

Miki Hermann
Andrei Voronkov (Eds.)

LNAI 4246

Logic for Programming, Artificial Intelligence, and Reasoning

13th International Conference, LPAR 2006
Phnom Penh, Cambodia, November 2006
Proceedings



Springer

Miki Hermann Andrei Voronkov (Eds.)

Logic for Programming, Artificial Intelligence, and Reasoning

13th International Conference, LPAR 2006
Phnom Penh, Cambodia, November 13-17, 2006
Proceedings

Series Editors

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

Volume Editors

Miki Hermann
LIX, École Polytechnique
91128 Palaiseau, France
E-mail: hermann@lix.polytechnique.fr

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

Library of Congress Control Number: 2006934988

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

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-540-48281-4 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-48281-9 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springer.com

© Springer-Verlag Berlin Heidelberg 2006
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11916277 06/3142 5 4 3 2 1 0

Lecture Notes in Artificial Intelligence 4246

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Preface

This volume contains the papers presented at the 13th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR), held during November 13–17, 2006, in Phnom Penh, Cambodia, together with the 2nd International Workshop on Analytic Proof Systems (organized by Christian Fermüller and Matthias Baaz) and the 6th International Workshop on the Implementation of Logics (organized by Christoph Benzmüller, Bernd Fischer, and Geoff Sutcliffe).

The call for papers attracted 96 paper submissions, each of which was reviewed by at least three expert reviewers. The final decisions on the papers were taken during an electronic Program Committee meeting held on the Internet. The Internet-based submission, reviewing, and discussion software EasyChair, provided by the second PC Co-chair, supported each stage of the reviewing process. Nevertheless, the most important work was, of course, done by the 35 PC members and their external reviewers, who provided high-quality reviews. After intense discussions to resolve conflicts among the reviewers, the Program Committee decided to accept 38 papers.

The conference program also included an invited talk by Jean-Pierre Jouanaud, documented by an additional paper in these proceedings.

Apart from the authors, invited speaker, Program Committee members, and external reviewers, we would like to thank other people and organizations that made this LPAR conference a success: the Local Organization Chair, Sopheap Seng, and all the other people involved in the local organization; the Department of Information and Communication Technology Engineering at the Institut de Technologie du Cambodge in Phnom Penh, and the Kurt Gödel Society, which provided partial funding and support.

September 2006

Miki Hermann
Andrei Voronkov

Conference Organization

Program Chairs

Miki Hermann
Andrei Voronkov

Workshop Chair

Christelle Scharff

Program Committee

María Alpuente	Pascal van Hentenryck	Michel Parigot
Franz Baader	Brahim Hnich	Frank Pfenning
Matthias Baaz	Ian Horrocks	Reinhard Pichler
Christoph Benz Müller	Viktor Kuncak	Michael Rusinowitch
Koen Claessen	Orna Kupferman	Mooly Sagiv
Javier Esparza	Christopher Lynch	Gernot Salzer
Bernd Fischer	Dale Miller	Christelle Scharff
Jürgen Giesl	George Necula	Sopheap Seng
Jean Goubault-Larrecq	Joachim Niehren	Geoff Sutcliffe
Erich Grädel	Luke Ong	Sophie Tison
Ziyad Hanna	Catuscia Palamidessi	Margus Veanes

Local Organization

Uddam Chukmol	Sopheap Seng (Chair)	Touch Sereysethy
Seila Nuon		Kret Try

Supporting Institutions

Institut de Technologie du Cambodge, Phnom Penh, Cambodia

External Reviewers

Andreas Abel	Alexander Fuchs	Christian Michaux
Francis Alexandre	Marco Gavanelli	Alexandre Miquel
Torsten Altenkirch	Martin Giese	Barbara Morawska
Demis Ballis	Bernhard Gramlich	Ginés Moreno
Mike Barnett	Wolfgang Grieskamp	Boris Motik
Gregory Batt	Yuri Gurevich	Huu Hai Nguyen
Peter Baumgartner	Volker Haarslev	Robert Nieuwenhuis
Christoph Beierle	Peter Habermehl	Andreas Nonnengart
Sergey Berezin	Matthew Hague	Dirk Nowotka
Philippe Besnard	Dieter Hofbauer	Nicola Olivetti
Nikolaj Bjørner	Steffen Hölldobler	Bijan Parsia
Bruno Blanchet	Rosalie Iemhoff	Vicente Pelechano
Frédéric Blanqui	Radu Iosif	Enrico Pontelli
Hendrik Blockeel	Florent Jacquemard	Michael Poppleton
Bernard Boigelot	Gerda Janssens	Florian Rabe
Charles Bouillaguet	Yevgeny Kazakov	M. J. Ramírez-Quintana
Olivier Bournez	Michael Kohlhase	Silvio Ranise
Gerhard Brewka	Sebastien Konieczny	Christophe Ringeissen
Chad Brown	Adam Koprowski	Pritam Roy
Kai Brünner	Robert Kowalski	Paul Rozière
Maurice Bruynooghe	Ralf Kuesters	Eric Rutten
Elie Bursztein	Temur Kutsia	Kostis Sagonas
Michael Butler	Oliver Kutz	Sam Sanjabi
Diego Calvanese	Barbara König	Peter Schneider-Kamp
Colin Campbell	Gerardo Lafferriere	Danny De Schreye
Domenico Cantone	Ugo Dal Lago	Yann Secq
Felix Chang	Shuvendu Lahiri	Lionel Seinturier
Henning Christiansen	Alexander Leitsch	Jiří Srba
Agata Ciabattoni	Christopher Lepock	Colin Stirling
Marco Comini	Tal Lev-Ami	Lutz Straßburger
Álvaro Cortés	Francesco Logozzo	Sorin Stratulat
Jim Cunningham	Markus Lohrey	Niklas Sörensson
Mehdi Dastani	Salvador Lucas	Jean-Marc Talbot
Giorgio Delzanno	Sebastian Maneth	Sebastiaan Terwijn
Stéphane Demri	Claude Marché	Frank Theiss
Phan Minh Dung	Joao Marques-Silva	René Thiemann
Esra Erdem	Pierre Marquis	Nikolai Tillmann
Santiago Escobar	Viviana Mascardi	Alwen Tiu
Wolfgang Faber	Ralph Matthes	Salvatore La Torre
Ansgar Fehnker	Nancy Mazur	Josef Urban
Christian Fermüller	Stephen McCamant	Frank Valencia
Michael Fink	Erik Meijer	Jean-Stéphane Varré
Anders Franzén	George Metcalfe	Bjorn Victor

Alicia Villanueva
David Wahlstedt
Fang Wei
Benjamin Werner
Thomas Wies

Stefan Woltran
James Worrell
Hongwei Xi
Avi Yadgar
Greta Yorsh

Karen Zee
Juergen Zimmer
Evgeny Zolin

Conferences Preceding LPAR-13

RCLP, 1990, Irkutsk, Soviet Union
RCLP 1991, Leningrad, Soviet Union, aboard the ship “Michail Lomonosov”
LPAR 1992, St. Petersburg, Russia, aboard the ship “Michail Lomonosov”
LPAR 1993, St. Petersburg, Russia
LPAR 1994, Kiev, Ukraine, aboard the ship “Marshal Koshevoi”
LPAR 1999, Tbilisi, Republic of Georgia
LPAR 2000, Réunion Island, France
LPAR 2001, Havana, Cuba
LPAR 2002, Tbilisi, Republic of Georgia
LPAR 2003, Almaty, Kazakhstan
LPAR 2004, Montevideo, Uruguay
LPAR 2005, Montego Bay, Jamaica

Lecture Notes in Artificial Intelligence (LNAI)

- Vol. 4265: N. Lavrač, L. Todorovski, K.P. Jantke (Eds.), *Discovery Science*. XIV, 384 pages. 2006.
- Vol. 4264: J.L. Balcázar, P.M. Long, F. Stephan (Eds.), *Algorithmic Learning Theory*. XIII, 393 pages. 2006.
- Vol. 4253: B. Gabrys, R.J. Howlett, L.C. Jain (Eds.), *Knowledge-Based Intelligent Information and Engineering Systems, Part III*. XXXII, 1301 pages. 2006.
- Vol. 4252: B. Gabrys, R.J. Howlett, L.C. Jain (Eds.), *Knowledge-Based Intelligent Information and Engineering Systems, Part II*. XXXIII, 1335 pages. 2006.
- Vol. 4251: B. Gabrys, R.J. Howlett, L.C. Jain (Eds.), *Knowledge-Based Intelligent Information and Engineering Systems, Part I*. LXVI, 1297 pages. 2006.
- Vol. 4248: S. Staab, V. Svátek (Eds.), *Engineering Knowledge in the Age of the Semantic Web*. XIV, 400 pages. 2006.
- Vol. 4246: M. Hermann, A. Voronkov (Eds.), *Logic for Programming, Artificial Intelligence, and Reasoning*. XIII, 588 pages. 2006.
- Vol. 4223: L. Wang, L. Jiao, G. Shi, X. Li, J. Liu (Eds.), *Fuzzy Systems and Knowledge Discovery*. XXVIII, 1335 pages. 2006.
- Vol. 4213: J. Fürnkranz, T. Scheffer, M. Spiliopoulou (Eds.), *Knowledge Discovery in Databases: PKDD 2006*. XXII, 660 pages. 2006.
- Vol. 4212: J. Fürnkranz, T. Scheffer, M. Spiliopoulou (Eds.), *Machine Learning: ECML 2006*. XXIII, 851 pages. 2006.
- Vol. 4211: P. Vogt, Y. Sugita, E. Tuci, C. Nehaniv (Eds.), *Symbol Grounding and Beyond*. VIII, 237 pages. 2006.
- Vol. 4203: F. Esposito, Z.W. Raś, D. Malerba, G. Semeraro (Eds.), *Foundations of Intelligent Systems*. XVIII, 767 pages. 2006.
- Vol. 4201: Y. Sakakibara, S. Kobayashi, K. Sato, T. Nishino, E. Tomita (Eds.), *Grammatical Inference: Algorithms and Applications*. XII, 359 pages. 2006.
- Vol. 4200: I.F.C. Smith (Ed.), *Intelligent Computing in Engineering and Architecture*. XIII, 692 pages. 2006.
- Vol. 4198: O. Nasraoui, O. Zaiane, M. Spiliopoulou, B. Mobasher, B. Masand, P. Yu (Eds.), *Advances in Web Mining and Web Usage Analysis*. IX, 177 pages. 2006.
- Vol. 4196: K. Fischer, I.J. Timm, E. André, N. Zhong (Eds.), *Multiagent System Technologies*. X, 185 pages. 2006.
- Vol. 4188: P. Sojka, I. Kopeček, K. Pala (Eds.), *Text, Speech and Dialogue*. XV, 721 pages. 2006.
- Vol. 4183: J. Euzenat, J. Domingue (Eds.), *Artificial Intelligence: Methodology, Systems, and Applications*. XIII, 291 pages. 2006.
- Vol. 4180: M. Kohlhase, *OMDoc – An Open Markup Format for Mathematical Documents [version 1.2]*. XIX, 428 pages. 2006.
- Vol. 4177: R. Marín, E. Onaindía, A. Bugarín, J. Santos (Eds.), *Current Topics in Artificial Intelligence*. XV, 482 pages. 2006.
- Vol. 4160: M. Fisher, W.v.d. Hoek, B. Konev, A. Lisitsa (Eds.), *Logics in Artificial Intelligence*. XII, 516 pages. 2006.
- Vol. 4155: O. Stock, M. Schaerf (Eds.), *Reasoning, Action and Interaction in AI Theories and Systems*. XVIII, 343 pages. 2006.
- Vol. 4149: M. Klusch, M. Rovatsos, T.R. Payne (Eds.), *Cooperative Information Agents X*. XII, 477 pages. 2006.
- Vol. 4140: J.S. Sichman, H. Coelho, S.O. Rezende (Eds.), *Advances in Artificial Intelligence - IBERAMIA-SBIA 2006*. XXIII, 635 pages. 2006.
- Vol. 4139: T. Salakoski, F. Ginter, S. Pyysalo, T. Pahikkala, *Advances in Natural Language Processing*. XVI, 771 pages. 2006.
- Vol. 4133: J. Gratch, M. Young, R. Aylett, D. Ballin, P. Olivier (Eds.), *Intelligent Virtual Agents*. XIV, 472 pages. 2006.
- Vol. 4130: U. Furbach, N. Shankar (Eds.), *Automated Reasoning*. XV, 680 pages. 2006.
- Vol. 4120: J. Calmet, T. Ida, D. Wang (Eds.), *Artificial Intelligence and Symbolic Computation*. XIII, 269 pages. 2006.
- Vol. 4114: D.-S. Huang, K. Li, G.W. Irwin (Eds.), *Computational Intelligence, Part II*. XXVII, 1337 pages. 2006.
- Vol. 4108: J.M. Borwein, W.M. Farmer (Eds.), *Mathematical Knowledge Management*. VIII, 295 pages. 2006.
- Vol. 4106: T.R. Roth-Berghofer, M.H. Göker, H.A. Güvenir (Eds.), *Advances in Case-Based Reasoning*. XIV, 566 pages. 2006.
- Vol. 4099: Q. Yang, G. Webb (Eds.), *PRICAI 2006: Trends in Artificial Intelligence*. XXVIII, 1263 pages. 2006.
- Vol. 4095: S. Nolfi, G. Baldassarre, R. Calabretta, J.C.T. Hallam, D. Marocco, J.-A. Meyer, O. Miglino, D. Parisi (Eds.), *From Animals to Animats 9*. XV, 869 pages. 2006.
- Vol. 4093: X. Li, O.R. Zaiane, Z. Li (Eds.), *Advanced Data Mining and Applications*. XXI, 1110 pages. 2006.
- Vol. 4092: J. Lang, F. Lin, J. Wang (Eds.), *Knowledge Science, Engineering and Management*. XV, 664 pages. 2006.

- Vol. 4088: Z.-Z. Shi, R. Sadananda (Eds.), *Agent Computing and Multi-Agent Systems*. XVII, 827 pages. 2006.
- Vol. 4087: F. Schwenker, S. Marinai (Eds.), *Artificial Neural Networks in Pattern Recognition*. IX, 299 pages. 2006.
- Vol. 4068: H. Schärfe, P. Hitzler, P. Øhrstrøm (Eds.), *Conceptual Structures: Inspiration and Application*. XI, 455 pages. 2006.
- Vol. 4065: P. Perner (Ed.), *Advances in Data Mining*. XI, 592 pages. 2006.
- Vol. 4062: G. Wang, J.F. Peters, A. Skowron, Y. Yao (Eds.), *Rough Sets and Knowledge Technology*. XX, 810 pages. 2006.
- Vol. 4049: S. Parsons, N. Maudet, P. Moraitis, I. Rahwan (Eds.), *Argumentation in Multi-Agent Systems*. XIV, 313 pages. 2006.
- Vol. 4048: L. Goble, J.-J.C. Meyer (Eds.), *Deontic Logic and Artificial Normative Systems*. X, 273 pages. 2006.
- Vol. 4045: D. Barker-Plummer, R. Cox, N. Swoboda (Eds.), *Diagrammatic Representation and Inference*. XII, 301 pages. 2006.
- Vol. 4031: M. Ali, R. Dapoigny (Eds.), *Advances in Applied Artificial Intelligence*. XXIII, 1353 pages. 2006.
- Vol. 4029: L. Rutkowski, R. Tadeusiewicz, L.A. Zadeh, J.M. Zurada (Eds.), *Artificial Intelligence and Soft Computing – ICAISC 2006*. XXI, 1235 pages. 2006.
- Vol. 4027: H.L. Larsen, G. Pasi, D. Ortiz-Arroyo, T. Andreassen, H. Christiansen (Eds.), *Flexible Query Answering Systems*. XVIII, 714 pages. 2006.
- Vol. 4021: E. André, L. Dybkjaer, W. Minker, H. Neumann, M. Weber (Eds.), *Perception and Interactive Technologies*. XI, 217 pages. 2006.
- Vol. 4020: A. Bredenfeld, A. Jacoff, I. Noda, Y. Takahashi (Eds.), *RoboCup 2005: Robot Soccer World Cup IX*. XVII, 727 pages. 2006.
- Vol. 4013: L. Lamontagne, M. Marchand (Eds.), *Advances in Artificial Intelligence*. XIII, 564 pages. 2006.
- Vol. 4012: T. Washio, A. Sakurai, K. Nakajima, H. Takeda, S. Tojo, M. Yokoo (Eds.), *New Frontiers in Artificial Intelligence*. XIII, 484 pages. 2006.
- Vol. 4008: J.C. Augusto, C.D. Nugent (Eds.), *Designing Smart Homes*. XI, 183 pages. 2006.
- Vol. 4005: G. Lugosi, H.U. Simon (Eds.), *Learning Theory*. XI, 656 pages. 2006.
- Vol. 3978: B. Hnich, M. Carlsson, F. Fages, F. Rossi (Eds.), *Recent Advances in Constraints*. VIII, 179 pages. 2006.
- Vol. 3963: O. Dikenelli, M.-P. Gleizes, A. Ricci (Eds.), *Engineering Societies in the Agents World VI*. XII, 303 pages. 2006.
- Vol. 3960: R. Vieira, P. Quaresma, M.d.G.V. Nunes, N.J. Mamede, C. Oliveira, M.C. Dias (Eds.), *Computational Processing of the Portuguese Language*. XII, 274 pages. 2006.
- Vol. 3955: G. Antoniou, G. Potamias, C. Spyropoulos, D. Plexousakis (Eds.), *Advances in Artificial Intelligence*. XVII, 611 pages. 2006.
- Vol. 3949: F.A. Savacı (Ed.), *Artificial Intelligence and Neural Networks*. IX, 227 pages. 2006.
- Vol. 3946: T.R. Roth-Berghofer, S. Schulz, D.B. Leake (Eds.), *Modeling and Retrieval of Context*. XI, 149 pages. 2006.
- Vol. 3944: J. Quiñonero-Candela, I. Dagan, B. Magnini, F. d'Alché-Buc (Eds.), *Machine Learning Challenges*. XIII, 462 pages. 2006.
- Vol. 3937: H. La Poutré, N.M. Sadeh, S. Janson (Eds.), *Agent-Mediated Electronic Commerce*. X, 227 pages. 2006.
- Vol. 3932: B. Mobasher, O. Nasraoui, B. Liu, B. Masand (Eds.), *Advances in Web Mining and Web Usage Analysis*. X, 189 pages. 2006.
- Vol. 3930: D.S. Yeung, Z.-Q. Liu, X.-Z. Wang, H. Yan (Eds.), *Advances in Machine Learning and Cybernetics*. XXI, 1110 pages. 2006.
- Vol. 3918: W.-K. Ng, M. Kitsuregawa, J. Li, K. Chang (Eds.), *Advances in Knowledge Discovery and Data Mining*. XXIV, 879 pages. 2006.
- Vol. 3913: O. Boissier, J. Padget, V. Dignum, G. Lindemann, E. Matson, S. Ossowski, J.S. Sichman, J. Vázquez-Salceda (Eds.), *Coordination, Organizations, Institutions, and Norms in Multi-Agent Systems*. XII, 259 pages. 2006.
- Vol. 3910: S.A. Brueckner, G.D.M. Serugendo, D. Hales, F. Zambonelli (Eds.), *Engineering Self-Organising Systems*. XII, 245 pages. 2006.
- Vol. 3904: M. Baldoni, U. Endriss, A. Omicini, P. Torroni (Eds.), *Declarative Agent Languages and Technologies III*. XII, 245 pages. 2006.
- Vol. 3900: F. Toni, P. Torroni (Eds.), *Computational Logic in Multi-Agent Systems*. XVII, 427 pages. 2006.
- Vol. 3899: S. Frintrop, VOCUS: A Visual Attention System for Object Detection and Goal-Directed Search. XIV, 216 pages. 2006.
- Vol. 3898: K. Tuyls, P.J. 't Hoen, K. Verbeeck, S. Sen (Eds.), *Learning and Adaption in Multi-Agent Systems*. X, 217 pages. 2006.
- Vol. 3891: J.S. Sichman, L. Antunes (Eds.), *Multi-Agent-Based Simulation VI*. X, 191 pages. 2006.
- Vol. 3890: S.G. Thompson, R. Ghanea-Hercock (Eds.), *Defence Applications of Multi-Agent Systems*. XII, 141 pages. 2006.
- Vol. 3885: V. Torra, Y. Narukawa, A. Valls, J. Domingo-Ferrer (Eds.), *Modeling Decisions for Artificial Intelligence*. XII, 374 pages. 2006.
- Vol. 3881: S. Gibet, N. Courty, J.-F. Kamp (Eds.), *Gesture in Human-Computer Interaction and Simulation*. XIII, 344 pages. 2006.
- Vol. 3874: R. Missaoui, J. Schmidt (Eds.), *Formal Concept Analysis*. X, 309 pages. 2006.
- Vol. 3873: L. Maicher, J. Park (Eds.), *Charting the Topic Maps Research and Applications Landscape*. VIII, 281 pages. 2006.
- Vol. 3864: Y. Cai, J. Abascal (Eds.), *Ambient Intelligence in Everyday Life*. XII, 323 pages. 2006.
- Vol. 3863: M. Kohlhasse (Ed.), *Mathematical Knowledge Management*. XI, 405 pages. 2006.

Table of Contents

Higher-Order Termination: From Kruskal to Computability	1
<i>Frédéric Blanqui, Jean-Pierre Jouannaud, Albert Rubio</i>	
Deciding Satisfiability of Positive Second Order Joinability Formulae	15
<i>Sébastien Limet, Pierre Pillot</i>	
SAT Solving for Argument Filterings	30
<i>Michael Codish, Peter Schneider-Kamp, Vitaly Lagoon, René Thiemann, Jürgen Giesl</i>	
Inductive Decidability Using Implicit Induction	45
<i>Stephan Falke, Deepak Kapur</i>	
Matching Modulo Superdevelopments Application to Second-Order Matching	60
<i>Germain Faure</i>	
Derivational Complexity of Knuth-Bendix Orders Revisited	75
<i>Georg Moser</i>	
A Characterization of Alternating Log Time by First Order Functional Programs	90
<i>Guillaume Bonfante, Jean-Yves Marion, Romain Péchoux</i>	
Combining Typing and Size Constraints for Checking the Termination of Higher-Order Conditional Rewrite Systems	105
<i>Frédéric Blanqui, Colin Riba</i>	
On a Local-Step Cut-Elimination Procedure for the Intuitionistic Sequent Calculus	120
<i>Kentaro Kikuchi</i>	
Modular Cut-Elimination: Finding Proofs or Counterexamples	135
<i>Agata Ciabattoni, Kazushige Terui</i>	
An Executable Formalization of the HOL/Nuprl Connection in the Metalogical Framework Twelf	150
<i>Carsten Schürmann, Mark-Oliver Stehr</i>	
A Semantic Completeness Proof for TaMeD	167
<i>Richard Bonichon, Olivier Hermant</i>	

Saturation Up to Redundancy for Tableau and Sequent Calculi	182
<i>Martin Giese</i>	
Branching-Time Temporal Logic Extended with Qualitative Presburger Constraints	197
<i>Laura Bozzelli, Régis Gascon</i>	
Combining Supervaluation and Degree Based Reasoning Under Vagueness	212
<i>Christian G. Fermüller, Robert Kosik</i>	
A Comparison of Reasoning Techniques for Querying Large Description Logic ABoxes	227
<i>Boris Motik, Ulrike Sattler</i>	
A Local System for Intuitionistic Logic	242
<i>Alwen Tiu</i>	
CIC [∞] : Type-Based Termination of Recursive Definitions in the Calculus of Inductive Constructions	257
<i>Gilles Barthe, Benjamin Grégoire, Fernando Pastawski</i>	
Reducing Nondeterminism in the Calculus of Structures	272
<i>Ozan Kahramanoğlu</i>	
A Relaxed Approach to Integrity and Inconsistency in Databases	287
<i>Hendrik Decker, Davide Martinenghi</i>	
On Locally Checkable Properties	302
<i>Orna Kupferman, Yoad Lustig, Moshe Y. Vardi</i>	
Deciding Key Cycles for Security Protocols	317
<i>Véronique Cortier, Eugen Zălinescu</i>	
Automating Verification of Loops by Parallelization	332
<i>Tobias Gedell, Reiner Hähnle</i>	
On Computing Fixpoints in Well-Structured Regular Model Checking, with Applications to Lossy Channel Systems	347
<i>Christel Baier, Nathalie Bertrand, Philippe Schnoebelen</i>	
Verification Condition Generation Via Theorem Proving	362
<i>John Matthews, J. Strother Moore, Sandip Ray, Daron Vroon</i>	
An Incremental Approach to Abstraction-Carrying Code	377
<i>Elvira Albert, Puri Arenas, Germán Puebla</i>	
Context-Sensitive Multivariant Assertion Checking in Modular Programs	392
<i>Paweł Pietrzak, Jesús Correias, Germán Puebla, Manuel V. Hermenegildo</i>	

Representation of Partial Knowledge and Query Answering in Locally Complete Databases	407
<i>Álvaro Cortés-Calabuig, Marc Denecker, Ofer Arieli, Maurice Bruynooghe</i>	
Sequential, Parallel, and Quantified Updates of First-Order Structures . . .	422
<i>Philipp Rümmer</i>	
Representing Defaults and Negative Information Without Negation-as-Failure	437
<i>Pablo R. Fillottrani, Guillermo R. Simari</i>	
Constructing Camin-Sokal Phylogenies Via Answer Set Programming	452
<i>Jonathan Kavanagh, David Mitchell, Eugenia Ternovska, Ján Maňuch, Xiaohong Zhao, Arvind Gupta</i>	
Automata for Positive Core XPath Queries on Compressed Documents	467
<i>Barbara Fila, Siva Anantharaman</i>	
Boolean Rings for Intersection-Based Satisfiability	482
<i>Nachum Dershowitz, Jieh Hsiang, Guan-Shieng Huang, Daher Kaiss</i>	
Theory Instantiation	497
<i>Harald Ganzinger, Konstantin Korovin</i>	
Splitting on Demand in SAT Modulo Theories	512
<i>Clark Barrett, Robert Nieuwenhuis, Albert Oliveras, Cesare Tinelli</i>	
Delayed Theory Combination vs. Nelson-Oppen for Satisfiability Modulo Theories: A Comparative Analysis	527
<i>Roberto Bruttomesso, Alessandro Cimatti, Anders Franzén, Alberto Griggio, Roberto Sebastiani</i>	
Automatic Combinability of Rewriting-Based Satisfiability Procedures	542
<i>Hélène Kirchner, Silvio Ranise, Christophe Ringeissen, Duc-Khanh Tran</i>	
To Ackermann-ize or Not to Ackermann-ize? On Efficiently Handling Uninterpreted Function Symbols in $SMT(\mathcal{EUF} \cup \mathcal{T})$	557
<i>Roberto Bruttomesso, Alessandro Cimatti, Anders Franzén, Alberto Griggio, Alessandro Santuari, Roberto Sebastiani</i>	
Lemma Learning in the Model Evolution Calculus	572
<i>Peter Baumgartner, Alexander Fuchs, Cesare Tinelli</i>	
Author Index	587

Higher-Order Termination: From Kruskal to Computability

Frédéric Blanqui¹, Jean-Pierre Jouannaud^{2,*}, and Albert Rubio³

¹ INRIA & LORIA, BP 101, 54602 Villiers-lès-Nancy CEDEX, France

² LIX, École Polytechnique, 91400 Palaiseau, France

³ Technical University of Catalonia, Pau Gargallo 5, 08028 Barcelona, Spain

1 Introduction

Termination is a major question in both logic and computer science. In logic, termination is at the heart of proof theory where it is usually called strong normalization (of cut elimination). In computer science, termination has always been an important issue for showing programs correct. In the early days of logic, strong normalization was usually shown by assigning ordinals to expressions in such a way that eliminating a cut would yield an expression with a smaller ordinal. In the early days of verification, computer scientists used similar ideas, interpreting the arguments of a program call by a natural number, such as their size. Showing the size of the arguments to decrease for each recursive call gives a termination proof of the program, which is however rather weak since it can only yield quite small ordinals. In the sixties, Tait invented a new method for showing cut elimination of natural deduction, based on a *predicate* over the set of terms, such that the membership of an expression to the predicate implied the strong normalization property for that expression. The predicate being defined by induction on types, or even as a fixpoint, this method could yield much larger ordinals. Later generalized by Girard under the name of *reducibility* or *computability candidates*, it showed very effective in proving the strong normalization property of typed lambda-calculi with polymorphic types, dependent types, inductive types, and finally a cumulative hierarchy of universes. On the programming side, research on termination shifted from programming to executable specification languages based on rewriting, and concentrated on automatable methods based on the construction on well-founded orderings of the set of terms. The milestone here is Dershowitz's *recursive path ordering* (RPO), in the late seventies, whose well-foundedness proof is based on a powerful combinatorial argument, Kruskal's tree theorem, which also yields rather large ordinals. While the computability predicates must be defined for each particular case, and their properties proved by hand, the recursive path ordering can be effectively automated.

These two methods are completely different. Computability arguments show *termination*, that is, infinite decreasing sequences of expressions $e_0 \succ e_1 \succ \dots e_n \succ e_{n+1} \dots$ do not exist. Kruskal's based arguments show *well-orderedness*: for any infinite sequence of expressions $\{e_i\}_i$, there is a pair $j < k$ such that $e_j \preceq e_k$. It is easy to see that well-orderedness implies termination, but the converse is not true.

* Project LogiCal, Pôle Commun de Recherche en Informatique du Plateau de Saclay, CNRS, École Polytechnique, INRIA, Université Paris-Sud.

In the late eighties, a new question arose: termination of a simply-typed lambda-calculus language in which beta-reduction would be supplemented with terminating first-order rewrite rules. Breazu-Tannen and Gallier on the one hand [12], and Okada [23] on the other hand, showed that termination was satisfied by the combination by using computability arguments. Indeed, when rewriting operates at basic types and is generated by first-order rewrite rules, beta-reduction and rewriting do not interfere. Their result, proved for a polymorphic λ -calculus, was later generalized to the calculus of constructions [1]. The situation becomes radically different with higher-order rewriting generated by rules operating on arrow-types, or involving lambda-bindings or higher-order variables. Such an example is provided by Gödel's system T , in which higher-order primitive recursion for natural numbers generated by Peano's constructors 0 and s is described by the following two higher-order rules:

$$\begin{aligned} \text{rec}(0, U, V) &\rightarrow U \\ \text{rec}(s(X), U, V) &\rightarrow @(V, X, \text{rec}(X, U, V)) \end{aligned}$$

where rec is a function symbol of type $\mathbb{N} \rightarrow T \rightarrow (\mathbb{N} \rightarrow T \rightarrow T) \rightarrow T$, U is a higher-order variable of type T and V a higher-order variable of type $\mathbb{N} \rightarrow T \rightarrow T$, for all type T . Jouannaud and Okada invented the so-called general-schema [17], a powerful generalization of Gödel's higher-order primitive recursion of higher types. Following the path initiated by Breazu-Tannen and Gallier on the one hand, and Okada on the other hand, termination of calculi based on the general schema was proved by using computability arguments as well [17,18,2]. The general schema was then reformulated by Blanqui, Jouannaud and Okada [3,4] in order to incorporate computability arguments directly in its definition, opening the way to new generalizations. Gödel's system T can be generalized in two ways, by introducing type constructors and dependent types, yielding the Calculus of Constructions, and by introducing strictly positive inductive types. Both together yield the Calculus of Inductive Constructions [24], the theory underlying the Coq system [14], in which rewrite rules like strong elimination operate on types, raising new difficulties. Blanqui gave a generalization of the general schema which includes the Calculus of Inductive Constructions as a particular case under the name of Calculus of Algebraic Constructions [6,7].

The general schema, however, is too simple to analyze complex calculi defined by higher-order rewrite rules such as encodings of logics. For that purpose, Jouannaud and Rubio generalized the recursive path ordering to the higher-order case, yielding the higher-order recursive path ordering (HORPO) [19]. The RPO well-foundedness proof follows from Kruskal's tree theorem, but no such theorem exists in presence of a binding construct, and it is not at all clear that such a theorem may exist. What is remarkable is that computability arguments fit with RPO's recursive structure. When applied to RPO, these arguments result in a new, simple, well-foundedness proof of RPO. One could even argue that this is the *first* well-foundedness proof of RPO, since Dershowitz showed *more*: well-orderedness.

Combining the general schema and the HORPO is indeed easy because their termination properties are both based on computability arguments. The resulting relation, HORPO with closure, combines an ordering relation with a membership predicate. In this paper, we reformulate and improve a recent idea of Blanqui [9] by defining a new

version of the HORPO with closure which integrates smoothly the idea of the general schema into HORPO in the form of a new ordering definition.

So far, we have considered the kind of higher-order rewriting defined by using first-order pattern matching as in the calculus of constructions. These orderings need to contain β - and η -reductions. Showing termination of higher-order rewrite rules based on higher-order pattern matching, that is, rewriting modulo β and η now used as equalities, turns out to require simple modifications of HORPO [20]. We will therefore concentrate here on higher-order orderings containing β - and η -reductions.

We introduce higher-order algebras in Section 2. In Section 3, we recall the computability argument for this variation of the simply typed lambda calculus. Using a computability argument again, we show in Section 4 that RPO is well-founded. We introduce the general schema in section 5, and the HORPO in Section 6 before to combine both in Section 7. We end up with related work and open problems in the last two sections.

2 Higher-Order Algebras

The notion of a higher-order algebra given here is the monomorphic version of the notion of polymorphic higher-order algebra defined in [21]. Polymorphism has been ruled out for simplicity.

2.1 Types, Signatures and Terms

Given a set \mathcal{S} of *sort symbols* of a fixed arity, denoted by $s : *^n \Rightarrow *$, the set $\mathcal{T}_{\mathcal{S}}$ of *types* is generated from these sets by the arrow constructor:

$$\mathcal{T}_{\mathcal{S}} := s(\mathcal{T}_{\mathcal{S}}^n) \mid (\mathcal{T}_{\mathcal{S}} \rightarrow \mathcal{T}_{\mathcal{S}}) \\ \text{for } s : *^n \Rightarrow * \in \mathcal{S}$$

Types headed by \rightarrow are *arrow types* while the others are *basic types*. *Type declarations* are expressions of the form $\sigma_1 \times \cdots \times \sigma_n \rightarrow \sigma$, where n is the *arity* of the type declaration, and $\sigma_1, \dots, \sigma_n, \sigma$ are types. A type declaration is *first-order* if it uses only sorts, otherwise *higher-order*.

We assume given a set of function symbols which are meant to be algebraic operators. Each function symbol f is equipped with a type declaration $f : \sigma_1 \times \cdots \times \sigma_n \rightarrow \sigma$. We use \mathcal{F}_n for the set of function symbols of arity n . \mathcal{F} is a *first-order signature* if all its type declarations are first-order, and a higher-order signature otherwise.

The set of *raw terms* is generated from the signature \mathcal{F} and a denumerable set \mathcal{X} of variables according to the grammar:

$$\mathcal{T} := \mathcal{X} \mid (\lambda \mathcal{X}. \mathcal{T}) \mid @(\mathcal{T}, \mathcal{T}) \mid \mathcal{F}(\mathcal{T}, \dots, \mathcal{T}).$$

Terms generated by the first two grammar rules are called *algebraic*. Terms of the form $\lambda x. u$ are called *abstractions* while terms of the form $@(u, v)$ are called *applications*. The term $@(u, \bar{v})$ is called a (partial) *left-flattening* of $@(\dots @(@ (u, v_1), v_2), \dots, v_n)$, with u being possibly an application itself. Terms other than abstractions are said to be *neutral*. We denote by $\mathcal{V}ar(t)$ ($\mathcal{B}\mathcal{V}ar(t)$) the set of free (bound) variables of t . We may

assume for convenience (and without further notice) that bound variables in a term are all different, and are different from the free ones.

Terms are identified with finite labeled trees by considering $\lambda x.$, for each variable x , as a unary function symbol. *Positions* are strings of positive integers, the empty string Λ denoting the root position. The *subterm* of t at position p is denoted by $t|_p$, and by $t[u]_p$ the result of replacing $t|_p$ at position p in t by u . We write $s \triangleright u$ if u is a strict subterm of s . We use $t[\]_p$ for a term with a hole, called a context. The notation \bar{s} will be ambiguously used to denote a list, a multiset, or a set of terms s_1, \dots, s_n .

2.2 Typing Rules

Typing rules restrict the set of terms by constraining them to follow a precise discipline. Environments are sets of pairs written $x : \sigma$, where x is a variable and σ is a type. Let $\text{Dom}(\Gamma) = \{x \mid x : \sigma \in \Gamma \text{ for some type } \sigma\}$. We assume there is a unique pair of the form $x : \sigma$ for every variable $x \in \text{Dom}(\Gamma)$. Our typing judgments are written as $\Gamma \vdash M : \sigma$ if the term M can be proved to have the type σ in the environment Γ . A term M has type σ in the environment Γ if $\Gamma \vdash M : \sigma$ is provable in the inference system of Figure 1. A term M is typable in the environment Γ if there exists a type σ such that M has type σ in the environment Γ . A term M is typable if it is typable in some environment Γ . Note that function symbols are uncurried, hence must come along with all their arguments.

<p>Variables:</p> $\frac{x : \sigma \in \Gamma}{\Gamma \vdash x : \sigma}$	<p>Functions:</p> $\frac{f : \sigma_1 \times \dots \times \sigma_n \rightarrow \sigma \quad \Gamma \vdash t_1 : \sigma_1 \dots \Gamma \vdash t_n : \sigma_n}{\Gamma \vdash f(t_1, \dots, t_n) : \sigma}$
<p>Abstraction:</p> $\frac{\Gamma \cup \{x : \sigma\} \vdash t : \tau}{\Gamma \vdash (\lambda x : \sigma. t) : \sigma \rightarrow \tau}$	<p>Application:</p> $\frac{\Gamma \cup \{x : \sigma\} \vdash s : \sigma \rightarrow \tau \quad \Gamma \vdash t : \sigma}{\Gamma \vdash @(s, t) : \tau}$

Fig. 1. Typing judgments in higher-order algebras

2.3 Higher-Order Rewrite Rules

Substitutions are written as in $\{x_1 : \sigma_1 \mapsto (\Gamma_1, t_1), \dots, x_n : \sigma_n \mapsto (\Gamma_n, t_n)\}$ where, for every $i \in [1..n]$, t_i is assumed different from x_i and $\Gamma_i \vdash t_i : \sigma_i$. We also assume that $\bigcup_i \Gamma_i$ is an environment. We often write $x \mapsto t$ instead of $x : \sigma \mapsto (\Gamma, t)$, in particular when t is ground. We use the letter γ for substitutions and postfix notation for their application. Substitutions behave as endomorphisms defined on free variables. A (possibly higher-order) *term rewriting system* is a set of rewrite rules $R = \{\Gamma_i \vdash l_i \rightarrow r_i : \sigma_i\}_i$, where l_i and r_i are higher-order terms such that l_i and r_i have the same type σ_i in the environment Γ_i . Given a term rewriting system R , a term s rewrites to a term t at position p with the rule $l \rightarrow r$ and the substitution γ , written $s \xrightarrow[p]{p} t$, or simply $s \rightarrow_R t$, if $s|_p = l\gamma$ and $t = s[r\gamma]_p$.