

Festschrift

LNCS 3113

Juhani Karhumäki
Hermann Maurer
Gheorghe Păun
Grzegorz Rozenberg (Eds.)

Theory Is Forever

Essays Dedicated to Arto Salomaa
on the Occasion of His 70th Birthday



Springer

Juhani Karhumäki Hermann Maurer
Gheorghe Păun Grzegorz Rozenberg (Eds.)

Theory Is Forever

Essays Dedicated to Arto Salomaa
on the Occasion of His 70th Birthday



Springer

Volume Editors

Juhani Karhumäki
University of Turku, Department of Mathematics
20014 Turku, Finland
E-mail: karhumak@cs.utu.fi

Hermann Maurer
Graz University of Technology
Institute for Information Systems and Computer Media
Inffeldgasse 16c, 8010 Graz, Austria
E-mail: hmaurer@iicm.edu

Gheorghe Păun
Romanian Academy, Institute of Mathematics
P.O. Box 1-764, 70700 Bucharest, Romania
E-mail: george.paun@imar.ro

Grzegorz Rozenberg
Leiden University, Leiden Institute of Advanced Computer Science (LIACS)
Niels Bohrweg 1, 2333 CA Leiden, The Netherlands
E-mail: rozenber@liacs.nl

The illustration appearing on the cover of this book is the work of Daniel Rozenberg (DADARA)

Library of Congress Control Number: 2004108213

CR Subject Classification (1998): F.1, F.3, F.4, G.1, G.2

ISSN 0302-9743
ISBN 3-540-22393-2 Springer-Verlag 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-Verlag. Violations are liable to prosecution under the German Copyright Law.

Springer-Verlag is a part of Springer Science+Business Media
springeronline.com

© Springer-Verlag Berlin Heidelberg 2004
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Boller Mediendesign
Printed on acid-free paper SPIN: 11019213 06/3142 5 4 3 2 1 0

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

University of Dortmund, Germany

Madhu Sudan

Massachusetts Institute of Technology, MA, USA

Demetri Terzopoulos

New York University, NY, USA

Doug Tygar

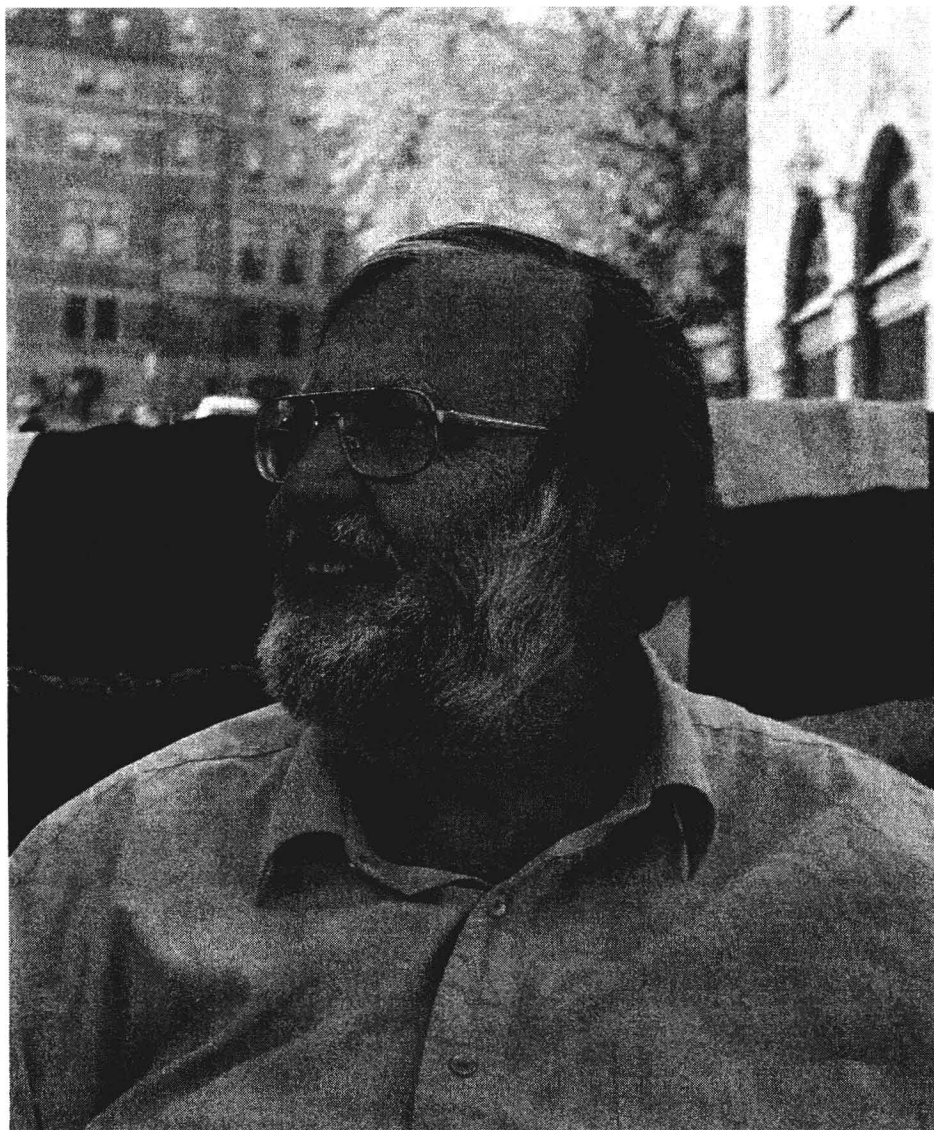
University of California, Berkeley, CA, USA

Moshe Y. Vardi

Rice University, Houston, TX, USA

Gerhard Weikum

Max-Planck Institute of Computer Science, Saarbruecken, Germany



Arto Salomaa

Preface

This Festschrift celebrates the 70th birthday of Arto Kustaa Salomaa (born in Turku, Finland on June 6, 1934), one of the most influential researchers in theoretical computer science.

Most of his research concerns theory – he is one of the founding fathers of formal language and automata theory, but he has also made important contributions to cryptography and natural computing. His approach to research in theoretical computer science is exemplary and inspirational for his students, collaborators, and the readers of his papers and books. For him, the role of theory (in computer science) is to discover general rules of information processing that hold within computer science and in the world around us. One should not waste time on research concerning passing artifacts (or fashionable topics of the moment) in computer science – theory should be permanently predictive, insightful, and inspiring. That’s why we chose the title “Theory is Forever”.

The main source of his influence on theoretical computer science is his publications. Arto is a born writer – his papers and books are always most elegant. He has a unique gift for identifying the real essence of a research problem, and then presenting it in an incisive and eloquent way. He can write about a very involved formal topic and yet avoid a (much too common) overformalization. Many of his writings are genuine jewels and belong to the classics of theoretical computer science. They have inspired generations of students and researchers. Indeed, even computers as well as computer science have learned a lot from Arto’s publications – this is nicely illustrated by DADARA on the cover of this volume. His writing talent extends beyond science – he writes beautiful and engaging stories, and his close friends very much enjoy receiving his long, entertaining and informative letters.

There is much other information that could be cited in this preface, such as the fact that he is one of the most celebrated computer scientists (e.g., he holds eight honorary degrees), or that he has been very instrumental in providing the organizational infrastructure for theoretical computer science in Europe (e.g., he is the past President of the European Association for Theoretical Computer Science), or that he is an absolute authority on the Finnish sauna (including both theory and practice). However, all of these accomplishments have been documented already in many places (e.g., in the companion book “Jewels are Forever”¹ published on the occasion of Arto’s 65th birthday). Thus we have restricted ourselves to reflections on his research and writings.

We are indebted to all the contributors for their tribute to Arto through this book. We ourselves have benefited enormously through many years of collabo-

¹ J. Karhumäki, H. Maurer, G. Păun, G. Rozenberg, *Jewels are Forever, Contributions on Theoretical Computer Science in Honor of Arto Salomaa*, Springer-Verlag, 1999.

ration with Arto from his guidance and friendship – editing this volume is just a token of our gratitude. We are also indebted to Mrs. Ingeborg Mayer from Springer-Verlag for the pleasant and efficient collaboration in producing this volume. As a matter of fact this collaboration is quite symbolic, as Arto has worked very closely with Springer-Verlag, especially with Mrs. Ingeborg Mayer and Dr. Hans Wössner, on many projects over many years. Finally, our special thanks go to T. Harju, M. Hirvensalo, A. Lepistö, and Kalle Saari for their work on this book.

April 2004

Juhani Karhumäki
Hermann Maurer
Gheorghe Păun
Grzegorz Rozenberg

Lecture Notes in Computer Science

For information about Vols. 1–3019

please contact your bookseller or Springer-Verlag

- Vol. 3125: D. Kozen (Ed.), *Mathematics of Program Construction*. X, 401 pages. 2004.
- Vol. 3123: A. Belz, R. Evans, P. Piwek (Eds.), *Natural Language Generation*. X, 219 pages. 2004. (Subseries LNAI).
- Vol. 3120: J. Shawe-Taylor, Y. Singer (Eds.), *Learning Theory*. X, 648 pages. 2004. (Subseries LNAI).
- Vol. 3116: C. Rattray, S. Maharaj, C. Shankland (Eds.), *Algebraic Methodology and Software Technology*. XI, 569 pages. 2004.
- Vol. 3114: R. Alur, D.A. Peled (Eds.), *Computer Aided Verification*. XII, 536 pages. 2004.
- Vol. 3113: J. Karhumäki, H. Maurer, G. Păun, G. Rozenberg (Eds.), *Theory Is Forever*. X, 283 pages. 2004.
- Vol. 3112: H. Williams, L. MacKinnon (Eds.), *New Horizons in Information Management*. XII, 265 pages. 2004.
- Vol. 3111: T. Hagerup, J. Katajainen (Eds.), *Algorithm Theory - SWAT 2004*. XI, 506 pages. 2004.
- Vol. 3109: S.C. Sahinalp, S. Muthukrishnan, U. Dogrusoz (Eds.), *Combinatorial Pattern Matching*. XII, 486 pages. 2004.
- Vol. 3105: S. Göbel, U. Spierling, A. Hoffmann, I. Iurgel, O. Schneider, J. Dechau, A. Feix (Eds.), *Technologies for Interactive Digital Storytelling and Entertainment*. XVI, 304 pages. 2004.
- Vol. 3104: R. Kralovic, O. Sykora (Eds.), *Structural Information and Communication Complexity*. X, 303 pages. 2004.
- Vol. 3103: K. Deb (Ed.), *Genetic and Evolutionary Computation - GECCO 2004*. XLIX, 1439 pages. 2004.
- Vol. 3102: K. Deb (Ed.), *Genetic and Evolutionary Computation - GECCO 2004*. L, 1445 pages. 2004.
- Vol. 3101: M. Masoodian, S. Jones, B. Rogers (Eds.), *Computer Human Interaction*. XIV, 694 pages. 2004.
- Vol. 3099: J. Cortadella, W. Reisig (Eds.), *Applications and Theory of Petri Nets 2004*. XI, 505 pages. 2004.
- Vol. 3098: J. Desel, W. Reisig, G. Rozenberg (Eds.), *Lectures on Concurrency and Petri Nets*. VIII, 849 pages. 2004.
- Vol. 3097: D. Basin, M. Rusinowitch (Eds.), *Automated Reasoning*. XII, 493 pages. 2004. (Subseries LNAI).
- Vol. 3096: G. Melnik, H. Holz (Eds.), *Advances in Learning Software Organizations*. X, 173 pages. 2004.
- Vol. 3094: A. Nürnberger, M. Detyniecki (Eds.), *Adaptive Multimedia Retrieval*. VIII, 229 pages. 2004.
- Vol. 3093: S.K. Katsikas, S. Gritzalis, J. Lopez (Eds.), *Public Key Infrastructure*. XIII, 380 pages. 2004.
- Vol. 3092: J. Eckstein, H. Baumeister (Eds.), *Extreme Programming and Agile Processes in Software Engineering*. XVI, 358 pages. 2004.
- Vol. 3091: V. van Oostrom (Ed.), *Rewriting Techniques and Applications*. X, 313 pages. 2004.
- Vol. 3089: M. Jakobsson, M. Yung, J. Zhou (Eds.), *Applied Cryptography and Network Security*. XIV, 510 pages. 2004.
- Vol. 3086: M. Odersky (Ed.), *ECOOP 2004 – Object-Oriented Programming*. XIII, 611 pages. 2004.
- Vol. 3085: S. Berardi, M. Coppo, F. Damiani (Eds.), *Types for Proofs and Programs*. X, 409 pages. 2004.
- Vol. 3084: A. Persson, J. Stima (Eds.), *Advanced Information Systems Engineering*. XIV, 596 pages. 2004.
- Vol. 3083: W. Emmerich, A.L. Wolf (Eds.), *Component Deployment*. X, 249 pages. 2004.
- Vol. 3080: J. Desel, B. Pernici, M. Weske (Eds.), *Business Process Management*. X, 307 pages. 2004.
- Vol. 3079: Z. Mammeri, P. Lorenz (Eds.), *High Speed Networks and Multimedia Communications*. XVIII, 1103 pages. 2004.
- Vol. 3078: S. Cotin, D.N. Metaxas (Eds.), *Medical Simulation*. XVI, 296 pages. 2004.
- Vol. 3077: F. Roli, J. Kittler, T. Windeatt (Eds.), *Multiple Classifier Systems*. XII, 386 pages. 2004.
- Vol. 3076: D. Buell (Ed.), *Algorithmic Number Theory*. XI, 451 pages. 2004.
- Vol. 3074: B. Kuijpers, P. Revesz (Eds.), *Constraint Databases and Applications*. XII, 181 pages. 2004.
- Vol. 3073: H. Chen, R. Moore, D.D. Zeng, J. Leavitt (Eds.), *Intelligence and Security Informatics*. XV, 536 pages. 2004.
- Vol. 3072: D. Zhang, A.K. Jain (Eds.), *Biometric Authentication*. XVII, 800 pages. 2004.
- Vol. 3070: L. Rutkowski, J. Siekmann, R. Tadeusiewicz, L.A. Zadeh (Eds.), *Artificial Intelligence and Soft Computing - ICAISC 2004*. XXV, 1208 pages. 2004. (Subseries LNAI).
- Vol. 3068: E. André, L. Dybkjær, W. Minker, P. Heisterkamp (Eds.), *Affective Dialogue Systems*. XII, 324 pages. 2004. (Subseries LNAI).
- Vol. 3067: M. Dastani, J. Dix, A. El Fallah-Seghrouchni (Eds.), *Programming Multi-Agent Systems*. X, 221 pages. 2004. (Subseries LNAI).
- Vol. 3066: S. Tsumoto, R. Slowiński, J. Komorowski, J.W. Grzymala-Busse (Eds.), *Rough Sets and Current Trends in Computing*. XX, 853 pages. 2004. (Subseries LNAI).
- Vol. 3065: A. Lomuscio, D. Nute (Eds.), *Deontic Logic in Computer Science*. X, 275 pages. 2004. (Subseries LNAI).
- Vol. 3064: D. Bienstock, G. Nemhauser (Eds.), *Integer Programming and Combinatorial Optimization*. XI, 445 pages. 2004.

- Vol. 3063: Å. Llamas, A. Strohmeier (Eds.), *Reliable Software Technologies - Ada-Europe 2004*. XIII, 333 pages. 2004.
- Vol. 3062: J.L. Pfaltz, M. Nagl, B. Böhlen (Eds.), *Applications of Graph Transformations with Industrial Relevance*. XV, 500 pages. 2004.
- Vol. 3061: F.F. Ramas, H. Unger, V. Larios (Eds.), *Advanced Distributed Systems*. VIII, 285 pages. 2004.
- Vol. 3060: A.Y. Tawfik, S.D. Goodwin (Eds.), *Advances in Artificial Intelligence*. XIII, 582 pages. 2004. (Subseries LNAI).
- Vol. 3059: C.C. Ribeiro, S.L. Martins (Eds.), *Experimental and Efficient Algorithms*. X, 586 pages. 2004.
- Vol. 3058: N. Sebe, M.S. Lew, T.S. Huang (Eds.), *Computer Vision in Human-Computer Interaction*. X, 233 pages. 2004.
- Vol. 3057: B. Jayaraman (Ed.), *Practical Aspects of Declarative Languages*. VIII, 255 pages. 2004.
- Vol. 3056: H. Dai, R. Srikant, C. Zhang (Eds.), *Advances in Knowledge Discovery and Data Mining*. XIX, 713 pages. 2004. (Subseries LNAI).
- Vol. 3055: H. Christiansen, M.-S. Hacid, T. Andreassen, H.L. Larsen (Eds.), *Flexible Query Answering Systems*. X, 500 pages. 2004. (Subseries LNAI).
- Vol. 3054: I. Crnkovic, J.A. Stafford, H.W. Schmidt, K. Wallnau (Eds.), *Component-Based Software Engineering*. XI, 311 pages. 2004.
- Vol. 3053: C. Bussler, J. Davies, D. Fensel, R. Studer (Eds.), *The Semantic Web: Research and Applications*. XIII, 490 pages. 2004.
- Vol. 3052: W. Zimmermann, B. Thalheim (Eds.), *Abstract State Machines 2004. Advances in Theory and Practice*. XII, 235 pages. 2004.
- Vol. 3051: R. Berghammer, B. Möller, G. Struth (Eds.), *Relational and Kleene-Algebraic Methods in Computer Science*. X, 279 pages. 2004.
- Vol. 3050: J. Domingo-Ferrer, V. Torra (Eds.), *Privacy in Statistical Databases*. IX, 367 pages. 2004.
- Vol. 3049: M. Bruynooghe, K.-K. Lau (Eds.), *Program Development in Computational Logic*. VIII, 539 pages. 2004.
- Vol. 3047: F. Oquendo, B. Warboys, R. Morrison (Eds.), *Software Architecture*. X, 279 pages. 2004.
- Vol. 3046: A. Laganà, M.L. Gavrilova, V. Kumar, Y. Mun, C.J.K. Tan, O. Gervasi (Eds.), *Computational Science and Its Applications - ICCSA 2004*. LIII, 1016 pages. 2004.
- Vol. 3045: A. Laganà, M.L. Gavrilova, V. Kumar, Y. Mun, C.J.K. Tan, O. Gervasi (Eds.), *Computational Science and Its Applications - ICCSA 2004*. LIII, 1040 pages. 2004.
- Vol. 3044: A. Laganà, M.L. Gavrilova, V. Kumar, Y. Mun, C.J.K. Tan, O. Gervasi (Eds.), *Computational Science and Its Applications - ICCSA 2004*. LIII, 1140 pages. 2004.
- Vol. 3043: A. Laganà, M.L. Gavrilova, V. Kumar, Y. Mun, C.J.K. Tan, O. Gervasi (Eds.), *Computational Science and Its Applications - ICCSA 2004*. LIII, 1180 pages. 2004.
- Vol. 3042: N. Mitrou, K. Kontovasilis, G.N. Rouskas, I. Iliadis, L. Merakos (Eds.), *NETWORKING 2004, Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications*. XXXIII, 1519 pages. 2004.
- Vol. 3040: R. Conejo, M. Urretavizcaya, J.-L. Pérez-de-la-Cruz (Eds.), *Current Topics in Artificial Intelligence*. XIV, 689 pages. 2004. (Subseries LNAI).
- Vol. 3039: M. Bubak, G.D.v. Albada, P.M.A. Sloot, J.J. Dongarra (Eds.), *Computational Science - ICCS 2004*. LXVI, 1271 pages. 2004.
- Vol. 3038: M. Bubak, G.D.v. Albada, P.M.A. Sloot, J.J. Dongarra (Eds.), *Computational Science - ICCS 2004*. LXVI, 1311 pages. 2004.
- Vol. 3037: M. Bubak, G.D.v. Albada, P.M.A. Sloot, J.J. Dongarra (Eds.), *Computational Science - ICCS 2004*. LXVI, 745 pages. 2004.
- Vol. 3036: M. Bubak, G.D.v. Albada, P.M.A. Sloot, J.J. Dongarra (Eds.), *Computational Science - ICCS 2004*. LXVI, 713 pages. 2004.
- Vol. 3035: M.A. Wimmer (Ed.), *Knowledge Management in Electronic Government*. XII, 326 pages. 2004. (Subseries LNAI).
- Vol. 3034: J. Favela, E. Menasalvas, E. Chávez (Eds.), *Advances in Web Intelligence*. XIII, 227 pages. 2004. (Subseries LNAI).
- Vol. 3033: M. Li, X.-H. Sun, Q. Deng, J. Ni (Eds.), *Grid and Cooperative Computing*. XXXVIII, 1076 pages. 2004.
- Vol. 3032: M. Li, X.-H. Sun, Q. Deng, J. Ni (Eds.), *Grid and Cooperative Computing*. XXXVII, 1112 pages. 2004.
- Vol. 3031: A. Butz, A. Krüger, P. Olivier (Eds.), *Smart Graphics*. X, 165 pages. 2004.
- Vol. 3030: P. Giorgini, B. Henderson-Sellers, M. Winikoff (Eds.), *Agent-Oriented Information Systems*. XIV, 207 pages. 2004. (Subseries LNAI).
- Vol. 3029: B. Orchard, C. Yang, M. Ali (Eds.), *Innovations in Applied Artificial Intelligence*. XXI, 1272 pages. 2004. (Subseries LNAI).
- Vol. 3028: D. Neuenchwander, *Probabilistic and Statistical Methods in Cryptology*. X, 158 pages. 2004.
- Vol. 3027: C. Cachin, J. Camenisch (Eds.), *Advances in Cryptology - EUROCRYPT 2004*. XI, 628 pages. 2004.
- Vol. 3026: C.V. Ramamoorthy, R. Lee, K.W. Lee (Eds.), *Software Engineering Research and Applications*. XV, 377 pages. 2004.
- Vol. 3025: G.A. Vouros, T. Panayiotopoulos (Eds.), *Methods and Applications of Artificial Intelligence*. XV, 546 pages. 2004. (Subseries LNAI).
- Vol. 3024: T. Pajdla, J. Matas (Eds.), *Computer Vision - ECCV 2004*. XXVIII, 621 pages. 2004.
- Vol. 3023: T. Pajdla, J. Matas (Eds.), *Computer Vision - ECCV 2004*. XXVIII, 611 pages. 2004.
- Vol. 3022: T. Pajdla, J. Matas (Eds.), *Computer Vision - ECCV 2004*. XXVIII, 621 pages. 2004.
- Vol. 3021: T. Pajdla, J. Matas (Eds.), *Computer Vision - ECCV 2004*. XXVIII, 633 pages. 2004.

Table of Contents

Duality for Three: Ternary Symmetry in Process Spaces	1
<i>Janusz Brzozowski, Radu Negulescu</i>	
Mathematical Proofs at a Crossroad?	15
<i>Cristian S. Calude, Solomon Marcus</i>	
Rational Relations as Rational Series	29
<i>Christian Choffrut</i>	
Networks of Standard Watson-Crick D0L Systems with Incomplete Information Communication	35
<i>Erzsébet Csuhaj-Varjú</i>	
On the Size of Components of Probabilistic Cooperating Distributed Grammar Systems	49
<i>Erzsébet Csuhaj-Varjú, Jürgen Dassow</i>	
Remarks on Sublanguages Consisting of Primitive Words of Slender Regular and Context-Free Languages	60
<i>Pál Dömösi, Carlos Martín-Vide, Victor Mitrana</i>	
A Semiring-Semimodule Generalization of ω -Context-Free Languages . . .	68
<i>Zoltán Ésik, Werner Kuich</i>	
Integer Weighted Finite Automata, Matrices, and Formal Power Series over Laurent Polynomials	81
<i>Vesa Halava</i>	
Two Models for Gene Assembly in Ciliates	89
<i>Tero Harju, Ion Petre, Grzegorz Rozenberg</i>	
On Self-Dual Bases of the Extensions of the Binary Field	102
<i>Mika Hirvensalo, Jyrki Lahtonen</i>	
On NFA Reductions	112
<i>Lucian Ilie, Gonzalo Navarro, Sheng Yu</i>	
Some Results on Directable Automata	125
<i>Masami Ito, Kayoko Shikishima-Tsuji</i>	
Rectangles and Squares Recognized by Two-Dimensional Automata	134
<i>Jarkko Kari, Cristopher Moore</i>	
Substitution on Trajectories	145
<i>Lila Kari, Stavros Konstantinidis, Petr Sosik</i>	

Recombination Systems 159
Mikko Koivisto, Pasi Rastas, Esko Ukkonen

Algebraic Aspects of Parikh Matrices 170
Alexandru Mateescu

On Distributed Computing on Elliptic Curves 181
Tommi Meskanen, Ari Renvall, Paula Steinby

On the Formal Modelling of Trust in Reputation-Based Systems 192
Mogens Nielsen, Karl Krukow

Issues with Applying Cryptography in Wireless Systems 205
Valtteri Niemi

On a Tomographic Equivalence Between (0,1)-Matrices 216
Maurice Nivat

P Systems with Tables of Rules 235
Gheorghe Păun, Mario J. Pérez-Jiménez, Agustín Riscos-Núñez

Some Properties of Multistage Interconnection Networks 250
Azaria Paz

Structural Equivalence of Regularly Extended EOL Grammars:
An Automata Theoretic Proof 259
Kai Salomaa, Derick Wood

Complexity of Evolving Interactive Systems 268
Peter Verbaan, Jan van Leeuwen, Jiří Wiedermann

Author Index 283

Duality for Three: Ternary Symmetry in Process Spaces

Janusz Brzozowski¹ and Radu Negulescu²

¹ School of Computer Science, University of Waterloo,
Waterloo, ON, Canada N2L 3G1

brzozo@uwaterloo.ca, <http://maveric.uwaterloo.ca>

² Department of Electrical and Computer Engineering
McGill University, Montreal, Québec, Canada H3A 2A7
radu@macs.ece.mcgill.ca

Abstract. Ternary algebra has been used for detection of hazards in logic circuits since 1948. Process spaces have been introduced in 1995 as abstract models of concurrent processes. Surprisingly, process spaces turned out to be special ternary algebras. We study symmetry in process spaces; this symmetry is analogous to duality, but holds among three algebras. An important role is played here by the uncertainty partial order, which has been used since 1972 in algebras dealing with ambiguity. We prove that each process space consists of three isomorphic Boolean algebras and elements related to partitions of a set into three blocks.

1 Introduction

The concept of duality is well known in mathematics. In this paper we study a similar concept, but one that applies to three objects instead of two. The road that led to the discovery of these properties deserves to be briefly mentioned, because several diverse topics come together in this work.

The usual tool for the analysis and design of digital circuits is Boolean algebra, based on two values. As early as 1948, however, it was recognized that three values are useful for describing certain phenomena in logic circuits [10]. We provide more information about the use of ternary algebra for hazard detection in Section 2.

Ternary algebra is closely related to ternary logic [11]. This type of logic, allowing a third, ambiguous value in addition to **true** and **false**, was studied by Mukaidono in 1972 [12], who introduced the *uncertainty* partial order, in addition to the usual lattice partial order. This partial order turned out to be very useful; see, for example, [3, 6]. It also plays an important role in the ternary symmetry we are about to describe.

In 1995 Negulescu [13] introduced process spaces as abstract models of concurrent processes. Surprisingly, process spaces turned out to be special types of ternary algebras. It is in process spaces that “ternary duality” exists. Similar properties also hold in so-called *linear logic*, which has been used as another

framework for representing concurrent processes, and has connections to Petri nets [17]. This topic is outside the scope of the present paper.

The remainder of the paper is structured as follows. Section 2 illustrates hazard detection using ternary algebra. We also recall some basic concepts from lattice theory and summarize the properties of ternary algebras. Process spaces are defined in Section 3. Ternary symmetry is next discussed in Section 4. In Section 5 we show that each process space contains three isomorphic Boolean algebras. Section 6 characterizes elements of a process space that are outside the Boolean algebras, and Section 7 summarizes our results.

We assume that unary operations have precedence over binary operations. For example, $-x + -y$ denotes $(-x) + (-y)$. Sequences of unary operations are written without parentheses; for example, $-/-x$ denotes $-/((-x))$. Set inclusion is denoted by \subseteq and proper inclusion, by \subset . Proofs that are straightforward and involve only elementary set theory are omitted.

2 Ternary Algebras

The logic values are 0 and 1, and a third value, denoted here by Φ , is used to represent an intermediate or uncertain signal. This idea was used by many authors, but we mention here only Eichelberger's 1965 ternary simulation algorithm [8] and its later characterizations [6]. More information about hazard detection can be found in a recent survey [4]. The following example illustrates the use of ternary simulation to detect hazards in logic circuits.

Example 1. Consider the behavior of the circuit of Fig. 1(a) when its input x changes from 0 to 1. Initially, $x = 0$, $y = 1$, and $z = 0$. After the transition, $x = 1$, $y = 0$, and $z = 0$. Thus, z is not supposed to change during this transition. If the inverter has a sufficiently large delay, however, for a short time both inputs to the AND gate may be 1, and there may be a short 1-pulse in z . Such a pulse is undesirable, because it may cause an error in the computation.

In the first part of the ternary simulation, Algorithm A, we change the input to Φ , which indicates that the input is first going through an intermediate, uncertain value. See Fig. 1(b); the first two entries on each line illustrate Algorithm A. The circuit is then analyzed in ternary algebra to determine which gates will undergo changes; the outputs of the changing gates become Φ . In our example, the inverter output becomes uncertain because its input is uncertain. Also, since one input of the AND gate is 1 and the other uncertain, z becomes Φ .

In the second part, Algorithm B, the input is changed to its final binary value, and the circuit is again simulated in ternary algebra. Some gate outputs that became Φ in Algorithm A will become binary, while others remain Φ . In our example, both y and z become 0; see the last two entries in Fig. 1(b). If a gate output has the same (binary) value in the initial state and also at the end of Algorithm B, then that output is not supposed to change during the transition in question. If, however, that output is Φ after Algorithm A is applied, then we have detected a *hazard*, meaning that an undesired pulse may occur. This happens to the output z . \square

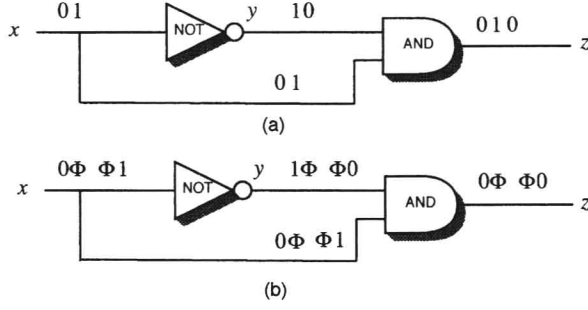


Fig. 1. Circuit with hazard: (a) binary analysis (b) ternary analysis

We now recall some concepts from algebra. For more information about lattices see [1, 7]. We use the following terminology. A *semilattice* [2] is an algebra (S, \sqcup) , where S is a set and \sqcup is an idempotent, commutative and associative binary operation on S . We define the partial order \sqsubseteq_{\sqcup} on S by

$$x \sqsubseteq_{\sqcup} y \Leftrightarrow x \sqcup y = y.$$

A *bisemilattice* is an algebra (S, \sqcup, \sqcap) in which (S, \sqcup) and (S, \sqcap) are semilattices,

Table 1. Laws of de Morgan Algebras

M1	$x \sqcup x = x$	M1'	$x \sqcap x = x$
M2	$x \sqcup y = y \sqcup x$	M2'	$x \sqcap y = y \sqcap x$
M3	$x \sqcup (y \sqcup z) = (x \sqcup y) \sqcup z$	M3'	$x \sqcap (y \sqcap z) = (x \sqcap y) \sqcap z$
M4	$x \sqcup (x \sqcap y) = x$	M4'	$x \sqcap (x \sqcup y) = x$
M5	$x \sqcup \perp = x$	M5'	$x \sqcap \top = x$
M6	$x \sqcup \top = \top$	M6'	$x \sqcap \perp = \perp$
M7	$x \sqcup (y \sqcap z) = (x \sqcup y) \sqcap (x \sqcup z)$	M7'	$x \sqcap (y \sqcup z) = (x \sqcap y) \sqcup (x \sqcap z)$
M8	$--x = x$		
M9	$-(x \sqcup y) = -x \sqcap -y$	M9'	$-(x \sqcap y) = -x \sqcup -y$

i.e., laws M1–M3, M1'–M3' of Table 1 hold. A bisemilattice has two partial orders \sqsubseteq_{\sqcup} and \sqsubseteq_{\sqcap} , the latter defined by

$$x \sqsubseteq_{\sqcap} y \Leftrightarrow x \sqcap y = x.$$

If a bisemilattice satisfies the absorption laws M4 and M4', then it is a *lattice*. The two partial orders \sqsubseteq_{\sqcup} and \sqsubseteq_{\sqcap} then coincide, and are denoted by \sqsubseteq . The converse of \sqsubseteq is denoted by \sqsupseteq . The operations \sqcup and \sqcap are the *join* and *meet* of the lattice, respectively. A lattice is *bounded* if it has greatest and least elements

\top (*top*) and \perp (*bottom*) satisfying M5, M6, M5', M6'. A bounded lattice is represented by $(S, \sqcup, \sqcap, \perp, \top)$. A lattice satisfying the distributive laws M7 and M7' is *distributive*.

A *de Morgan algebra* is an algebra $(S, \sqcup, \sqcap, -, \perp, \top)$, where $(S, \sqcup, \sqcap, \perp, \top)$ is a bounded distributive lattice, and $-$ is a unary operation, called *quasi-complement*, that satisfies M8 and de Morgan's laws M9 and M9'.

A *Boolean algebra* is a de Morgan algebra $(S, \sqcup, \sqcap, -, \perp, \top)$, which also satisfies the complement laws:

$$x \sqcup -x = \top \qquad x \sqcap -x = \perp$$

A *ternary algebra* $(S, \sqcup, \sqcap, -, \perp, \Phi, \top)$ is a de Morgan algebra $(S, \sqcup, \sqcap, -, \perp, \top)$ with an additional constant Φ satisfying

$$\begin{aligned} \text{T1} \quad & -\Phi = \Phi \\ \text{T2} \quad & (x \sqcup -x) \sqcup \Phi = x \sqcup -x \qquad \text{T2'} \quad (x \sqcap -x) \sqcap \Phi = x \sqcap -x \end{aligned}$$

For more information about ternary algebras the reader is referred to [5, 6, 9, 12]. Here, we mention only the uncertainty partial order and the subset-pair representation of ternary algebras.

Figure 2(a) shows the lattice order \sqsubseteq of the 3-element ternary algebra $\mathbf{T}_3 = (\{\perp, \Phi, \top\}, \sqcup, \sqcap, -, \perp, \Phi, \top)$, and Fig. 2(b), its *uncertainty* partial order \preceq [6, 12], where Φ represents the unknown or uncertain value, and \perp and \top are the known or certain values.

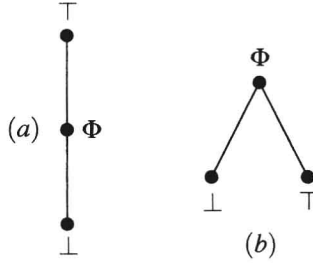


Fig. 2. Partial orders in \mathbf{T}_3 : (a) \sqsubseteq (b) \preceq

For any $x, y \in \mathbf{T}_3$, the least upper bound of $\{x, y\}$ in the partial order \preceq can be expressed as $(x \sqcap y) \sqcup ((x \sqcup y) \sqcap \Phi)$ [6]. We extend this to any ternary algebra $(S, \sqcup, \sqcap, -, \perp, \Phi, \top)$ by defining the binary operation \vee [3] as

$$x \vee y = (x \sqcap y) \sqcup ((x \sqcup y) \sqcap \Phi).$$

It is easily verified that (S, \vee) is a semilattice. The semilattice partial order is

$$x \preceq y \Leftrightarrow x \vee y = y.$$

Let \mathcal{E} be a nonempty set, and \mathcal{P} , a collection of ordered pairs (X, X') of subsets of \mathcal{E} such that $X \cup X' = \mathcal{E}$. For $(X, X'), (Y, Y') \in \mathcal{P}$, let

$$(X, X') \sqcup (Y, Y') = (X \cap Y, X' \cup Y'), \quad (1)$$

$$(X, X') \sqcap (Y, Y') = (X \cup Y, X' \cap Y'), \quad (2)$$

$$-(X, X') = (X', X). \quad (3)$$

Let $\perp = (\mathcal{E}, \emptyset)$, $\Phi = (\mathcal{E}, \mathcal{E})$, and $\top = (\emptyset, \mathcal{E})$. Then $(\mathcal{P}, \sqcup, \sqcap, -, \perp, \Phi, \top)$ is a *subset-pair algebra* [5] if \mathcal{P} is closed under \sqcup , \sqcap , and $-$, and contains the constants \perp , Φ , and \top . The following result was shown in [5, 9]:

Theorem 1. *Every subset-pair algebra is a ternary algebra, and every ternary algebra is isomorphic to a subset-pair algebra.*

It is easy to verify that

$$(X, X') \sqsubseteq (Y, Y') \Leftrightarrow X \supseteq Y, \text{ and } X' \subseteq Y', \quad (4)$$

$$(X, X') \vee (Y, Y') = (X \cup Y, X' \cup Y'), \quad (5)$$

$$(X, X') \preceq (Y, Y') \Leftrightarrow X \subseteq Y \text{ and } X' \subseteq Y'. \quad (6)$$

3 Process Spaces

The material in this section is based on [14, 15]. The discussion of applications of process spaces is beyond the scope of this paper, and we treat process spaces only as mathematical objects. However, we do give a simple example to motivate the reader.

Let \mathcal{E} be any nonempty set; a *process* x over \mathcal{E} is an ordered pair $x = (X, X')$ of subsets of \mathcal{E} such that $X \cup X' = \mathcal{E}$.

We refer to \mathcal{E} as a set of *executions*. Several different examples of execution sets have been used [14, 15]. For the purposes of this paper, however, we may think of \mathcal{E} as the set of all sequences of *actions* from some action *universe* \mathcal{U} ; thus $\mathcal{E} = \mathcal{U}^*$. A process $x = (X, X')$ represents a contract between a device and its environment: the device guarantees that only executions from X occur, and the environment guarantees that only executions from X' occur. Thus, $\overline{X} = \mathcal{E} \setminus X$ is the set of executions in which the device violates the contract. Similarly, for executions in $\overline{X'}$, the environment violates the contract. The condition $X \cup X' = \mathcal{E}$, or equivalently $\overline{X} \cap \overline{X'} = \emptyset$, means that the blame for violating the contract can be assigned to either the device or the environment, but not both. The set X is called the set of *accessible* executions of x , and X' is the set of *acceptable* executions.

Example 2. Figure 3 (a) shows a symbol for a buffer, and Fig. 3 (b) shows a sequential machine describing its behavior. The buffer starts in the state marked by an incoming arrow. If it receives a signal on its input a , it moves to a new state. It is expected to respond by producing a signal on its output b and returning to the original state. Thus, the normal operation of the buffer consists of an alternating sequence of a 's and b 's starting with a . The two states involved in this normal operation are marked g , representing the fact that they are the *goal* states of the process.

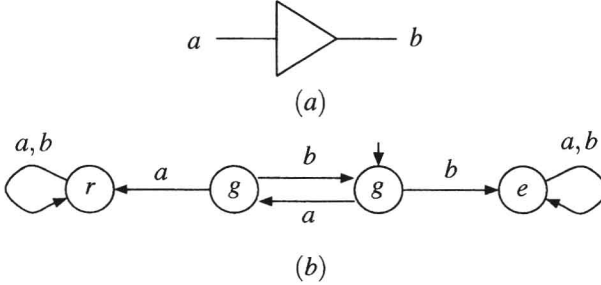


Fig. 3. Buffer process: (a) block diagram (b) behavior

It is possible that the environment of the buffer does not behave according to the specified goal, and produces two consecutive a 's in the initial state. From the point of view of the buffer, this environment behavior can be rejected as illegal; hence the state diagram moves to a *reject* state marked r , and remains in that state thereafter. It is also possible that the buffer malfunctions by producing b in the initial state. This is a violation of the contract by the buffer, and the process moves to the state labelled e ; such executions have been called the *escapes* of the process.

Let L_g be the set of all words taking the machine of Fig. 3 (b) to a state marked g , and let L_e and L_r be defined similarly. One verifies that $L_g = (ab)^*(\epsilon \cup a)$, where ϵ is the empty word, $L_e = (ab)^*b(a \cup b)^*$, and $L_r = (ab)^*aa(a \cup b)^*$. The buffer process is $(X, X') = (L_g \cup L_e, L_g \cup L_r)$. \square

The *process space* over \mathcal{E} is denoted by $\mathcal{P}_{\mathcal{E}}$, and it is the set of all processes over \mathcal{E} . Note that each set \mathcal{E} defines a unique process space.

In constructing $\mathcal{P}_{\mathcal{E}}$ we must put each element of \mathcal{E} in X or X' or both. Hence, if \mathcal{E} has cardinality n , then $\mathcal{P}_{\mathcal{E}}$ has cardinality 3^n . The smallest process space has three elements. If $\mathcal{E} = \{1\}$, say, then the three processes are: $(\{1\}, \emptyset)$, $(\{1\}, \{1\})$, and $(\emptyset, \{1\})$.

In every process space we identify three special elements: *bottom*, $\perp = (\mathcal{E}, \emptyset)$, *void*, $\Phi = (\mathcal{E}, \mathcal{E})$, and *top*, $\top = (\emptyset, \mathcal{E})$.