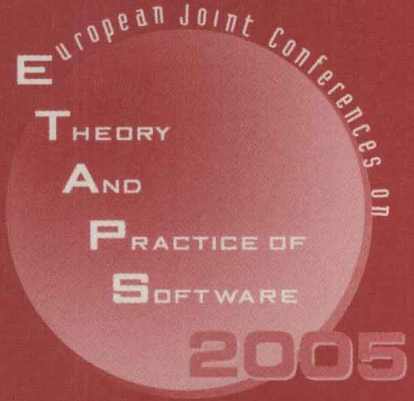
Vladimiro Sassone (Ed.)

LNCS 3441

Foundations of Software Science and Computation Structures

8th International Conference, FOSSACS 2005 Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2005 Edinburgh, UK, April 2005, Proceedings



Springer

Vladimiro Sassone (Ed.)

Foundations of Software Science and Computation Structures

8th International Conference, FOSSACS 2005
Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2005
Edinburgh, UK, April 4-8, 2005
Proceedings

Springer

Volume Editor

Vladimiro Sassone
University of Sussex
Dept. of Informatics
Brighton BN1 9QH, UK
E-mail: v.sassone@sussex.ac.uk

Library of Congress Control Number: Applied for

CR Subject Classification (1998): F.3, F.4.2, F.1.1, D.3.3-4, D.2.1

ISSN 0302-9743
ISBN 3-540-25388-2 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11402060 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 3441

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

Editorial Board

Foreword

ETAPS 2005 was the eighth instance of the *European Joint Conferences on Theory and Practice of Software*. ETAPS is an annual federated conference that was established in 1998 by combining a number of existing and new conferences. This year it comprised five conferences (CC, ESOP, FASE, FOSSACS, TACAS), 17 satellite workshops (AVIS, BYTECODE, CEES, CLASE, CMSB, COCV, FAC, FESCA, FINCO, GCW-DSE, GLPL, LDTA, QAPL, SC, SLAP, TGC, UITP), seven invited lectures (not including those that were specific to the satellite events), and several tutorials. We received over 550 submissions to the five conferences this year, giving acceptance rates below 30% for each one. Congratulations to all the authors who made it to the final program! I hope that most of the other authors still found a way of participating in this exciting event and I hope you will continue submitting.

The events that comprise ETAPS address various aspects of the system development process, including specification, design, implementation, analysis and improvement. The languages, methodologies and tools which support these activities are all well within its scope. Different blends of theory and practice are represented, with an inclination towards theory with a practical motivation on the one hand and soundly based practice on the other. Many of the issues involved in software design apply to systems in general, including hardware systems, and the emphasis on software is not intended to be exclusive.

ETAPS is a loose confederation in which each event retains its own identity, with a separate program committee and proceedings. Its format is open-ended, allowing it to grow and evolve as time goes by. Contributed talks and system demonstrations are in synchronized parallel sessions, with invited lectures in plenary sessions. Two of the invited lectures are reserved for "unifying" talks on topics of interest to the whole range of ETAPS attendees. The aim of cramming all this activity into a single one-week meeting is to create a strong magnet for academic and industrial researchers working on topics within its scope, giving them the opportunity to learn about research in related areas, and thereby to foster new and existing links between work in areas that were formerly addressed in separate meetings.

ETAPS 2005 was organized by the School of Informatics of the University of Edinburgh, in cooperation with

- European Association for Theoretical Computer Science (EATCS);
- European Association for Programming Languages and Systems (EAPLS);
- European Association of Software Science and Technology (EASST).

The organizing team comprised:

- Chair: Don Sannella
- Publicity: David Aspinall
- Satellite Events: Massimo Felici
- Secretariat: Dyane Goodchild
- Local Arrangements: Monika-Jeannette Lekuse

- Tutorials: Alberto Momigliano
- Finances: Ian Stark
- Website: Jennifer Tenzer, Daniel Winterstein
- Fundraising: Phil Wadler

ETAPS 2005 received support from the University of Edinburgh.

Overall planning for ETAPS conferences is the responsibility of its Steering Committee, whose current membership is:

> Perdita Stevens (Edinburgh, Chair), Luca Aceto (Aalborg and Reykjavík), Rastislav Bodik (Berkeley), Maura Cerioli (Genoa), Evelyn Duesterwald (IBM, USA), Hartmut Ehrig (Berlin), José Fiadeiro (Leicester), Marie-Claude Gaudel (Paris), Roberto Gorrieri (Bologna), Reiko Heckel (Paderborn), Holger Hermanns (Saarbrücken), Joost-Pieter Katoen (Aachen), Paul Klint (Amsterdam), Jens Knoop (Vienna), Kim Larsen (Aalborg), Tiziana Margaria (Dortmund), Ugo Montanari (Pisa), Hanne Riis Nielson (Copenhagen), Fernando Orejas (Barcelona), Mooly Sagiv (Tel Aviv), Don Sannella (Edinburgh), Vladimiro Sassone (Sussex), Peter Sestoft (Copenhagen), Michel Wermelinger (Lisbon), Igor Walukiewicz (Bordeaux), Andreas Zeller (Saarbrücken), Lenore Zuck (Chicago).

I would like to express my sincere gratitude to all of these people and organizations, the program committee chairs and PC members of the ETAPS conferences, the organizers of the satellite events, the speakers themselves, the many reviewers, and Springer for agreeing to publish the ETAPS proceedings. Finally, I would like to thank the organizer of ETAPS 2005, Don Sannella. He has been instrumental in the development of ETAPS since its beginning; it is quite beyond the limits of what might be expected that, in addition to all the work he has done as the original ETAPS Steering Committee Chairman and current ETAPS Treasurer, he has been prepared to take on the task of organizing this instance of ETAPS. It gives me particular pleasure to thank him for organizing ETAPS in this wonderful city of Edinburgh in this my first year as ETAPS Steering Committee Chair.

Edinburgh, January 2005

Perdita Stevens
ETAPS Steering Committee Chair

Preface

This volume collects the proceedings of "Foundations of Software Science and Computation Structures," FOSSACS 2005. FOSSACS is a member conference of ETAPS, the "European Joint Conferences on Theory and Practice of Software," dedicated to foundational research for software science. It invites submissions on theories and methods to underpin the analysis, integration, synthesis, transformation, and verification of programs and software systems. Topics covered usually include: algebraic models; automata and language theory; behavioral equivalences; categorical models; computation processes over discrete and continuous data; computation structures; logics of programs; modal, spatial, and temporal logics; models of concurrent, reactive, distributed, and mobile systems; models of security and trust; language-based security; process algebras and calculi; semantics of programming languages; software specification and refinement; and type systems and type theory.

FOSSACS 2005 consisted of one invited and 30 contributed papers, selected out of 108 submissions, yielding an acceptance rate of less than 28%. The quality of the manuscripts was very high indeed, and the Program Committee had to reject several deserving ones. Besides making for a strong 2005 program, this is an indication that FOSSACS is becoming an established point of reference in the international landscape of theoretical computer science. This is a trend that I believe will continue in its forthcoming editions.

Besides Marcelo Fiore's invited talk, the volume includes Ugo Montanari's invited address as an ETAPS unifying speaker. Ugo's '*Model Checking for Nominal Calculi*' reflects broadly on topics in semantics, weaving together verification via semantic equivalences and model checking, Web services, the π-calculus, and the derivation of bisimulation congruences over reactive systems. Marcelo's contribution, '*Mathematical Models of Computational and Combinatorial Structures*,' advocates a combinatorial approach to semantic models by introducing a calculus of *generalized species of structures* as a unification and generalization of models arising in several distinct areas, including his previous work on denotational models of the π-calculus and of variable-binding operators. The conference program was organized into nine sessions, each focusing on reflecting common research topics among the accepted papers. The order of presentation of the papers in this volume maintains the structure of those sessions.

I have a debt of gratitude to the Program Committee for their scholarly effort during the discussion phase; to the referees, for carrying out the reviewing task with competence, care, and precision; to the invited speakers for their inspired work; and ultimately to the authors for submitting their best work to FOSSACS. Thanks to David Aspinall and Don Sannella for the local organization, and to Martin Karusseit and Tiziana Margaria for their support with the conference electronic management system.

I hope you enjoy the volume.

Sussex, January 2005

Vladimiro Sassone
Program Chair
FOSSACS 2005

Organization

Program Committee

Luca Aceto (Aalborg, Denmark)
Luís Caires (Lisbon, Portugal)
Witold Charatonik (Wroclaw, Poland)
Robert Harper (CMU, USA)
Naoki Kobayashi (Tokyo, Japan)
Guy McCusker (Sussex, UK)
Anca Muscholl (LIAFA Paris, France)
Andrew Pitts (Cambridge, UK)
David Sands (Chalmers, Sweden)
Vladimiro Sassone (Sussex, UK)
Peter Selinger (Ottawa, Canada)
Glynn Winskel (Cambridge, UK)
Michele Bugliesi (Venice, Italy)
Giuseppe Castagna (ENS Paris, France)
Vincent Danos (PPS Paris, France)
Petr Jančar (Ostrava, Czech Republic)
Orna Kupferman (Jerusalem, Israel)
Ugo Montanari (Pisa, Italy)
Tobias Nipkow (Munich, Denmark)
Amir Pnueli (Weizmann, Israel and New York, USA)
Andre Scedrov (UPenn, USA)
Wolfgang Thomas (Aachen, Denmark)
Nobuko Yoshida (Imperial, UK)

Referees

Reynald Affeldt
Jonathan Aldrich
Jan Altenbernd
Torben Amtoft
Eugene Asarin
David Aspinall
Franz Baader
Christel Baier
Patrick Baillot
Sebastian Bala
Paolo Baldan
Richard Banach
Nicolas Baudru
Gerd Behrmann
Martin Berger
Ulrich Berger
Gerd Berhmann
Marco Bernardo
Alexis Bes
Stephen Bloom
Richard Blute
Viviana Bono
Ana Bove
Tomas Brázdil
Thomas Brihaye
Stephen Brookes
Franck van Breugel
Marzia Buscemi
Michael Butler
Marco Carbone
Josep Carmona
Alberto Casagrande
Ilaria Castellani
Amine Chaieb
Stefano Chessa
Corina Cirstea
Giovanni Conforti
Thierry Coquand
Silvia Crafa
Karl Crary
Federico Crazzolara
Pedro D'Argenio
Jim Davies
Josée Desharnais
Pietro Di Gianantonio
Ernst-Erich Doberkat
Marie Duflot
Martín Escardó
Alessandro Fantechi
Marcelo Fiore
Riccardo Focardi
Alain Frisch
Fabio Gadducci
Philippe Gaucher
Simon Gay
Blaise Genest
Neil Ghani
Rob van Glabbeek
Daniele Gorla
Eric Goubault
Jean Goubault-Larrecq
Susanne Graf
Erich Grädel
S. Gutierrez-Nolasco
Joshua Guttman
Peter Habermehl

Masahito Hasegawa
Ichiro Hasuo
Thomas Hildebrandt
Daniel Hirschkoff
Kohei Honda
Haruo Hosoya
Jesse Hughes
Michael Huth
Atsushi Igarashi
Florent Jacquemard
Radha Jagadeesan
Alan Jeffrey
Ole Jensen
Gabriel Juhas
Tomasz Jurdzinski
Joost-Pieter Katoen
Emanuel Kieronski
Bartek Klin
Teodor Knapik
Martin Kot
Pavel Krčál
Neel Krishnaswami
Jean Krivine
Jim Laird
Martin Lange
Diego Latella
Francesca Levi
Paul Blain Levy
Christof Löding
Etienne Lozes
Christoph Lüth
Zhaohui Luo
Yoad Lustig
Bas Luttik
Damiano Macedonio
Matteo Maffei
Sergio Maffeis
Jean Mairesse
Rupak Majumdar
Jean-Yves Marion
Keye Martin
Luis Mateu
Paul-André Melliès
Robin Milner
Faron Moller
Luis Monteiro
Carroll Morgan
Rémi Morin
Madhavan Mukund
Markus Müller-Olm
Sumit Nain
Aleks Nanevski
Francesco Zappa Nardelli
Peter Niebert
Damian Niwinski
Gethin Norman
Karol Ostrovsky
Sam Owre
Prakash Panangaden
George Pappas
Matthew Parkinson
Doron Peled
Frank Pfenning
Iain Philipps
Andrew Phillips
Carla Piazza
Brigitte Pientka
Benjamin Pierce
Jean-Eric Pin
Lucia Pomello
K.V.S. Prasad
Sanjiva Prasad
Francesco Ranzato
Julian Rathke
Arend Rensink
James Riely
Philipp Rohde
Bill Roscoe
Sabina Rossi
Pawel Rychlikowski
Zdeněk Sawa
Norbert Schirmer
Alan Schmitt
Lutz Schröder
Robert Seely
Roberto Segala
Olivier Serre
Peter Sewell
Janos Simon
Alex Simpson
Christian Skalka
Paweł Sobociński
Jiří Srba
Ian Stark
Colin Stirling
Mariëlle Stoelinga
Kristian Stovring Sorensen
Oldrich Stražovský
Eijiro Sumii
Vasco T. Vasconcelos
Gabriele Taentzer
Jean-Marc Talbot
Kazushige Terui
Stavros Tripakis
Tomasz Truderung
Emilio Tuosto
Irek Ulidowski
Christian Urban
Tarmo Uustalu
Franck Van Breugel
Daniele Varacca
Maria Grazia Vigliotti
David Walker
Nico Wallmeier
Igor Walukiewicz
Volker Weber
Carsten Weise
Joe Wells
Benjamin Werner
Piotr Wieczorek
Stefan Wöhrle
Burkhart Wolff
James Worrell
Kwangkeun Yi
Shoji Yuen
Marc Zeitoun

Lecture Notes in Computer Science

For information about Vols. 1–3334

please contact your bookseller or Springer

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

Vol. 3448: G.R. Raidl, J. Gottlieb (Eds.), Evolutionary Computation in Combinatorial Optimization. XI, 271 pages. 2005.

Vol. 3441: V. Sassone (Ed.), Foundations of Software Science and Computation Structures. XVIII, 521 pages. 2005.

Vol. 3436: B. Bouyssounouse, J. Sifakis (Eds.), Embedded Systems Design. XV, 492 pages. 2005.

Vol. 3433: S. Bhalla (Ed.), Databases in Networked Information Systems. VII, 319 pages. 2005.

Vol. 3432: M. Beigl, P. Lukowicz (Eds.), Systems Aspects in Organic and Pervasive Computing - ARCS 2005. X, 265 pages. 2005.

Vol. 3427: G. Kotsis, O. Spaniol, Wireless Systems and Mobility in Next Generation Internet. VIII, 249 pages. 2005.

Vol. 3423: J.L. Fiadeiro, P.D. Mosses, F. Orejas (Eds.), Recent Trends in Algebraic Development Techniques. VIII, 271 pages. 2005.

Vol. 3422: R.T. Mittermeir (Ed.), From Computer Literacy to Informatics Fundamentals. X, 203 pages. 2005.

Vol. 3419: B. Faltings, A. Petcu, F. Fages, F. Rossi (Eds.), Constraint Satisfaction and Constraint Logic Programming. X, 217 pages. 2005. (Subseries LNAI).

Vol. 3418: U. Brandes, T. Erlebach (Eds.), Network Analysis. XII, 471 pages. 2005.

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

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

Vol. 3414: M. Morari, L. Thiele (Eds.), Hybrid Systems: Computation and Control. XII, 684 pages. 2005.

Vol. 3412: X. Franch, D. Port (Eds.), COTS-Based Software Systems. XVI, 312 pages. 2005.

Vol. 3411: S.H. Myaeng, M. Zhou, K.-F. Wong, H.-J. Zhang (Eds.), Information Retrieval Technology. XIII, 337 pages. 2005.

Vol. 3410: C.A. Coello Coello, A. Hernández Aguirre, E. Zitzler (Eds.), Evolutionary Multi-Criterion Optimization. XVI, 912 pages. 2005.

Vol. 3409: N. Guelfi, G. Reggio, A. Romanovsky (Eds.), Scientific Engineering of Distributed Java Applications. X, 127 pages. 2005.

Vol. 3408: D.E. Losada, J.M. Fernández-Luna (Eds.), Advances in Information Retrieval. XVII, 572 pages. 2005.

Vol. 3407: Z. Liu, K. Araki (Eds.), Theoretical Aspects of Computing - ICTAC 2004. XIV, 562 pages. 2005.

Vol. 3406: A. Gelbukh (Ed.), Computational Linguistics and Intelligent Text Processing. XVII, 829 pages. 2005.

Vol. 3404: V. Diekert, B. Durand (Eds.), STACS 2005. XVI, 706 pages. 2005.

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

Vol. 3401: Z. Li, L.G. Vulkov, J. Waśniewski (Eds.), Numerical Analysis and Its Applications. XIII, 630 pages. 2005.

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

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

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

Vol. 3395: J. Grabowski, B. Nielsen (Eds.), Formal Approaches to Software Testing. X, 225 pages. 2005.

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

Vol. 3393: H.-J. Kreowski, U. Montanari, F. Orejas, G. Rozenberg, G. Taentzer (Eds.), Formal Methods in Software and Systems Modeling. XXVII, 413 pages. 2005.

Vol. 3391: C. Kim (Ed.), Information Networking. XVII, 936 pages. 2005.

Vol. 3390: R. Choren, A. Garcia, C. Lucena, A. Romanovsky (Eds.), Software Engineering for Multi-Agent Systems III. XII, 291 pages. 2005.

Vol. 3389: P. Van Roy (Ed.), Multiparadigm Programming in Mozart/OZ. XV, 329 pages. 2005.

Vol. 3388: J. Lagergren (Ed.), Comparative Genomics. VII, 133 pages. 2005. (Subseries LNBI).

Vol. 3387: J. Cardoso, A. Sheth (Eds.), Semantic Web Services and Web Process Composition. VIII, 147 pages. 2005.

Vol. 3386: S. Vaudenay (Ed.), Public Key Cryptography - PKC 2005. IX, 436 pages. 2005.

Vol. 3385: R. Cousot (Ed.), Verification, Model Checking, and Abstract Interpretation. XII, 483 pages. 2005.

Vol. 3383: J. Pach (Ed.), Graph Drawing. XII, 536 pages. 2005.

Vol. 3382: J. Odell, P. Giorgini, J.P. Müller (Eds.), Agent-Oriented Software Engineering V. X, 239 pages. 2005.

Vol. 3381: P. Vojtáš, M. Bieliková, B. Charron-Bost, O. Sýkora (Eds.), SOFSEM 2005: Theory and Practice of Computer Science. XV, 448 pages. 2005.

Vol. 3379: M. Hemmje, C. Niederee, T. Risse (Eds.), From Integrated Publication and Information Systems to Information and Knowledge Environments. XXIV, 321 pages. 2005.

Vol. 3378: J. Kilian (Ed.), Theory of Cryptography. XII, 621 pages. 2005.

Vol. 3377: B. Goethals, A. Siebes (Eds.), Knowledge Discovery in Inductive Databases. VII, 190 pages. 2005.

Vol. 3376: A. Menezes (Ed.), Topics in Cryptology – CT-RSA 2005. X, 385 pages. 2005.

Vol. 3375: M.A. Marsan, G. Bianchi, M. Listanti, M. Meo (Eds.), Quality of Service in Multiservice IP Networks. XIII, 656 pages. 2005.

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

Vol. 3372: C. Bussler, V. Tannen, I. Fundulaki (Eds.), Semantic Web and Databases. X, 227 pages. 2005.

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

Vol. 3370: A. Konagaya, K. Satou (Eds.), Grid Computing in Life Science. X, 188 pages. 2005. (Subseries LNBI).

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

Vol. 3368: L. Paletta, J.K. Tsotsos, E. Rome, G.W. Humphreys (Eds.), Attention and Performance in Computational Vision. VIII, 231 pages. 2005.

Vol. 3367: W.S. Ng, B.C. Ooi, A. Ouksel, C. Sartori (Eds.), Databases, Information Systems, and Peer-to-Peer Computing. X, 231 pages. 2005.

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

Vol. 3365: G. Mauri, G. Păun, M.J. Pérez-Jiménez, G. Rozenberg, A. Salomaa (Eds.), Membrane Computing. IX, 415 pages. 2005.

Vol. 3363: T. Eiter, L. Libkin (Eds.), Database Theory - ICDT 2005. XI, 413 pages. 2004.

Vol. 3362: G. Barthe, L. Burdy, M. Huisman, J.-L. Lanet, T. Muntean (Eds.), Construction and Analysis of Safe, Secure, and Interoperable Smart Devices. IX, 257 pages. 2005.

Vol. 3361: S. Bengio, H. Bourlard (Eds.), Machine Learning for Multimodal Interaction. XII, 362 pages. 2005.

Vol. 3360: S. Spaccapietra, E. Bertino, S. Jajodia, R. King, D. McLeod, M.E. Orlowska, L. Strous (Eds.), Journal on Data Semantics II. XI, 223 pages. 2005.

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

Vol. 3358: J. Cao, L.T. Yang, M. Guo, F. Lau (Eds.), Parallel and Distributed Processing and Applications. XXIV, 1058 pages. 2004.

Vol. 3357: H. Handschuh, M.A. Hasan (Eds.), Selected Areas in Cryptography. XI, 354 pages. 2004.

Vol. 3356: G. Das, V.P. Gulati (Eds.), Intelligent Information Technology. XII, 428 pages. 2004.

Vol. 3355: R. Murray-Smith, R. Shorten (Eds.), Switching and Learning in Feedback Systems. X, 343 pages. 2005.

Vol. 3354: M. Margenstern (Ed.), Machines, Computations, and Universality. VIII, 329 pages. 2005.

Vol. 3353: J. Hromkovič, M. Nagl, B. Westfechtel (Eds.), Graph-Theoretic Concepts in Computer Science. XI, 404 pages. 2004.

Vol. 3352: C. Blundo, S. Cimato (Eds.), Security in Communication Networks. XI, 381 pages. 2005.

Vol. 3351: G. Persiano, R. Solis-Oba (Eds.), Approximation and Online Algorithms. VIII, 295 pages. 2005.

Vol. 3350: M. Hermenegildo, D. Cabeza (Eds.), Practical Aspects of Declarative Languages. VIII, 269 pages. 2005.

Vol. 3349: B.M. Chapman (Ed.), Shared Memory Parallel Programming with Open MP. X, 149 pages. 2005.

Vol. 3348: A. Canteaut, K. Viswanathan (Eds.), Progress in Cryptology - INDOCRYPT 2004. XIV, 431 pages. 2004.

Vol. 3347: R.K. Ghosh, H. Mohanty (Eds.), Distributed Computing and Internet Technology. XX, 472 pages. 2004.

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

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

Vol. 3344: J. Malenfant, B.M. Østvold (Eds.), Object-Oriented Technology. ECOOP 2004 Workshop Reader. VIII, 215 pages. 2005.

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

Vol. 3342: E. Şahin, W.M. Spears (Eds.), Swarm Robotics. IX, 175 pages. 2005.

Vol. 3341: R. Fleischer, G. Trippen (Eds.), Algorithms and Computation. XVII, 935 pages. 2004.

Vol. 3340: C.S. Calude, E. Calude, M.J. Dinneen (Eds.), Developments in Language Theory. XI, 431 pages. 2004.

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

Vol. 3338: S.Z. Li, J. Lai, T. Tan, G. Feng, Y. Wang (Eds.), Advances in Biometric Person Authentication. XVIII, 699 pages. 2004.

Vol. 3337: J.M. Barreiro, F. Martin-Sanchez, V. Maojo, F. Sanz (Eds.), Biological and Medical Data Analysis. XI, 508 pages. 2004.

Vol. 3336: D. Karagiannis, U. Reimer (Eds.), Practical Aspects of Knowledge Management. X, 523 pages. 2004. (Subseries LNAI).

Vol. 3335: M. Malek, M. Reitenspieß, J. Kaiser (Eds.), Service Availability. X, 213 pages. 2005.

Table of Contents

Invited Talks

Rule Formats and Bisimulation

Probabilistic Models

Algebraic Models

Games and Automata

Language Analysis

Partial Order Models

Logics

Coalgebraic Modal Logics

Computational Models

Model Checking for Nominal Calculi*

Gian Luigi Ferrari, Ugo Montanari, and Emilio Tuosto

Dipartimento di Informatica, Largo Bruno Pontecorvo 3, 56127 Pisa – Italy

Abstract. Nominal calculi have been shown very effective to formally model a variety of computational phenomena. The models of nominal calculi have often infinite states, thus making model checking a difficult task. In this note we survey some of the approaches for model checking nominal calculi. Then, we focus on *History-Dependent automata*, a syntax-free automaton-based model of mobility. History-Dependent automata have provided the formal basis to design and implement some existing verification toolkits. We then introduce a novel syntax-free setting to model the symbolic semantics of a nominal calculus. Our approach relies on the notions of reactive systems and observed borrowed contexts introduced by Leifer and Milner, and further developed by Sassone, Lack and Sobocinski. We argue that the symbolic semantics model based on borrowed contexts can be conveniently applied to web service discovery and binding.

1 Summary

Model checking has been shown very effective for proving properties of system behaviour whenever a finite model of it can be constructed. The approach is convenient since it does not require formal proofs and since the same automaton-like model can accommodate system specification languages with substantially different syntax and semantics. Among the properties which can be checked, behavioural equivalence is especially important for matching specifications and implementations, for proving the system resistant to certain attacks and for replacing the system with a simpler one with the same properties.

Names have been used in process calculi for representing a variety of different informations concerning addresses, mobility links, continuations, localities, causal dependencies, security keys and session identifiers. When an unbound number of new names can be generated during execution, the models tend to be infinite even in the simplest cases, unless explicit mechanisms are introduced to allocate and garbage collect names, allowing the same states to be reused with different name meanings.

We review some existing syntax-free models for name-passing calculi and focus on *History-Dependent automata* (HD-automata), introduced by Montanari and Pistore in 1995 [62]. HD-automata [62, 63, 71] have been shown a suitable automata-based model for representing Petri nets, CCS with causality and localities and some versions of π-calculus [59, 75].

* Work supported by European Union project PROFUNDIS, Contract No. IST-2001-33100.

V. Sassone (Ed.): FOSSACS 2005, LNCS 3441, pp. 1–24, 2005.

Different versions of HD-automata have been defined. The simplest version can be easily translated to ordinary automata, but possibly with a larger number of states. In a second version, the states are equipped with name symmetries which further reduce the size of the automata. Furthermore, a theory based on coalgebras in a category of "named sets" can be developed for this kind of HD-automata, which extends the applicability of the approach to other nominal calculi and guarantees the existence of the minimal automaton within the same bisimilarity class [64, 34].

HD-automata also constitute the formal basis upon which several verification toolkits have been defined and implemented. The front end towards the π-calculus and the translation algorithm for the simplest version of HD-automata have been implemented in the HAL tool [31, 32], which relies on the JACK verification environment [7] for handling the resulting ordinary automata. The minimisation algorithm, naturally suggested by the coalgebraic framework, has been implemented in the Mihda toolkit [35, 36] within the European project PROFUNDIS. Other versions of HD-automata can be equipped with algebraic operations, and are based on a algebraic-coalgebraic theory [61].

Here we propose a further instance handling the symbolic versions of nominal calculi, where inputs are represented as variables which are instantiated only when needed. As it is the case for logic programming unification, one would like the variables to be instantiated only the least possible, still guaranteeing that all behaviours are eventually explored. The approach we follow relies on the notion of reactive system and of observable borrowed contexts introduced by Leifer and Milner [53, 52] and further developed by Sassone, Lack and Sobocinski [76, 78, 50] using G-categories and adhesive categories. The reduction semantics of reactive systems is extended in order to introduce as borrowed contexts both the variable instantiations needed in the transitions and the ordinary π-calculus actions. It is argued that the symbolic semantics model based on borrowed contexts can be conveniently applied to web service discovery and binding.

In this paper we review the main results on HD-automata setting them in the mainstream research on nominal calculi. The final part of the paper introduces a novel symbolic semantics of π-calculus based on reactive systems and observed borrowed contexts. In our approach, unification is the basic interaction mechanism. We consider this as being the first step toward the definition of a formal framework (models, proof techniques and verification toolkits) for the so-called *service oriented computing* paradigm.

2 Verification via Semantics Equivalence

In the last thirty years the application of formal methods to software engineering has generated techniques and tools to deal with the various facets of the software development process (see e.g. [19] and the references therein). One of the main advantages of exploiting formal techniques consists of the possibility of constructing *abstractions* that approximate behaviours of the system under development. Often, these abstractions are amenable to automatic verification of properties thus providing a support to the certification of software quality.

Among the different proposals, *verification via semantics equivalence* provides a well established framework to deal with the checking of behavioural properties. In this approach, checking behavioural properties is reduced to the problem of contrasting two system abstractions in order to determine whether their behaviours coincide with respect to a suitable notion of semantics equivalence. For instance, it is possible to verify whether an abstraction of the implementation is consistent with its abstract specification. Another example is provided by the *information leak* detection; in [39] the analysis of information flow is done by verifying that the abstraction of the system P is equivalent to another abstraction obtained by suitably restricting the behaviour of P. A similar idea has been exploited in [1] for the analysis of cryptographic protocols.

Bisimilarity [69] has been proved to be an effective basis for verification based on semantics-equivalence of system abstractions described in some process calculus, i.e. Milner's Calculus of Communicating Systems (CCS) [58]. Bisimilarity is a *co-inductive* relation defined over a special class of automata called *labelled transition systems*. A generic labelled transition system (LTS) describes the evolution of a system by its interactions with the external environment. The co-inductive nature of bisimulation provides an effective proof method to establish semantics equivalence: it is sufficient to exhibit a bisimulation relating the two abstractions. Bisimulation-based proof methods have been exploited to establish properties of a variety of systems such as communication protocols, hardware designs and embedded controllers. Moreover, they have been incorporated in several toolkits for the verification of properties. Indeed, finite state verification environments have enjoyed substantial and growing use over the last years. Here, we mention the Concurrency WorkBench [21], the Meije-FC2 tools [8] and the JACK toolkit [7] to cite a few. Several systems of considerable complexity have been formalised and proved correct by exploiting these semantics-based verification environments.

The advent of mobile computing and wireless communication together with the development of applications running over the Internet (*Global Computing Systems*) have introduced software engineering scenarios that are much more dynamic than those handled with the techniques discussed above. Indeed, finite state verification of global computing systems is much more difficult: in this case, even simple systems can generate infinite state spaces. An illustrative example is provided by the π-calculus [59, 75]. The π-calculus primitives are simple but expressive: channel names can be created, communicated (thus giving the possibility of dynamically reconfiguring process acquaintances) and they are subjected to sophisticated scoping rules. The π-calculus is the archetype of name passing or nominal process calculi. Nominal calculi emphasise the principle that name mechanisms (e.g. local name generation, name exchanges, etc.) provide a suitable abstraction to formally explain a wide range of phenomena of global computing systems (see e.g. [80, 41]). Moreover, nominal calculi provide a basic programming model that has been incorporated in suitable libraries or novel programming languages [22, 4]. Finally, the usefulness of names has been also emphasised in practice. For instance, Needham [66] pointed out the role of names for the security of distributed systems. The World Wide Web provides an excellent (perhaps the most important) example of the power of names and name binding/resolution.