich (Ed.), Frontiers of Combining Systems. X, 321 pages. 2005. (Subseries LNAI).

Vol. 3715: E. Dawson, S. Vaudenay (Eds.), Progress in Cryptology – Mycrypt 2005. XI, 329 pages. 2005.

Vol. 3714: H. Obbink, K. Pohl (Eds.), Software Product Lines. XIII, 235 pages. 2005.

Vol. 3713: L. Briand, C. Williams (Eds.), Model Driven Engineering Languages and Systems. XV, 722 pages. 2005.

Vol. 3712: R. Reussner, J. Mayer, J.A. Stafford, S. Overhage, S. Becker, P.J. Schroeder (Eds.), Quality of Software Architectures and Software Quality. XIII, 289 pages. 2005.

Vol. 3711: F. Kishino, Y. Kitamura, H. Kato, N. Nagata (Eds.), Entertainment Computing - ICEC 2005. XXIV, 540 pages. 2005.

Vol. 3710: M. Barni, I. Cox, T. Kalker, H.J. Kim (Eds.), Digital Watermarking. XII, 485 pages. 2005.

Vol. 3708: J. Blanc-Talon, W. Philips, D. Popescu, P. Scheunders (Eds.), Advanced Concepts for Intelligent Vision Systems. XXII, 725 pages. 2005.

Vol. 3707: D.A. Peled, Y.-K. Tsay (Eds.), Automated Technology for Verification and Analysis. XII, 506 pages. 2005.

Vol. 3706: H. Fuks, S. Lukosch, A.C. Salgado (Eds.), Groupware: Design, Implementation, and Use. XII, 378 pages. 2005.

Vol. 3703: F. Fages, S. Soliman (Eds.), Principles and Practice of Semantic Web Reasoning. VIII, 163 pages. 2005.

Vol. 3702: B. Beckert (Ed.), Automated Reasoning with Analytic Tableaux and Related Methods. XIII, 343 pages. 2005. (Subseries LNAI).

Vol. 3699: C.S. Calude, M.J. Dinneen, G. Păun, M. J. Pérez-Jiménez, G. Rozenberg (Eds.), Unconventional Computation. XI, 267 pages. 2005.

Vol. 3698: U. Furbach (Ed.), KI 2005: Advances in Artificial Intelligence. XIII, 409 pages. 2005. (Subseries LNAI).

Vol. 3697: W. Duch, J. Kacprzyk, E. Oja, S. Zadrożny (Eds.), Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005, Part II. XXXII, 1045 pages. 2005.

Vol. 3696: W. Duch, J. Kacprzyk, E. Oja, S. Zadrożny (Eds.), Artificial Neural Networks: Biological Inspirations – ICANN 2005, Part I. XXXI, 703 pages. 2005.

Vol. 3695: M.R. Berthold, R. Glen, K. Diederichs, O. Kohlbacher, I. Fischer (Eds.), Computational Life Sciences. XI, 277 pages. 2005. (Subseries LNBI).

Vol. 3694: M. Malek, E. Nett, N. Suri (Eds.), Service Availability. VIII, 213 pages. 2005.

Vol. 3693: A.G. Cohn, D.M. Mark (Eds.), Spatial Information Theory. XII, 493 pages. 2005.

Vol. 3692: R. Casadio, G. Myers (Eds.), Algorithms in Bioinformatic. X, 436 pages. 2005. (Subseries LNBI).

Vol. 3691: A. Gagalowicz, W. Philips (Eds.), Computer Analysis of Images and Patterns. XIX, 865 pages. 2005.

Vol. 3690: M. Pěchouček, P. Petta, L.Z. Varga (Eds.), Multi-Agent Systems and Applications IV. XVII, 667 pages. 2005. (Subseries LNAI).

Vol. 3687: S. Singh, M. Singh, C. Apte, P. Perner (Eds.), Pattern Recognition and Image Analysis, Part II. XXV, 809 pages. 2005.

Vol. 3686: S. Singh, M. Singh, C. Apte, P. Perner (Eds.), Pattern Recognition and Data Mining, Part I. XXVI, 689 pages. 2005.

Vol. 3685: V. Gorodetsky, I. Kotenko, V. Skormin (Eds.), Computer Network Security. XIV, 480 pages. 2005.

Vol. 3684: R. Khosla, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part IV. LXXIX, 933 pages. 2005. (Subseries LNAI).

Vol. 3683: R. Khosla, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part III. LXXX, 1397 pages. 2005. (Subseries LNAI).

Vol. 3682: R. Khosla, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part II. LXXIX, 1371 pages. 2005. (Subseries LNAI).

Vol. 3681: R. Khosla, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part I. LXXX, 1319 pages. 2005. (Subseries LNAI).

Vol. 3679: S.d.C. di Vimercati, P. Syverson, D. Gollmann (Eds.), Computer Security – ESORICS 2005. XI, 509 pages. 2005.

Vol. 3678: A. McLysaght, D.H. Huson (Eds.), Comparative Genomics. VIII, 167 pages. 2005. (Subseries LNBI).

Vol. 3677: J. Dittmann, S. Katzenbeisser, A. Uhl (Eds.), Communications and Multimedia Security. XIII, 360 pages. 2005.

Vol. 3676: R. Glück, M. Lowry (Eds.), Generative Programming and Component Engineering. XI, 448 pages. 2005.

Vol. 3675: Y. Luo (Ed.), Cooperative Design, Visualization, and Engineering. XI, 264 pages. 2005.

Vol. 3674: W. Jonker, M. Petković (Eds.), Secure Data Management. X, 241 pages. 2005.

Vol. 3673: S. Bandini, S. Manzoni (Eds.), AI*IA 2005: Advances in Artificial Intelligence. XIV, 614 pages. 2005. (Subseries LNAI).

Vol. 3672: C. Hankin, I. Siveroni (Eds.), Static Analysis. X, 369 pages. 2005.

Vol. 3671: S. Bressan, S. Ceri, E. Hunt, Z.G. Ives, Z. Bellahsène, M. Rys, R. Unland (Eds.), Database and XML Technologies. X, 239 pages. 2005.

Vol. 3670: M. Bravetti, L. Kloul, G. Zavattaro (Eds.), Formal Techniques for Computer Systems and Business Processes. XIII, 349 pages. 2005.

Vol. 3669: G.S. Brodal, S. Leonardi (Eds.), Algorithms – ESA 2005. XVIII, 901 pages. 2005.

Vol. 3668: M. Gabbrielli, G. Gupta (Eds.), Logic Programming. XIV, 454 pages. 2005.

Vol. 3666: B.D. Martino, D. Kranzlmüller, J. Dongarra (Eds.), Recent Advances in Parallel Virtual Machine and Message Passing Interface. XVII, 546 pages. 2005.

Vol. 3665: K. S. Candan, A. Celentano (Eds.), Advances in Multimedia Information Systems. X, 221 pages. 2005.

Vol. 3664: C. Türker, M. Agosti, H.-J. Schek (Eds.), Peer-to-Peer, Grid, and Service-Orientation in Digital Library Architectures. X, 261 pages. 2005.

Vol. 3663: W.G. Kropatsch, R. Sablatnig, A. Hanbury (Eds.), Pattern Recognition. XIV, 512 pages. 2005.

Vol. 3662: C. Baral, G. Greco, N. Leone, G. Terracina (Eds.), Logic Programming and Nonmonotonic Reasoning. XIII, 454 pages. 2005. (Subseries LNAI).

Vol. 3661: T. Panayiotopoulos, J. Gratch, R. Aylett, D. Ballin, P. Olivier, T. Rist (Eds.), Intelligent Virtual Agents. XIII, 506 pages. 2005. (Subseries LNAI).

Vol. 3660: M. Beigl, S. Intille, J. Rekimoto, H. Tokuda (Eds.), UbiComp 2005: Ubiquitous Computing. XVII, 394 pages. 2005.

Vol. 3659: J.R. Rao, B. Sunar (Eds.), Cryptographic Hardware and Embedded Systems – CHES 2005. XIV, 458 pages. 2005.

Vol. 3658: V. Matoušek, P. Mautner, T. Pavelka (Eds.), Text, Speech and Dialogue. XV, 460 pages. 2005. (Subseries LNAI).

Vol. 3657: F.S. de Boer, M.M. Bonsangue, S. Graf, W.-P. de Roever (Eds.), Formal Methods for Components and Objects. VIII, 325 pages. 2005.

Vol. 3656: M. Kamel, A. Campilho (Eds.), Image Analysis and Recognition. XXIV, 1279 pages. 2005.

Vol. 3655: A. Aldini, R. Gorrieri, F. Martinelli (Eds.), Foundations of Security Analysis and Design III. VII, 273 pages. 2005.

Vol. 3654: S. Jajodia, D. Wijesekera (Eds.), Data and Applications Security XIX. X, 353 pages. 2005.

Vol. 3653: M. Abadi, L. de Alfaro (Eds.), CONCUR 2005 – Concurrency Theory. XIV, 578 pages. 2005.

Vol. 3652: A. Rauber, S. Christodoulakis, A M. Tjoa (Eds.), Research and Advanced Technology for Digital Libraries. XVIII, 545 pages. 2005.

Vol. 3651: R. Dale, K.-F. Wong, J. Su, O.Y. Kwong (Eds.), Natural Language Processing – IJCNLP 2005. XXI, 1031 pages. 2005.

Vol. 3650: J. Zhou, J. Lopez, R.H. Deng, F. Bao (Eds.), Information Security. XII, 516 pages. 2005.

Vol. 3649: W.M. P. van der Aalst, B. Benatallah, F. Casati, F. Curbera (Eds.), Business Process Management. XII, 472 pages. 2005.

Vol. 3648: J.C. Cunha, P.D. Medeiros (Eds.), Euro-Par 2005 Parallel Processing. XXXVI, 1299 pages. 2005.

Vol. 3646: A. F. Famili, J.N. Kok, J.M. Peña, A. Siebes, A. Feelders (Eds.), Advances in Intelligent Data Analysis VI. XIV, 522 pages. 2005.

Vol. 3645: D.-S. Huang, X.-P. Zhang, G.-B. Huang (Eds.), Advances in Intelligent Computing, Part II. XIII, 1010 pages. 2005.

Vol. 3644: D.-S. Huang, X.-P. Zhang, G.-B. Huang (Eds.), Advances in Intelligent Computing, Part I. XXVII, 1101 pages. 2005.

Vol. 3643: R. Moreno Díaz, F. Pichler, A. Quesada Arencibia (Eds.), Computer Aided Systems Theory – EUROCAST 2005. XIV, 629 pages. 2005.

Vol. 3642: D. Ślezak, J. Yao, J.F. Peters, W. Ziarko, X. Hu (Eds.), Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, Part II. XXIII, 738 pages. 2005. (Subseries LNAI).

Vol. 3641: D. Ślezak, G. Wang, M. Szczuka, I. Düntsch, Y. Yao (Eds.), Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, Part I. XXIV, 742 pages. 2005. (Subseries LNAI).

Vol. 3639: P. Godefroid (Ed.), Model Checking Software. XI, 289 pages. 2005.

Vol. 3638: A. Butz, B. Fisher, A. Krüger, P. Olivier (Eds.), Smart Graphics. XI, 269 pages. 2005.

Vol. 3637: J. M. Moreno, J. Madrenas, J. Cosp (Eds.), Evolvable Systems: From Biology to Hardware. XI, 227 pages. 2005.

Vol. 3636: M.J. Blesa, C. Blum, A. Roli, M. Sampels (Eds.), Hybrid Metaheuristics. XII, 155 pages. 2005.

Vol. 3634: L. Ong (Ed.), Computer Science Logic. XI, 567 pages. 2005.

Vol. 3633: C. Bauzer Medeiros, M. Egenhofer, E. Bertino (Eds.), Advances in Spatial and Temporal Databases. XIII, 433 pages. 2005.

Vol. 3632: R. Nieuwenhuis (Ed.), Automated Deduction – CADE-20. XIII, 459 pages. 2005. (Subseries LNAI).

Vol. 3631: J. Eder, H.-M. Haav, A. Kalja, J. Penjam (Eds.), Advances in Databases and Information Systems. XIII, 393 pages. 2005.

Vol. 3630: M.S. Capcarrere, A.A. Freitas, P.J. Bentley, C.G. Johnson, J. Timmis (Eds.), Advances in Artificial Life. XIX, 949 pages. 2005. (Subseries LNAI).

¥604.16

Preface

The Automated Technology for Verification and Analysis (ATVA) international symposium series was initiated in 2003, responding to a growing interest in formal verification spurred by the booming IT industry, particularly hardware design and manufacturing in East Asia. Its purpose is to promote research on automated verification and analysis in the region by providing a forum for interaction between the regional and the international research/industrial communities of the field. ATVA 2005, the third of the ATVA series, was held in Taipei, Taiwan, October 4–7, 2005. The main theme of the symposium encompasses design, complexities, tools, and applications of automated methods for verification and analysis. The symposium was co-located and had a two-day overlap with FORTE 2005, which was held October 2–5, 2005.

We received a total of 95 submissions from 17 countries. Each submission was assigned to three Program Committee members, who were helped by their subreviewers, for rigorous and fair evaluation. The final deliberation by the Program Committee was conducted over email for a duration of about 10 days after nearly all review reports had been collected. In the end, 33 papers were selected for inclusion in the program. ATVA 2005 had three keynote speeches given respectively by Amir Pnueli (joint with FORTE 2005), Zohar Manna, and Wolfgang Thomas. The main symposium was preceded by a tutorial day, consisting of three two-hour lectures given also by the keynote speakers.

ATVA 2005 was supported by National Science Council, Ministry of Education, and Academia Sinica of Taiwan and also by the Center for Information and Electronics Technologies at National Taiwan University and Cadence Design Systems. Their generous sponsorships are gratefully acknowledged. We would like to thank the Program Committee members and their subreviewers for the hard work in evaluating the submissions and selecting the program. We thank the keynote speakers for their extra effort in delivering the tutorials. We thank the Steering Committee for their advice, particularly Farn Wang, who also served as program chair of the two previous ATVA symposia and of FORTE 2005, for providing many valuable suggestions and for being very cooperative with the joint events of ATVA 2005 and FORTE 2005.

For administrative support, we thank the Department of Information Management and the Department of Electrical Engineering at National Taiwan University. In particular, we thank Mr. Yu-Fang Chen for maintaining the conference Web site among many other administrative chores. We thank also the MyReview team for making available a free and convenient submission system.

October 2005 Doron A. Peled and Yih-Kuen Tsay

Organization

Steering Committee

E. Allen Emerson	University of Texas at Austin, USA
Oscar H. Ibarra	University of California, Santa Barbara, USA
Insup Lee	University of Pennsylvania, USA
Doron A. Peled	University of Warwick, UK
Farn Wang	National Taiwan University, Taiwan
Hsu-Chun Yen	National Taiwan University, Taiwan

General Chair

Insup Lee	University of Pennsylvania, USA

Program Committee

Parosh Aziz Abdulla	Uppsala University, Sweden
Rajeev Alur	University of Pennsylvania, USA
Christel Baier	University of Bonn, Germany
Tevfik Bultan	University of California, Santa Barbara, USA
Yung-Pin Cheng	National Taiwan Normal University, Taiwan
Ching-Tsun Chou	Intel, USA
Jin Song Dong	National University of Singapore, Singapore
Susanne Graf	VERIMAG, France
Teruo Higashino	Osaka University, Japan
Pei-Hsin Ho	Synopsys, USA
Gerard J. Holzmann	NASA/JPL, USA
Pao-Ann Hsiung	National Chung Cheng University, Taiwan
Chung-Yang Huang	National Taiwan University, Taiwan
Oscar H. Ibarra	University of California, Santa Barbara, USA
Bengt Jonsson	Uppsala University, Sweden
Orna Kupferman	Hebrew University, Israel
Robert P. Kurshan	Cadence, USA
Shaoying Liu	Hosei University, Japan
Doron A. Peled	University of Warwick, UK (Co-chair)
Scott Smolka	SUNY, Stony Brook, USA
Yih-Kuen Tsay	National Taiwan University, Taiwan (Co-chair)
Moshe Y. Vardi	Rice University, USA
Bow-Yaw Wang	Academia Sinica, Taiwan
Hsu-Chun Yen	National Taiwan University, Taiwan
Tomohiro Yoneda	Tokyo Institute of Technology, Japan
Lenore Zuck	University of Illinois, Chicago, USA

Local Arrangements

Chung-Yang Huang	National Taiwan University, Taiwan
Bow-Yaw Wang	Academia Sinica, Taiwan

Reviewers

Zaher S. Andraus
Ittai Balaban
Constantinos Bartzis
Samik Basu
Frederic Beal
Aysu Betin-Can
Per Bjesse
Bernhard Boigelot
Chunqing Chen
Yean-Ru Chen
Yu-Fang Chen
Johann Deneux
Yifei Dong
Julien Dorso
Ashvin Dsouza
Lars-Henrik Eriksson
Yi Fang
Yuzheng Feng
Martin Fränzle
Xiang Fu
Noriyuki Fujimoto
Jim Grundy
Anubhav Gupta
Kiyoharu Hamaguchi
Ziyad Hanna
Kunihiko Hiraishi
Chun-Hsian Huang
Geng-Dian Huang
Marcin Jurdzinski
Andreas Kassler
Tomoya Kitani
Barbara König
Gregor von Laszewksi
Martin Leucker
Yuan Fang Li
Shang-Wei Lin
Annie Liu
Shiyong Lu
Rupak Majumdar
Oded Maler
In-Ho Moon
Akio Nakata
Andrei Paun
Andreas Podelski
Hongyang Qu
Jakob Rehof
Kai Salomaa
Pierluigi San Pietro
Hiroyuki Seki
Oleg Sokolsky
Martin Steffen
Scott Stoller
Jun Sun
Ashutosh Trivedi
Ming-Hsien Tsai
Tatsuhiro Tsuchiya
Takaaki Umedu
Björn Victor
Dirk Walther
Dong Wang
Heike Wehrheim
Frank Wolter
Keh-Ren Wu
Ping Yang
Pei Ye
Wang Yi
Fang Yu
Gaoyan Xie

Sponsoring Institutions

National Science Council, Taiwan (ROC)
Ministry of Education, Taiwan
Institute of Information Science, Academia Sinica, Taiwan
National Taiwan University (NTU), Taiwan
Center for Information and Electronics Technologies, NTU, Taiwan
Department of Information Management, NTU, Taiwan
Department of Electrical Engineering, NTU, Taiwan
Cadence Design Systems

Table of Contents

Keynote Speeches

Model Checking

Combined Methods

Timed, Embedded, and Hybrid Systems (I)

Abstraction and Reduction Techniques

Decidability and Complexity

Established Formalisms and Standards

Compositional Verification and Games

Timed, Embedded, and Hybrid Systems (II)

Protocols Analysis, Case Studies, and Tools

Infinite-State and Parameterized Systems

Ranking Abstraction as a Companion to Predicate Abstraction*,**

Amir Pnueli[1,2]

[1] New York University, New York
amir@cs.nyu.edu
[2] Weizmann Institute of Science

Abstract. Predicate abstraction has become one of the most successful methodologies for proving safety properties of programs. Unfortunately, it cannot be used for verifying all liveness properties. In order to handle liveness properties, we introduce the method of *ranking abstraction*. This method augments the analyzed system by a "progress monitor" which observes whether a given ranking function decreases or increases at any step of the program. The fact that the ranking function ranges over a well-founded domain is expressed by a *compassion* (strong fairness) requirement, which states that a function over a well-founded domain cannot decrease infinitely many times without also increasing infinitely many times. In analogy to predicate abstraction which uses a predicate base $\mathcal{P} = \{P_1, \ldots, P_m\}$ consisting of a set of predicates, we augment the program with a *ranking core* $\Delta = \{\delta_1, \ldots, \delta_n\}$ consisting of several ranking components. The augmented system is then abstracted using standard predicate abstraction, but retaining all the compassion requirements. The abstracted augmented system is then model checked for an arbitrary LTL property. The ranking abstraction method is shown to be sound and (relatively) complete for proving all LTL properties, including safety and liveness.

In the presented talk we focus on the strong analogy between predicate abstraction and ranking abstraction. Predicate abstraction can be viewed as a process which determines the best inductive invariant which can be formed as a boolean combination of the predicate base. In a similar way, ranking abstraction can be viewed as a search for the best well-founded global ranking function which can be formed as a lexicographic combination of the ranking components included in the ranking core Δ. In the talk, we present an algorithm for an explicit construction of such a global ranking function. Another important element of the predicate abstraction methodology is that of *abstraction refinement* by which, a coarse abstraction can be refined by analyzing a spurious counterexample. We show that ranking abstraction also possesses an analogous refinement process. We discuss how a spurious counter example can lead to a refinement of either the current predicate base or ranking core.

The talk is based on results obtained through joint research with I. Balaban, Y. Kesten, and L.D. Zuck.

* The full version of this paper is included in the proceedings of FORTE'05.

** This research was supported in part by NSF grant CCR-0205571, ONR grant N00014-99-1-0131, and Israel Science Foundation grant 106/02-1.

D.A. Peled and Y.-K. Tsay (Eds.): ATVA 2005, LNCS 3707, p. 1, 2005.

Termination and Invariance Analysis of Loops

Aaron Bradley and Zohar Manna

Computer Science Department, Stanford University

Abstract. Deductive verification aims to prove deep properties about programs. The classic Floyd-Hoare-style approach to verifying sequential programs reduces program validity queries to first-order validity queries via verification conditions. Proving that a program is totally correct requires proving the safety aspect with invariants and the progress aspect with invariants and ranking functions. Where do the invariants and ranking functions come from?

A verifying compiler that reads program annotations enables the programmer to write desired properties as assertions. Unfortunately, verifying a safety property requires strengthening it to an inductive assertion, while proving termination requires finding ranking functions. The strengthening process often involves writing many tedious facts, while ranking functions are not always intuitive. In practice, programmers do not want or are unable to invent inductive assertions and ranking functions. Instead, the ideal verifying compiler strengthens the given assertions with facts learned through static analysis. Invariant generators are a class of static analyzers that automatically synthesize inductive invariants. Ranking function generators automatically synthesize ranking functions, sometimes with supporting invariants. Together, they reduce the burden on the programmer by automatically learning facts about programs.

In this talk, we discuss our approach to invariant and ranking function generation. A constraint-based method labels program points with parameterized expressions, which encode the shape of the desired inductive assertions or ranking functions. For example, the shape of an inductive invariant could be an inequality between affine combinations of program variables, while the shape of a ranking function could be an affine combination of program variables. It then generates a set of parameterized verification conditions and solves for the parameter values that make them valid. Instantiating the parameterized expressions with these values results in a set of inductive assertions or ranking functions. We discuss recent work for analyzing termination of programs that manipulate variables via affine expressions. We also discuss a constraint-based analysis for programs with integer division and modulo operators. Finally, we present experimental evidence indicating that invariant and ranking function generation is a powerful technique for scaling deductive verification to large programs.

D.A. Peled and Y.-K. Tsay (Eds.): ATVA 2005, LNCS 3707, p. 2, 2005.

Some Perspectives of Infinite-State Verification

Wolfgang Thomas

RWTH Aachen, Lehrstuhl Informatik 7, 52056 Aachen, Germany
thomas@informatik.rwth-aachen.de

Abstract. We report on recent progress in the study of infinite transition systems for which interesting properties (like reachability of designated states) can be checked algorithmically. Two methods for the generation of such models are discussed: the construction from simpler models via operations like unfolding and synchronized product, and the internal representation of state spaces by regular sets of words or trees.

1 Introduction

The method of model-checking has developed largely in the domain of finite system models, and its success in industrial applications is built on highly efficient data structures for system representation. Over infinite models, the situation is different, and for practical applications the field is still in its beginnings. Even simple properties may be undecidable over infinite state spaces, and thus a careful preparatory analysis is necessary in order to determine the possible range of fully automatic verification.

The purpose of the present short survey is to report on some techniques which yield classes of infinite models such that the model-checking problem is decidable for interesting properties. Our presentation is far from complete; it is biased towards results which were obtained in the author's research group and collaborations with other groups (mostly that of D. Caucal, Rennes). We focus on system models in the form of edge-labelled transition graphs; thus a central aspect is the investigation of structural properties of infinite graphs. An alternative and equally fundamental approach for introducing infinite models, which is not discussed in this paper, is to extend finite transition graphs by infinite data structures, for example over the natural or real numbers (as in timed systems).

Transition graphs are considered in the format $G = (V, (E_a)_{a \in \Sigma})$ where V is the set of states (vertices) and where E_a (for a symbol a from a finite alphabet Σ) is the set of a-labelled edges. We write E for the union of the E_a. State-properties may be introduced by subsets V_a of V, where a is from a second label alphabet Γ.

The logics we consider allow to express the reachability relation E^*, the reflexive transitive closure of E, since reachability is the most fundamental property arising in verification. A prominent logic of this kind is monadic second-order logic MSO. It encompasses most standard temporal logics. On the other end,

D.A. Peled and Y.-K. Tsay (Eds.): ATVA 2005, LNCS 3707, pp. 3–10, 2005.

as a kind of minimal logic in this context, we consider FO(R) ("first-order logic with reachability"), the extension of first-order logic by a relation symbol for E^*.

We shall address two methods for constructing infinite transition graphs where model-checking (with respect to MSO or FO(R)) is decidable. First we review the effect of fundamental model constructions – namely, interpretation, unfolding, and synchronized product – on the existence of model-checking procedures. Secondly, we discuss model-checking as based on "regular" internal representations of infinite transition graphs, using finite automata over strings or trees, respectively.

2 Operations on Graphs

2.1 Interpretations

Rabin's Tree Theorem [19] states that the MSO-theory of the infinite binary tree T_2 is decidable (or in other terminology: that model-checking the binary tree with respect to MSO-properties is decidable). We can view T_2 as a graph $(\{1,2\}^*, S_1, S_2)$, where $\{1,2\}^*$ is the set of vertices and S_1, S_2 the successor relations with $S_i = \{(v, vi) \mid v \in \{1,2\}^*\}$. Many other theories were shown decidable (already in [19]) using interpretations in the tree T_2. To show that the model-checking problem for a structure S with respect to formulas of a logic L is decidable one proceeds as follows: One gives an MSO-description of S within the binary tree T_2, and using this one provides a translation of L-formulas φ into MSO-formulas φ' such that $S \models \varphi$ iff $T_2 \models \varphi'$. Taking $L =$ MSO, we see that an MSO-interpretation (i.e., a model description using MSO-formulas) preserves decidability of model-checking with respect to MSO-formulas.

As a simple example of interpretation consider the n-ary branching tree T_n (for $n > 2$), with vertices in the set $\{1, \ldots, n\}^*$ rather than $\{1,2\}^*$ as for T_2. We may represent the vertex $i_1 \ldots i_r$ of T_n by $1^{i_1}2 \ldots 1^{i_r}2$ in T_2. It is easy to give an MSO-definition of the range of this coding in T_2 and to supply the translation $\varphi \mapsto \varphi'$ as above. As a second example, consider a pushdown automaton A with stack alphabet $\{1, \ldots, k\}$ and states $q_1, \ldots, q_m$. Let $G_A = (V_A, E_A)$ be its configuration graph; here V_A consists of A-configurations $(q_j, i_1 \ldots i_r)$ (with state q_j and stack content $i_1 \ldots i_r$, reading i_1 as top symbol), and we restrict to those configurations which are reachable from the initial one (say $(q_1, 1)$). The edge relation E_A is the one-step transition relation of A between configurations. Choosing $n = \max(k, m)$, we can exhibit an MSO-interpretation of G_A in T_n: Just represent configuration $(q_j, i_1 \ldots i_r)$ by the vertex $i_r \ldots i_1 j$ of T_n. Note that then the A-steps lead to local moves in T_n, from one T_n-vertex to another, e.g. in a push step from vertex $i_r \ldots i_1 j$ to a vertex $i_r \ldots i_1 i_0 j'$. These moves are easily definable in MSO, and reachability (from the initial vertex 11) as well. Due to this interpretation, we obtain the fundamental result of Muller and Schupp ([18]): *For the configuration graph of a pushdown automaton, checking MSO-properties is decidable.*

It is known that the ε-closures of the pushdown transition graphs capture precisely those graphs which are MSO-interpretable in T_2 (or equivalently in T_n);