Geoff Sutcliffe
Andrei Voronkov (Eds.)

LNAI 3835

Logic for Programming, Artificial Intelligence, and Reasoning

12th International Conference, LPAR 2005
Montego Bay, Jamaica, December 2005
Proceedings

Springer

Geoff Sutcliffe Andrei Voronkov (Eds.)

Logic for Programming, Artificial Intelligence, and Reasoning

12th International Conference, LPAR 2005
Montego Bay, Jamaica, December 2-6, 2005
Proceedings

Springer

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Volume Editors

Geoff Sutcliffe
University of Miami, Department of Computer Science
P.O. Box 248154, Coral Gables, FL 33124, USA
E-mail: geoff@cs.miami.edu

Andrei Voronkov
University of Manchester, Department of Computer Science
Oxford Road, Manchester M13 9PL, UK
E-mail: voronkov@cs.man.ac.uk

Library of Congress Control Number: 2005936393

CR Subject Classification (1998): I.2.3, I.2, F.4.1, F.3, D.2.4, D.1.6

ISSN	0302-9743
ISBN-10	3-540-30553-X Springer Berlin Heidelberg New York
ISBN-13	978-3-540-30553-8 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11591191 06/3142 5 4 3 2 1 0

Lecture Notes in Artificial Intelligence 3835

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Preface

This volume contains the full papers presented at the 12th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR), held 2-6 December 2006, in Montego Bay, Jamaica. The call for papers attracted 108 full paper submissions, each of which were reviewed by at least three reviewers. The Program Committee accepted the 46 papers that appear in these proceedings. The conference program also included 4 invited talks, by Tom Ball of Microsoft Research, Doug Lenat of Cycorp, Roberto Nieuwenhuis of the Universidad Politécnica de Cataluña, and Allen Van Gelder of the University of California at Santa Cruz. Papers or abstracts for the invited talks are in these proceedings.

In addition to the main program, the conference offered a short paper track, which attracted 13 submissions, of which 12 were accepted, and the Workshop on Emperically Successful Higher Order Logic (ESHOL).

Thanks go to: the authors (of both accepted and rejected papers); the Program Committee and their reviewers; the invited speakers; Christoph Benzmüller, John Harrison, and Carsten Schürmann for organizing ESHOL; Celia Alleyne-Ebanks for administering the conference in Jamaica; the Honorable Minister Phillip Paulwell of the Ministry of Commerce, Science and Technology for opening the conference (and Daphne Simmonds for introducing us to the minister); the Mona Institute of Applied Sciences at the University of the West Indies for their support; Microsoft Research for sponsorship of student regsitrations; the Kurt Gödel Society for taking registrations; and EasyChair for hosting the review process.

October 2006

Geoff Sutcliffe
Andrei Voronkov

Conference Organization

Program Chairs

Geoff Sutcliffe
Andrei Voronkov

Program Committee

Elvira Albert
Maria Alpuente
Matthias Baaz
Christoph Benzmüller
Koen Claessen
Anatoli Degtyarev
Thomas Eiter
Bernd Fischer
Rajeev Goré
Erich Grädel
John Harrison
Miki Hermann
Brahim Hnich
Ian Horrocks
Mateja Jamnik
Neil Jones
Christoph Koch
Christopher Lynch
Michael Maher
Maarten Marx
Catuscia Palamidessi
Peter Patel-Schneider
Jeff Pelletier
Harald Ruess
Carsten Schürmann
Stephan Schulz
John Keith Slaney
Cesare Tinelli
Ashish Tiwari
Margus Veanes

Local Organization

Celia Alleyne-Ebanks
Geoff Sutcliffe

External Reviewers

Andreas Abel
Amal Ahmed
Wolfgang Ahrendt
Anbulagan
Grigoris Antoniou
Puri Arenas
Jürgen Avenhaus
Demis Ballis
Clark Barrett
Peter Baumgartner
Michael Beeson
Leopoldo Bertossi
Gavin Bierman
Bernard Boigelot
Chad Brown
Colin Campbell
Luciano Caroprese
Manuel Carro
Claudio Castellini
Balder ten Cate
Patrice Chalin
Anatoly Chebotarev
Adam Chlipala
Agata Ciabattoni
Manuel Clavel
Jonathan Cohen
Jesús Correas
Stephen Craig
Medhi Dastani
Jeremy Dawson
Anatoli Degtyarev
Stephane Demri
Dan Dougherty
Esra Erdem
Santiago Escobar
Wolfgang Faber
Moreno Falaschi
Chris Fermüller
Massimo Franceschet
Anders Franzén
Carsten Fritz
John Gallagher
Stephane Gaubert
Samir Genaim
Jürgen Giesl
Birte Glimm
Eugene Goldberg
Georges Gonthier
Wolfgang Grieskamp
Yuri Gurevich
Reiner Hähnle
Jay Halcomb
Joe Hendrix
Hugo Herbelin
Mark Hills
Marieke Huisman
Dieter Hutter
Giovambattista Ianni
Rosalie Iemhoff
Pascual Julián Iranzo
Tommi Junttila
Nicolas Kicillof
Joseph Kiniry
Felix Klaedtke
Roman Kontchakov
Sergey Krivoi
Orna Kupferman
Oliver Kutz
Axel Legay
Stephane Lengrand
Martin Leucker
Lei Li
Ninghui Li
Guohui Lin
Christina Lindenberg
John Lloyd
Andrei Lopatenko
Salvador Lucas
Ines Lynce
Alexis Maciel
John Matthews
Farhad Mehta

George Metcalfe
Marino Miculan
Dale Miller
David Mitchell
Alberto Momigliano
José Morales
Ben Moszkowski
Boris Motik
Lev Nachmanson
Robert Nieuwenhuis
Andreas Nonnengart
Michael Norrish
Don Nute
Jan Obdrzalek
Albert Oliveras
Vincent van Oostrom
Sam Owre
Miguel Palomino
Jeff Pan
Grant Passmore
Lawrence C. Paulson
Brigitte Pientka
Andre Platzer
Erik Poll
Andrei Popescu
Steven Prestwich
Arthur Ramer
María José Ramírez
Christophe Ringeissen
Enric Rodríguez-Carbonell
Roberto Rossi
Grigore Rosu
Pritam Roy
Piotr Rudnicki
Jeffrey Sarnat
Roman Schindlauer
Renate Schmidt
Johann Schumann
Thomas Schwentick
Alberto Segre
Anton Setzer
Jatin Shah
Chung-chieh Shan
Jörg Siekmann
Konrad Slind
Maria Sorea
Mark Steedman
Graham Steel
Gernot Stenz
Charles Stewart
Lutz Strassburger
Ofer Strichman
Aaron Stump
Evgenia Ternovska
Sebastiaan Terwijn
Rene Thiemann
Hans Tompits
Leon van der Torre
Dmitry Tsarkov
Xavier Urbain
Alasdair Urquhart
Frank D. Valencia
Alex Vaynberg
Helmut Veith
Gérard Verfaillie
Alicia Villanueva
Fer-Jan de Vries
Emil Weydert
Wayne Wobcke
Stefan Woltran
Rostislav Yavorskiy
Richard Zach
Noam Zeilberger
Evgeny Zolin

Lecture Notes in Artificial Intelligence (LNAI)

Vol. 3835: G. Sutcliffe, A. Voronkov (Eds.), Logic for Programming, Artificial Intelligence, and Reasoning. XIV, 744 pages. 2005.

Vol. 3789: A. Gelbukh, Á. de Albornoz, H. Terashima-Marín (Eds.), MICAI 2005: Advances in Artificial Intelligence. XXVI, 1198 pages. 2005.

Vol. 3735: A. Hoffmann, H. Motoda, T. Scheffer (Eds.), Discovery Science. XVI, 400 pages. 2005.

Vol. 3734: S. Jain, H.U. Simon, E. Tomita (Eds.), Algorithmic Learning Theory. XII, 490 pages. 2005.

Vol. 3721: A. Jorge, L. Torgo, P.B. Brazdil, R. Camacho, J. Gama (Eds.), Knowledge Discovery in Databases: PKDD 2005. XXIII, 719 pages. 2005.

Vol. 3720: J. Gama, R. Camacho, P.B. Brazdil, A. Jorge, L. Torgo (Eds.), Machine Learning: ECML 2005. XXIII, 769 pages. 2005.

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

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

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

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

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

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

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

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

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

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

Vol. 3658: V. Matoušek, P. Mautner, T. Pavelka (Eds.), Text, Speech and Dialogue. XV, 460 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. 3642: D. Ślęzak, 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.

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

Vol. 3635: J.R. Winkler, M. Niranjan, N.D. Lawrence (Eds.), Deterministic and Statistical Methods in Machine Learning. VIII, 341 pages. 2005.

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

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

Vol. 3626: B. Ganter, G. Stumme, R. Wille (Eds.), Formal Concept Analysis. X, 349 pages. 2005.

Vol. 3625: S. Kramer, B. Pfahringer (Eds.), Inductive Logic Programming. XIII, 427 pages. 2005.

Vol. 3620: H. Muñoz-Ávila, F. Ricci (Eds.), Case-Based Reasoning Research and Development. XV, 654 pages. 2005.

Vol. 3614: L. Wang, Y. Jin (Eds.), Fuzzy Systems and Knowledge Discovery, Part II. XLI, 1314 pages. 2005.

Vol. 3613: L. Wang, Y. Jin (Eds.), Fuzzy Systems and Knowledge Discovery, Part I. XLI, 1334 pages. 2005.

Vol. 3607: J.-D. Zucker, L. Saitta (Eds.), Abstraction, Reformulation and Approximation. XII, 376 pages. 2005.

Vol. 3601: G. Moro, S. Bergamaschi, K. Aberer (Eds.), Agents and Peer-to-Peer Computing. XII, 245 pages. 2005.

Vol. 3596: F. Dau, M.-L. Mugnier, G. Stumme (Eds.), Conceptual Structures: Common Semantics for Sharing Knowledge. XI, 467 pages. 2005.

Vol. 3593: V. Mařík, R. W. Brennan, M. Pěchouček (Eds.), Holonic and Multi-Agent Systems for Manufacturing. XI, 269 pages. 2005.

Vol. 3587: P. Perner, A. Imiya (Eds.), Machine Learning and Data Mining in Pattern Recognition. XVII, 695 pages. 2005.

Vol. 3584: X. Li, S. Wang, Z.Y. Dong (Eds.), Advanced Data Mining and Applications. XIX, 835 pages. 2005.

Vol. 3581: S. Miksch, J. Hunter, E.T. Keravnou (Eds.), Artificial Intelligence in Medicine. XVII, 547 pages. 2005.

Vol. 3577: R. Falcone, S. Barber, J. Sabater-Mir, M.P. Singh (Eds.), Trusting Agents for Trusting Electronic Societies. VIII, 235 pages. 2005.

Vol. 3575: S. Wermter, G. Palm, M. Elshaw (Eds.), Biomimetic Neural Learning for Intelligent Robots. IX, 383 pages. 2005.

Vol. 3571: L. Godo (Ed.), Symbolic and Quantitative Approaches to Reasoning with Uncertainty. XVI, 1028 pages. 2005.

Vol. 3559: P. Auer, R. Meir (Eds.), Learning Theory. XI, 692 pages. 2005.

Vol. 3558: V. Torra, Y. Narukawa, S. Miyamoto (Eds.), Modeling Decisions for Artificial Intelligence. XII, 470 pages. 2005.

Vol. 3554: A.K. Dey, B. Kokinov, D.B. Leake, R. Turner (Eds.), Modeling and Using Context. XIV, 572 pages. 2005.

Vol. 3550: T. Eymann, F. Klügl, W. Lamersdorf, M. Klusch, M.N. Huhns (Eds.), Multiagent System Technologies. XI, 246 pages. 2005.

Vol. 3539: K. Morik, J.-F. Boulicaut, A. Siebes (Eds.), Local Pattern Detection. XI, 233 pages. 2005.

Vol. 3538: L. Ardissono, P. Brna, A. Mitrović (Eds.), User Modeling 2005. XVI, 533 pages. 2005.

Vol. 3533: M. Ali, F. Esposito (Eds.), Innovations in Applied Artificial Intelligence. XX, 858 pages. 2005.

Vol. 3528: P.S. Szczepaniak, J. Kacprzyk, A. Niewiadomski (Eds.), Advances in Web Intelligence. XVII, 513 pages. 2005.

Vol. 3518: T.-B. Ho, D. Cheung, H. Liu (Eds.), Advances in Knowledge Discovery and Data Mining. XXI, 864 pages. 2005.

Vol. 3508: P. Bresciani, P. Giorgini, B. Henderson-Sellers, G. Low, M. Winikoff (Eds.), Agent-Oriented Information Systems II. X, 227 pages. 2005.

Vol. 3505: V. Gorodetsky, J. Liu, V.A. Skormin (Eds.), Autonomous Intelligent Systems: Agents and Data Mining. XIII, 303 pages. 2005.

Vol. 3501: B. Kégl, G. Lapalme (Eds.), Advances in Artificial Intelligence. XV, 458 pages. 2005.

Vol. 3492: P. Blache, E.P. Stabler, J.V. Busquets, R. Moot (Eds.), Logical Aspects of Computational Linguistics. X, 363 pages. 2005.

Vol. 3490: L. Bolc, Z. Michalewicz, T. Nishida (Eds.), Intelligent Media Technology for Communicative Intelligence. X, 259 pages. 2005.

Vol. 3488: M.-S. Hacid, N.V. Murray, Z.W. Raś, S. Tsumoto (Eds.), Foundations of Intelligent Systems. XIII, 700 pages. 2005.

Vol. 3487: J.A. Leite, P. Torroni (Eds.), Computational Logic in Multi-Agent Systems. XII, 281 pages. 2005.

Vol. 3476: J.A. Leite, A. Omicini, P. Torroni, P. Yolum (Eds.), Declarative Agent Languages and Technologies II. XII, 289 pages. 2005.

Vol. 3464: S.A. Brueckner, G.D.M. Serugendo, A. Karageorgos, R. Nagpal (Eds.), Engineering Self-Organising Systems. XIII, 299 pages. 2005.

Vol. 3452: F. Baader, A. Voronkov (Eds.), Logic for Programming, Artificial Intelligence, and Reasoning. XI, 562 pages. 2005.

Vol. 3451: M.-P. Gleizes, A. Omicini, F. Zambonelli (Eds.), Engineering Societies in the Agents World V. XIII, 349 pages. 2005.

Vol. 3446: T. Ishida, L. Gasser, H. Nakashima (Eds.), Massively Multi-Agent Systems I. XI, 349 pages. 2005.

Vol. 3445: G. Chollet, A. Esposito, M. Faúndez-Zanuy, M. Marinaro (Eds.), Nonlinear Speech Modeling and Applications. XIII, 433 pages. 2005.

Vol. 3438: H. Christiansen, P.R. Skadhauge, J. Villadsen (Eds.), Constraint Solving and Language Processing. VIII, 205 pages. 2005.

Vol. 3430: S. Tsumoto, T. Yamaguchi, M. Numao, H. Motoda (Eds.), Active Mining. XII, 349 pages. 2005.

Vol. 3419: B.V. Faltings, A. Petcu, F. Fages, F. Rossi (Eds.), Recent Advances in Constraints. X, 217 pages. 2005.

Vol. 3416: M.H. Böhlen, J. Gamper, W. Polasek, M.A. Wimmer (Eds.), E-Government: Towards Electronic Democracy. XIII, 311 pages. 2005.

Vol. 3415: P. Davidsson, B. Logan, K. Takadama (Eds.), Multi-Agent and Multi-Agent-Based Simulation. X, 265 pages. 2005.

Vol. 3403: B. Ganter, R. Godin (Eds.), Formal Concept Analysis. XI, 419 pages. 2005.

Vol. 3398: D.-K. Baik (Ed.), Systems Modeling and Simulation: Theory and Applications. XIV, 733 pages. 2005.

Vol. 3397: T.G. Kim (Ed.), Artificial Intelligence and Simulation. XV, 711 pages. 2005.

Vol. 3396: R.M. van Eijk, M.-P. Huget, F.P. M. Dignum (Eds.), Agent Communication. X, 261 pages. 2005.

Vol. 3394: D. Kudenko, D. Kazakov, E. Alonso (Eds.), Adaptive Agents and Multi-Agent Systems II. VIII, 313 pages. 2005.

Vol. 3392: D. Seipel, M. Hanus, U. Geske, O. Bartenstein (Eds.), Applications of Declarative Programming and Knowledge Management. X, 309 pages. 2005.

Vol. 3374: D. Weyns, H. V.D. Parunak, F. Michel (Eds.), Environments for Multi-Agent Systems. X, 279 pages. 2005.

Vol. 3371: M.W. Barley, N. Kasabov (Eds.), Intelligent Agents and Multi-Agent Systems. X, 329 pages. 2005.

Vol. 3369: V. R. Benjamins, P. Casanovas, J. Breuker, A. Gangemi (Eds.), Law and the Semantic Web. XII, 249 pages. 2005.

Vol. 3366: I. Rahwan, P. Moraïtis, C. Reed (Eds.), Argumentation in Multi-Agent Systems. XII, 263 pages. 2005.

Vol. 3359: G. Grieser, Y. Tanaka (Eds.), Intuitive Human Interfaces for Organizing and Accessing Intellectual Assets. XIV, 257 pages. 2005.

Vol. 3346: R.H. Bordini, M. Dastani, J. Dix, A.E.F. Seghrouchni (Eds.), Programming Multi-Agent Systems. XIV, 249 pages. 2005.

Vol. 3345: Y. Cai (Ed.), Ambient Intelligence for Scientific Discovery. XII, 311 pages. 2005.

Vol. 3343: C. Freksa, M. Knauff, B. Krieg-Brückner, B. Nebel, T. Barkowsky (Eds.), Spatial Cognition IV. XIII, 519 pages. 2005.

Vol. 3339: G.I. Webb, X. Yu (Eds.), AI 2004: Advances in Artificial Intelligence. XXII, 1272 pages. 2004.

¥792.96元

Table of Contents

Independently Checkable Proofs from Decision Procedures: Issues and Progress

Allen Van Gelder

School of Engineering, University of California at Santa Cruz
avg@cs.ucsc.edu

Abstract. In many verification applications the desired outcome is that the formula is unsatisfiable: a satisfying assignment essentially exhibits a bug and unsatisfiability implies a lack of bugs, at least for the property being verified. Most current high-performance satisfiability solvers and special-theory decision procedures are unable to provide proof of unsatisfiability. Since bugs have been discovered in many such programs long after being put into service, an uncheckable decision poses a significant problem if important economic or safety decisions are to be based upon it. This talk develops the thesis is that decision procedures can and should be designed with the ability to output an independently checkable proof. While *finding* a proof is hard, *checking* a proof can be straightforward if the proof system is simple enough. (By a "proof" we mean a real proof, with no steps omitted.) In practice, most underlying theories can produce a resolution proof. We argue that outputting such a proof does not place an undue burden on the decision procedures. We report on practical progress in this area for satisfiability solvers. Experiments have been carried out with what might be the first implementations of solver and proof checker that were developed completely independently, having only the specifications of the proof-file format as common knowledge. There is a trend toward combining high-performance satisfiability solvers with other theorem-proving methods. As the total systems become more complex, the need for "independent audits" becomes greater. Design goals for checkable proofs are proposed.

G. Sutcliffe and A. Voronkov (Eds.): LPAR 2005, LNAI 3835, p. 1, 2005.

Zap: Automated Theorem Proving for Software Analysis

Thomas Ball, Shuvendu K. Lahiri, and Madanlal Musuvathi

Microsoft Research
{tball, shuvendu, madanm}@microsoft.com

Abstract. Automated theorem provers (ATPs) are a key component that many software verification and program analysis tools rely on. However, the basic interface provided by ATPs (validity/satisfiability checking of formulas) has changed little over the years. We believe that program analysis clients would benefit greatly if ATPs were to provide a richer set of operations. We describe our desiderata for such an interface to an ATP, the logics (theories) that an ATP for program analysis should support, and present how we have incorporated many of these ideas in Zap, an ATP built at Microsoft Research.

1 Introduction

To make statements about programs in the absence of concrete inputs requires some form of *symbolic reasoning*. For example, suppose we want to prove that the execution of the assignment statement `x:=x+1` from a state in which the formula $(x < 5)$ holds yields a state in which the formula $(x < 10)$ holds. To do so, we need machinery for manipulating and reasoning about formulas that represent sets of program states.

Automated theorem provers (ATPs) provide the machinery that enables such reasoning. Many questions about program behavior can be reduced to questions of the validity or satisfiability of a first-order formula, such as $\forall x : (x < 6) \implies (x < 10)$. For example, given a program P and a specification S, a verification condition $VC(P, S)$ is a formula that is valid if and only if program P satisfies specification S. The validity of $VC(P, S)$ can be determined using an ATP. The basic interface an ATP provides takes as input a formula and returns a Boolean ("Valid", "Invalid") answer. Of course, since the validity problem is undecidable for many logics, an ATP may return "Invalid" for a valid formula.

In addition to this basic interface, ATPs may generate proofs witnessing the validity of input formulas. This basic capability is essential to techniques such as proof-carrying code [Nec97], where the ATP is an untrusted and potentially complicated program and the proof generated by the ATP can be checked efficiently by a simple program.

Through our experience with the use of ATPs in program analysis clients, we often want ATPs to provide a richer interface so as to better support program analysis tasks. We group these tasks into four categories:

G. Sutcliffe and A. Voronkov (Eds.): LPAR 2005, LNAI 3835, pp. 2–22, 2005.

- **Symbolic Fixpoint Computation.** For propositional (Boolean) formulas, binary decision diagrams (BDDs) [Bry86] enable the computation of fixpoints necessary for symbolic reachability and symbolic CTL model checking [BCM+92] of finite state systems. The transition relation of a finite state system can be represented using a BDD, as well as the initial and reachable states of the system. A main advantage of BDDs is that every Boolean function has a normal form, which makes various operations efficient. The basic operations necessary for fixpoint computation are a subsumption test (to test for convergence), quantifier elimination (to eliminate temporary variables used in image computation) and a join operation (to combine formulas representing different sets of states; this is simply disjunction in the case of Boolean logic). We would like to lift these operations to logics that are more expressive than propositional logic, so as to enable the computation of symbolic fixpoints over structures that more closely correspond to the types in programming languages (integers, enumerations, pointers, etc.). While normal forms may not be achievable, simplification of formula is highly desirable to keep formulas small and increase the efficiency of the fixpoint computation.
- **Abstract Transformers.** A fundamental concept in analyzing infinite-state systems (such as programs) is that of abstraction. Often, a system may be converted to a simpler abstract form where certain questions are decidable, such that proofs in the abstract system carry over to proofs in the original system. Abstract interpretation is a framework for mathematically describing program abstractions and their meaning [CC77]. A basic step in the process is the creation of *abstract transformers*: each statement in the original program must be translated to a corresponding abstract statement. This step often is manual. Predicate abstraction is a means for automating the construction of finite-state abstract transformers from infinite-state systems using an ATP [GS97]. ATPs can also be used to create *symbolic best transformers* for other abstract domains [YRS04]. Unfortunately, these approaches suffer from having to make an exponential number of calls to the ATP. If an ATP provides an interface to find all the *consequences* of a set of facts, the process of predicate abstraction and creation of symbolic best transformers can be made more efficient [LBC05]. Consequence finding [Mar99] is a basic operation for the automated creation of abstract transformers that ATPs could support.
- **Property-guided Abstraction Refinement.** If an abstraction is not precise enough to establish the correctness of a program with respect to some property, we wish to find a way to make the abstraction more precise with respect to the property of interest [Kur94, CGJ+00, BR01]. Recently, McMillan showed how interpolants naturally describe how to refine (predicate) abstractions with respect to a property of interest [McM03, HJMM04]. An interpolating ATP [McM04] can support the automated refinement of abstractions.
- **Test Generation.** Finally, we would like to use ATPs to prove the presence of a bug to the user through the automated generation of failure-inducing