Lecture Notes in
Computer Science 1664

Jos C.M. Baeten Sjouke Mauw (Eds.)

CONCUR'99
Concurrency Theory

10th International Conference
Eindhoven, The Netherlands, August 1999
Proceedings

Springer

Jos C.M. Baeten Sjouke Mauw (Eds.)

CONCUR'99 Concurrency Theory

10th International Conference
Eindhoven, The Netherlands, August 24-27, 1999
Proceedings

Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editors

Jos C.M. Baeten
Sjouke Mauw
Eindhoven University of Technology, Department of Computing Science
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
E-mail: {josb,sjouke}@win.tue.nl

Cataloging-in-Publication data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Concurrency theory : 10th international conference ; proceedings / CONCUR '99, Eindhoven, The Netherlands, August 24 - 27, 1999. Jos C. M. Baeten ; Sjouke Mauw (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Singapore ; Tokyo : Springer, 1999
(Lecture notes in computer science ; Vol. 1664)
ISBN 3-540-66425-4

CR Subject Classification (1998): F.3, F.1, D.3, D.1, C.2

ISSN 0302-9743
ISBN 3-540-66425-4 Springer-Verlag Berlin Heidelberg New York

Typesetting: Camera-ready by author
SPIN: 10704232 06/3142 – 5 4 3 2 1 0 Printed on acid-free paper

Lecture Notes in Computer Science 1664

Edited by G. Goos, J. Hartmanis and J. van Leeuwen

Springer

Berlin
Heidelberg
New York
Barcelona
Hong Kong
London
Milan
Paris
Singapore
Tokyo

Preface

This volume contains the proceedings of the 10th International Conference on Concurrency Theory (CONCUR'99) held in Eindhoven, The Netherlands, 24-27 August 1999.

The purpose of the CONCUR conferences is to bring together researchers, developers and students in order to advance the theory of concurrency and promote its applications. Interest in this topic is continuously growing, as a consequence of the importance and ubiquity of concurrent systems and their applications, and of the scientific relevance of their foundations. The scope of CONCUR'99 covers all areas of semantics, logics and verification techniques for concurrent systems. A list of specific topics includes (but is not limited to) concurrency-related aspects of: models of computation and semantic domains, process algebras, Petri nets, event structures, real-time systems, hybrid systems, stochastic systems, decidability, model-checking, verification techniques, refinement techniques, term and graph rewriting, distributed programming, logic constraint programming, object-oriented programming, typing systems and algorithms, case studies, and tools and environments for programming and verification.

The first two CONCUR conferences were held in Amsterdam (NL) in 1990 and 1991, the following ones in Stony Brook (USA), Hildesheim (D), Uppsala (S), Philadelphia (USA), Pisa (I), Warsaw (PL) and Nice (F). The proceedings have appeared in Springer LNCS, as Volumes 458, 527, 630, 715, 836, 962, 1119, 1243, and 1466.

Of the 91 regular papers submitted this year, 32 were accepted for presentation at the conference and are included in the present volume. Apart from these, the conference included four invited presentations, by Rance Cleaveland (State University of New York at Stony Brook, USA), Javier Esparza (Technische Universität München, D), Rob van Glabbeek (Stanford University, USA) and Catuscia Palamidessi (Pennsylvania State University, USA), and three invited tutorials, by Petr Jančar (Technical University of Ostrava, CZ), Nils Klarlund (AT&T Labs Research, USA) and Jan Tretmans (Universiteit Twente, NL).

We want to thank all members of the program committee, and their subreferees, for selecting the papers to be presented.

Special thanks are due to the local organization committee, chaired by Jan Friso Groote. Dragan Bošnački arranged the tool demonstrations, André Engels was webmaster, Kees Middelburg was in charge of the tutorials, and Martijn Oostdijk took care of the submission software (written by Vladimiro Sassone). Local arrangements, and help with registration, were provided by Marcella de Rooij, Desiree Meijers, and Anne-Meta Oversteegen.

The conference had three satellite events, all held on 23 August 1999. These were PROBMIV'99 (Workshop on Probabilistic Methods in Verification), EXPRESS'99 (6th International Workshop on Expressiveness in Concurrency), and VFM (Symposium on Visual Formal Methods). We thank Eindhoven University of Technology for hosting the event and providing many facilities. We thank our sponsors IPA (Institute of Programming Research and Algorithmics, NL), Philips Research Eindhoven, and EESI (Eindhoven Embedded Systems Institute).

Eindhoven
June 1999

Jos Baeten and Sjouke Mauw
Program Committee Co-chairs

CONCUR Steering Committee

Jos Baeten (Chair, Eindhoven, NL)
Eike Best (Oldenburg, D)
Kim Larsen (Aalborg, DK)
Ugo Montanari (Pisa, I)
Scott Smolka (Stony Brook, USA)
Pierre Wolper (Liège, B)

CONCUR'99 Program Committee

Ralph Back (Åbo Akademi University, SF)
Jos Baeten (Eindhoven University of Technology, NL, Co-chair)
Jan Bergstra (Universiteit van Amsterdam, NL)
Manfred Broy (Technische Universität München, D)
Rocco De Nicola (Universitá di Firenze, I)
Andrew Gordon (Microsoft Research, UK)
Roberto Gorrieri (University of Bologna, I)
Tom Henzinger (University of California, USA)
Bengt Jonsson (Uppsala University, S)
Maciej Koutny (University of Newcastle, UK)
Nancy Lynch (MIT, USA)
Sjouke Mauw (Eindhoven University of Technology, NL, Co-chair)
Arend Rensink (University of Twente, NL)
Philippe Schnoebelen (Ecole Normale Supérieure de Cachan, F)
Robert de Simone (INRIA, F)
P.S. Thiagarajan (SPIC Mathematical Institute, IN)
David Walker (Oxford University, UK)
Glynn Winskel (University of Århus, DK)

Referees

Parosh Aziz Abdulla
Luca Aceto
Alessandro Aldini
Luca de Alfaro
Roberto Amadio
Henrik Reif Andersen
Suzana Andova
Ralph Back
Jos Baeten
Christel Baier
Twan Basten
Marek A. Bednarczyk
Michael von der Beeck
Saddek Bensalem
Jan Bergstra
Marco Bernardo
G. Berry
Eike Best
Michele Boreale
Andrzej Borzyszkowski
Victor Bos
Dragan Bošnački
Ahmed Bouajjani
Gerard Boudol
Mario Bravetti
Max Breitling
Antonio Brogi
Stephen Brookes
Nadia Busi
Martin Büchi
P. Caspi
Flavio Corradini
Jordi Cortadella
Pedro R. D'Argenio
Silvano Dal-Zilio
Mads Dam
Philippe Darondeau
Alexandre David
Anuj Dawar
Conrado Daws
Raymond Devillers
André Engels
Emanuela Fachini
G. Ferrari
Fabrice Le Fessant
Hans Fleischhack
Cedric Fournet
Daniel Fridlender
Fabio Gadducci
Mauro Gaspari
Simon Gay
Thomas Gehrke
Herman Geuvers
Rob van Glabbeek
Stefania Gnesi
Andrew Gordon
Roberto Gorrieri
Jan Friso Groote
Irène Guessarian
Dilian Gurov
Paul Hankin
Boudewijn Haverkort
Matthew Hennessy
Thomas Henzinger
Holger Hermanns
Thomas T. Hildebrandt
Tony Hoare
Leszek Holenderski
Kohei Honda
Furio Honsell
Benjamin Horowitz
Franz Huber
Michaela Huhn
Kees Huizing
Thomas Hune
Anna Ingólfsdóttir
Daniel Jackson
Petr Jančar
Ryszard Janicki
Alan Jeffrey
Bengt Jonsson
Marcin Jurdzinski
Joost-Pieter Katoen
Claude Kirchner
Hanna Klaudel
Maciej Koutny
Dexter Kozen
S Krishnan
Ingolf Krueger
Ruurd Kuiper
K. Narayan Kumar
Orna Kupferman
Marta Kwiatkowska
Anna Labella
Rom Langerak
Izak van Langevelde
François Laroussinie
Alexander Lavrov
Francesca Levi
Xuandong Li
Johan Lilius
Jorn Lind-Nielsen
Michele Loreti
Victor Luchangco
Bas Luttik
Nancy Lynch
Davide Marchignoli
A. Mateescu
Sjouke Mauw
Richard Mayr
Kees Middelburg
Leonid Mikhajlov
Anna Mikhajlova
Faron Moller
Rémi Morin
Laurent Mounier
Olaf Mueller
Madhavan Mukund
Gustaf Naeser
Rajagopal Nagarajan
Margherita Napoli
Wolfgang Naraschewski
Monica Nesi
Uwe Nestmann
Rocco De Nicola
Peter Niebert
Mogens Nielsen
Jan Nyström
Vincent van Oostrom
Giuseppe Pappalardo
Joachim Parrow
Adriano Peron

Paul Pettersson
Anna Philippou
Jan Philipps
Claudine Picaronny
Sophie Pinchinat
Marco Pistore
A. Podelski
Jaco van de Pol
Ivan Porres
John Power
Sanjiva Prasad
Leonor Prensa-Nieto
Corrado Priami
Rosario Pugliese
Paola Quaglia
Mauno Rönkkö
R. Ramanujam
J-F. Raskin
Julian Rathke
Andreas Rausch
Antonio Ravara
Michel Reniers
Arend Rensink
Christian Retore
Judi Romijn
Bill Roscoe
Giuseppe Rosolini
Brigitte Rozoy
Theo C. Ruys
Jeff Sanders
Robert Sandner
Davide Sangiorgi
Birgit Schieder
Vincent Schmitt
Philippe Schnoebelen
Claus Schröter
Bernhard Schätz
Roberto Segala
Peter Sewell
Robert de Simone
Arne Skou
Konrad Slind
Scott Smolka
Katharina Spies
Thomas Stauner
Jason Steggles
Colin Stirling
Laurent Théry
P. S. Thiagarajan
Luca Trevisan
Yaroslav S. Usenko
Mandana Vaziri
Betti Venneri
Bjorn Victor
Patrick Viry
Marc Voorhoeve
Farn Wang
Heike Wehrheim
Jack van Wijk
Tim Willemse
Poul Frederick Williams
Glynn Winskel
J. von Wright
Alex Yakovlev
Wang Yi
Nobuko Yoshida
Sergio Yovine
Mark van der Zwaag

Lecture Notes in Computer Science

For information about Vols. 1–1584
please contact your bookseller or Springer-Verlag

Vol. 1587: J. Pieprzyk, R. Safavi-Naini, J. Seberry (Eds.), Information Security and Privacy. Proceedings, 1999. XI, 327 pages. 1999.

Vol. 1589: J.L. Fiadeiro (Ed.), Recent Trends in Algebraic Development Techniques. Proceedings, 1998. X, 341 pages. 1999.

Vol. 1590: P. Atzeni, A. Mendelzon, G. Mecca (Eds.), The World Wide Web and Databases. Proceedings, 1998. VIII, 213 pages. 1999.

Vol. 1592: J. Stern (Ed.), Advances in Cryptology – EUROCRYPT '99. Proceedings, 1999. XII, 475 pages. 1999.

Vol. 1593: P. Sloot, M. Bubak, A. Hoekstra, B. Hertzberger (Eds.), High-Performance Computing and Networking. Proceedings, 1999. XXIII, 1318 pages. 1999.

Vol. 1594: P. Ciancarini, A.L. Wolf (Eds.), Coordination Languages and Models. Proceedings, 1999. IX, 420 pages. 1999.

Vol. 1595: K. Hammond, T. Davie, C. Clack (Eds.), Implementation of Functional Languages. Proceedings, 1998. X, 247 pages. 1999.

Vol. 1596: R. Poli, H.-M. Voigt, S. Cagnoni, D. Corne, G.D. Smith, T.C. Fogarty (Eds.), Evolutionary Image Analysis, Signal Processing and Telecommunications. Proceedings, 1999. X, 225 pages. 1999.

Vol. 1597: H. Zuidweg, M. Campolargo, J. Delgado, A. Mullery (Eds.), Intelligence in Services and Networks. Proceedings, 1999. XII, 552 pages. 1999.

Vol. 1598: R. Poli, P. Nordin, W.B. Langdon, T.C. Fogarty (Eds.), Genetic Programming. Proceedings, 1999. X, 283 pages. 1999.

Vol. 1599: T. Ishida (Ed.), Multiagent Platforms. Proceedings, 1998. VIII, 187 pages. 1999. (Subseries LNAI).

Vol. 1600: M. Wooldridge, M. Veloso (Eds.), Artificial Intelligence Today. VIII, 489 pages. 1999. (Subseries LNAI).

Vol. 1601: J.-P. Katoen (Ed.), Formal Methods for Real-Time and Probabilistic Systems. Proceedings, 1999. X, 355 pages. 1999.

Vol. 1602: A. Sivasubramaniam, M. Lauria (Eds.), Network-Based Parallel Computing. Proceedings, 1999. VIII, 225 pages. 1999.

Vol. 1603: J. Vitek, C.D. Jensen (Eds.), Secure Internet Programming. X, 501 pages. 1999.

Vol. 1604: M. Asada, H. Kitano (Eds.), RoboCup-98: Robot Soccer World Cup II. XI, 509 pages. 1999. (Subseries LNAI).

Vol. 1605: J. Billington, M. Diaz, G. Rozenberg (Eds.), Application of Petri Nets to Communication Networks. IX, 303 pages. 1999.

Vol. 1606: J. Mira, J.V. Sánchez-Andrés (Eds.), Foundations and Tools for Neural Modeling. Proceedings, Vol. I, 1999. XXIII, 865 pages. 1999.

Vol. 1607: J. Mira, J.V. Sánchez-Andrés (Eds.), Engineering Applications of Bio-Inspired Artificial Neural Networks. Proceedings, Vol. II, 1999. XXIII, 907 pages. 1999.

Vol. 1608: S. Doaitse Swierstra, P.R. Henriques, J.N. Oliveira (Eds.), Advanced Functional Programming. Proceedings, 1998. XII, 289 pages. 1999.

Vol. 1609: Z. W. Raś, A. Skowron (Eds.), Foundations of Intelligent Systems. Proceedings, 1999. XII, 676 pages. 1999. (Subseries LNAI).

Vol. 1610: G. Cornuéjols, R.E. Burkard, G.J. Woeginger (Eds.), Integer Programming and Combinatorial Optimization. Proceedings, 1999. IX, 453 pages. 1999.

Vol. 1611: I. Imam, Y. Kodratoff, A. El-Dessouki, M. Ali (Eds.), Multiple Approaches to Intelligent Systems. Proceedings, 1999. XIX, 899 pages. 1999. (Subseries LNAI).

Vol. 1612: R. Bergmann, S. Breen, M. Göker, M. Manago, S. Wess, Developing Industrial Case-Based Reasoning Applications. XX, 188 pages. 1999. (Subseries LNAI).

Vol. 1613: A. Kuba, M. Šámal, A. Todd-Pokropek (Eds.), Information Processing in Medical Imaging. Proceedings, 1999. XVII, 508 pages. 1999.

Vol. 1614: D.P. Huijsmans, A.W.M. Smeulders (Eds.), Visual Information and Information Systems. Proceedings, 1999. XVII, 827 pages. 1999.

Vol. 1615: C. Polychronopoulos, K. Joe, A. Fukuda, S. Tomita (Eds.), High Performance Computing. Proceedings, 1999. XIV, 408 pages. 1999.

Vol. 1616: P. Cointe (Ed.), Meta-Level Architectures and Reflection. Proceedings, 1999. XI, 273 pages. 1999.

Vol. 1617: N.V. Murray (Ed.), Automated Reasoning with Analytic Tableaux and Related Methods. Proceedings, 1999. X, 325 pages. 1999. (Subseries LNAI).

Vol. 1618: J. Bézivin, P.-A. Muller (Eds.), The Unified Modeling Language. Proceedings, 1998. IX, 443 pages. 1999.

Vol. 1619: M.T. Goodrich, C.C. McGeoch (Eds.), Algorithm Engineering and Experimentation. Proceedings, 1999. VIII, 349 pages. 1999.

Vol. 1620: W. Horn, Y. Shahar, G. Lindberg, S. Andreassen, J. Wyatt (Eds.), Artificial Intelligence in Medicine. Proceedings, 1999. XIII, 454 pages. 1999. (Subseries LNAI).

Vol. 1621: D. Fensel, R. Studer (Eds.), Knowledge Acquisition Modeling and Management. Proceedings, 1999. XI, 404 pages. 1999. (Subseries LNAI).

Vol. 1622: M. González Harbour, J.A. de la Puente (Eds.), Reliable Software Technologies – Ada-Europe'99. Proceedings, 1999. XIII, 451 pages. 1999.

Vol. 1623: T. Reinartz, Focusing Solutions for Data Mining. XV, 309 pages. 1999. (Subseries LNAI).

Vol. 1625: B. Reusch (Ed.), Computational Intelligence. Proceedings, 1999. XIV, 710 pages. 1999.

Vol. 1626: M. Jarke, A. Oberweis (Eds.), Advanced Information Systems Engineering. Proceedings, 1999. XIV, 478 pages. 1999.

Vol. 1627: T. Asano, H. Imai, D.T. Lee, S.-i. Nakano, T. Tokuyama (Eds.), Computing and Combinatorics. Proceedings, 1999. XIV, 494 pages. 1999.

Col. 1628: R. Guerraoui (Ed.), ECOOP'99 - Object-Oriented Programming. Proceedings, 1999. XIII, 529 pages. 1999.

Vol. 1629: H. Leopold, N. García (Eds.), Multimedia Applications, Services and Techniques - ECMAST'99. Proceedings, 1999. XV, 574 pages. 1999.

Vol. 1631: P. Narendran, M. Rusinowitch (Eds.), Rewriting Techniques and Applications. Proceedings, 1999. XI, 397 pages. 1999.

Vol. 1632: H. Ganzinger (Ed.), Automated Deduction – Cade-16. Proceedings, 1999. XIV, 429 pages. 1999. (Subseries LNAI).

Vol. 1633: N. Halbwachs, D. Peled (Eds.), Computer Aided Verification. Proceedings, 1999. XII, 506 pages. 1999.

Vol. 1634: S. Džeroski, P. Flach (Eds.), Inductive Logic Programming. Proceedings, 1999. VIII, 303 pages. 1999. (Subseries LNAI).

Vol. 1636: L. Knudsen (Ed.), Fast Software Encryption. Proceedings, 1999. VIII, 317 pages. 1999.

Vol. 1637: J.P. Walser, Integer Optimization by Local Search. XIX, 137 pages. 1999. (Subseries LNAI).

Vol. 1638: A. Hunter, S. Parsons (Eds.), Symbolic and Quantitative Approaches to Reasoning and Uncertainty. Proceedings, 1999. IX, 397 pages. 1999. (Subseries LNAI).

Vol. 1639: S. Donatelli, J. Kleijn (Eds.), Application and Theory of Petri Nets 1999. Proceedings, 1999. VIII, 425 pages. 1999.

Vol. 1640: W. Tepfenhart, W. Cyre (Eds.), Conceptual Structures: Standards and Practices. Proceedings, 1999. XII, 515 pages. 1999. (Subseries LNAI).

Vol. 1642: D.J. Hand, J.N. Kok, M.R. Berthold (Eds.), Advances in Intelligent Data Analysis. Proceedings, 1999. XII, 538 pages. 1999.

Vol. 1643: J. Nešetřil (Ed.), Algorithms – ESA '99. Proceedings, 1999. XII, 552 pages. 1999.

Vol. 1644: J. Wiedermann, P. van Emde Boas, M. Nielsen (Eds.), Automata, Languages, and Programming. Proceedings, 1999. XIV, 720 pages. 1999.

Vol. 1645: M. Crochemore, M. Paterson (Eds.), Combinatorial Pattern Matching. Proceedings, 1999. VIII, 295 pages. 1999.

Vol. 1647: F.J. Garijo, M. Boman (Eds.), Multi-Agent System Engineering. Proceedings, 1999. X, 233 pages. 1999. (Subseries LNAI).

Vol. 1648: M. Franklin (Ed.), Financial Cryptography. Proceedings, 1999. VIII, 269 pages. 1999.

Vol. 1649: R.Y. Pinter, S. Tsur (Eds.), Next Generation Information Technologies and Systems. Proceedings, 1999. IX, 327 pages. 1999.

Vol. 1650: K.-D. Althoff, R. Bergmann, L.K. Branting (Eds.), Case-Based Reasoning Research and Development. Proceedings, 1999. XII, 598 pages. 1999. (Subseries LNAI).

Vol. 1651: R.H. Güting, D. Papadias, F. Lochovsky (Eds.), Advances in Spatial Databases. Proceedings, 1999. XI, 371 pages. 1999.

Vol. 1652: M. Klusch, O.M. Shehory, G. Weiss (Eds.), Cooperative Information Agents III. Proceedings, 1999. XI, 404 pages. 1999. (Subseries LNAI).

Vol. 1653: S. Covaci (Ed.), Active Networks. Proceedings, 1999. XIII, 346 pages. 1999.

Vol. 1654: E.R. Hancock, M. Pelillo (Eds.), Energy Minimization Methods in Computer Vision and Pattern Recognition. Proceedings, 1999. IX, 331 pages. 1999.

Vol. 1656: S. Chatterjee, J.F. Prins, L. Carter, J. Ferrante, Z. Li, D. Sehr, P.-C. Yew (Eds.), Languages and Compilers for Parallel Computing. Proceedings, 1998. XI, 384 pages. 1999.

Vol. 1661: C. Freksa, D.M. Mark (Eds.), Spatial Information Theory. Proceedings, 1999. XIII, 477 pages. 1999.

Vol. 1662: V. Malyshkin (Ed.), Parallel Computing Technologies. Proceedings, 1999. XIX, 510 pages. 1999.

Vol. 1663: F. Dehne, A. Gupta. J.-R. Sack, R. Tamassia (Eds.), Algorithms and Data Structures. Proceedings, 1999. IX, 366 pages. 1999.

Vol. 1664: J.C.M. Baeten, S. Mauw (Eds.), CONCUR'99. Concurrency Theory. Proceedings, 1999. XI, 573 pages. 1999.

Vol. 1666: M. Wiener (Ed.), Advances in Cryptology – CRYPTO '99. Proceedings, 1999. XII, 639 pages. 1999.

Vol. 1668: J.S. Vitter, C.D. Zaroliagis (Eds.), Algorithm Engineering. Proceedings, 1999. VIII, 361 pages. 1999.

Vol. 1671: D. Hochbaum, K. Jansen, J.D.P. Rolim, A. Sinclair (Eds.), Randomization, Approximation, and Combinatorial Optimization. Proceedings, 1999. IX, 289 pages. 1999.

Vol. 1678: M.H. Böhlen, C.S. Jensen, M.O. Scholl (Eds.), Spatio-Temporal Database Management. Proceedings, 1999. X, 243 pages. 1999.

Vol. 1684: G. Ciobanu, G. Păun (Eds.), Fundamentals of Computation Theory. Proceedings, 1999. XI, 570 pages. 1999.

Vol. 1685: P. Amestoy, P. Berger, M. Daydé, I.S. Duff, V. Frayssé, L. Giraud, D. Ruiz (Eds.), Euro-Par'99. Parallel Processing. Proceedings, 1999. XXXII, 1503 pages. 1999.

Vol. 1688: P. Bouquet, L. Serafini, P. Brézillon, M. Benerecetti, F. Castellani (Eds.), Modeling and Using Context. Proceedings, 1999. XII, 528 pages. 1999. (Subseries LNAI).

Vol. 1689: F. Solina, A. Leonardis (Eds.), Computer Analysis of Images and Patterns. Proceedings, 1999. XIV, 650 pages. 1999.

Table of Contents

Invited Lectures

Tutorials

Accepted Papers

Temporal Process Logic*

Rance Cleaveland

Dept. of Comp. Sci., SUNY at Stony Brook, Stony Brook, NY 11794-4400, USA
rance@cs.sunysb.edu

Abstract of Invited Talk

Research in the specification and verification of concurrent systems falls into two general categories. The *temporal logic* school advocates temporal logic as a language for formulating system requirements, with the semantics of the logic being used as a basis for determining whether or not a system is correct. The *process-algebraic* community focuses on the use of "higher-level" system descriptions as specifications of "lower-level" ones, with a refinement relation being used to determine whether an implementation conforms to a specification. From a user's perspective, the approaches offer different benefits and drawbacks. Temporal logic supports "scenario-based" specifications, since formulas may be given that focus on single aspects of system behavior. On the other hand, temporal logic specifications suffer from a lack of compositionality, since the language of specifications differs from the system description language. In contrast, compositional specification is the hallmark of process algebraic reasoning, but at the expense of requiring what some view as overly detailed specifications. Although much research has studied the connections between the temporal logic and process algebra, a truly uniform formalism that combines the advantages of the two approaches has yet to emerge.

In my talk I present preliminary results obtained by Gerald Lüttgen, of ICASE, and me on the development of such a formalism. Our approach features a process-algebra-inspired notation that enriches traditional process algebras by allowing linear-time temporal formulas to be embedded in system descriptions. We show how the combined formalism may be given a uniform operational semantics in Plotkin's Structural Operational Semantics (SOS) style, and we define a refinement relation based on Denicola/Hennessy testing and discuss its congruence properties. We then demonstrate that traditional temporal-logic-style arguments about system correctness can be naturally captured via refinement; we also illustrate how the combination of logical and system operators allows users to define systems in which some "components" remain specified only as formulas.

* Research supported by NSF grants CCR-9257963, CCR-9505562 and CCR-9804091, AFOSR grant F49620-95-1-0508, and ARO grant P-38682-MA.

An Unfolding Algorithm for Synchronous Products of Transition Systems*

Javier Esparza and Stefan Römer

Institut für Informatik, Technische Universität München
e-mail: {esparza,roemer}@in.tum.de

Abstract. The unfolding method, initially introduced for systems modelled by Petri nets, is applied to synchronous products of transition systems, a model introduced by Arnold [2]. An unfolding procedure is provided which exploits the product structure of the model. Its performance is evaluated on a set of benchmarks.

1 Introduction

The unfolding method is a partial order approach to the verification of concurrent systems introduced by McMillan in his Ph. D. Thesis [6]. A finite state system, modelled as a Petri net, is *unfolded* to yield an equivalent acyclic net with a simpler structure. This net is usually infinite, and so in general it cannot be used for automatic verification. However, it is possible to construct a *complete finite prefix* of it containing as much information as the infinite net itself: Loosely speaking, this prefix already contains all the reachable states of the system. The prefix is usually far smaller than the state space, and often smaller than a BDD representation of it, and it can be used as input for efficient verification algorithms. A rather complete bibliography on the unfolding method, containing over 60 papers on semantics, algorithms, and applications is accessible online [1].

The thesis of this paper is that the unfolding method is applicable to any model of concurrency for which a notion of 'events occurring independently from each other' can be defined, and not only to Petri nets—as is often assumed. We provide evidence in favour of this thesis by applying the method to *synchronous products of labelled transition systems.* In this model, introduced by Arnold in [2], a system consists of a tuple of communicating sequential components. The communication discipline, formalised by means of so-called synchronisation vectors, is very general, and contains as special cases the communication mechanisms of process algebras like CCS and CSP.

Readers acquainted with both Arnold's and the Petri net model will probably think that our task is not very difficult, and they are right. It is indeed straightforward to give synchronous products of transition systems a Petri net semantics, and then apply the usual machinery. But we go a bit further: We show that the

* Work partially supported by the Teilprojekt A3 SAM of the Sonderforschungsbereich 342 "Werkzeuge und Methoden für die Nutzung paralleler Rechnerarchitekturen".

additional structure of Arnold's model with respect to Petri nets—the fact that we are given a decomposition of the system into sequential components—can be used to simplify the unfolding method. More precisely, in a former paper by Vogler and the authors [4], we showed that the key to an efficient algorithm for the construction of a complete finite prefix is to find a mathematical object called a *total adequate order*, and provided such an order for systems modelled by Petri nets[1]. In this paper we present a new total adequate order for synchronous products of labelled transition systems. The proof of adequacy for this new order is simpler than the proof of [4].

In a second part of the paper we describe an efficient implementation of the algorithm , and compare it with the algorithm of [4] on a set of benchmarks.

Very recently, further evidence for the wide applicability of unfoldings has been provided by Langerak and Brinksma in [5]. Independently from us, they have applied the unfolding technique to a CSP-like process algebra, a model even further away from Petri nets than ours. A brief discussion of the relation to our work can be found in the conclusions.

The paper is organised as follows. Section 2 introduces synchronous products of transition systems following [2], and Section 3 gives them a partial order semantics based on unfoldings. Section 4 describes an algorithm to construct a complete finite prefix. Section 5 discusses how to efficiently implement it. Section 6 discusses the performance of the new algorithm.

2 Synchronous Products of Transition Systems

In this section we introduce Arnold's model and its standard interleaving semantics. Notations follow [2] with very few minor changes.

2.1 Labelled Transition Systems

A *labelled transition system* is a tuple $\mathcal{A} = \langle S, T, \alpha, \beta, \lambda \rangle$, where S is a set of *states*, T is a set of *transitions*, $\alpha, \beta : T \to S$ are the *source* and *target* mappings, and $\lambda : T \to A$ is a labelling mapping assigning to each transition a letter from an alphabet A. We assume that A contains a special label ϵ, and that for each state $s \in S$ there is a transition ϵ_s such that $\alpha(\epsilon_s) = s = \beta(\epsilon_s)$, and $\lambda(\epsilon_s) = \epsilon$. Moreover, no other transitions are labelled by ϵ. Transitions labelled by ϵ are called *idle* transitions in the sequel.

We use a graphical representation for labelled transition systems. States are represented by circles, and a transition t with $\alpha(t) = s$, $\beta(t) = s'$, and $\lambda(t) = a$ is represented by an arrow leading from s to s' labelled by $t : a$. Idle transitions are not represented. Figure 1 shows two labelled transition systems.

[1] More exactly, systems modelled by 1-safe Petri nets, i.e., Petri nets whose places can hold at most one token.