

Oscar H. Ibarra
Zhe Dang (Eds.)

LNCS 4036

Developments in Language Theory

10th International Conference, DLT 2006
Santa Barbara, CA, USA, June 2006
Proceedings



Springer

TP311.11-53
D489
2006
Oscar H. Ibarra Zhe Dang (Eds.)

Developments in Language Theory

10th International Conference, DLT 2006
Santa Barbara, CA, USA, June 26-29, 2006
Proceedings



Springer



E200603635

Volume Editors

Oscar H. Ibarra
University of California, Department of Computer Science
Santa Barbara, CA 93106, USA
E-mail: ibarra@cs.ucsb.edu

Zhe Dang
Washington State University, School of Electrical Engineering and Computer Science
Pullman, WA 99163, USA
E-mail: zdang@eecs.wsu.edu

Library of Congress Control Number: 2006927282

CR Subject Classification (1998): F.4.3, F.4.2, F.4, F.3, F.1, G.2

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743
ISBN-10 3-540-35428-X Springer Berlin Heidelberg New York
ISBN-13 978-3-540-35428-4 Springer Berlin Heidelberg New York

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

Springer is a part of Springer Science+Business Media
springer.com

© Springer-Verlag Berlin Heidelberg 2006
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11779148 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

David Hutchison

Lancaster University, UK

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

University of California, Los Angeles, CA, 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

Preface

The 10th International Conference on Developments in Language Theory (DLT 2006) was held at the University of California, Santa Barbara, USA on June 26–29, 2006. This was the first DLT conference to take place in North America. Past meetings were held in Turku (1993), Magdeburg (1995), Thessaloniki (1997), Aachen (1999), Vienna (2001), Kyoto (2002), Szeged (2003), Auckland (2004), and Palermo (2005). The conference series is under the auspices of the European Association for Theoretical Computer Science.

The scope of the conference includes topics in the following areas: grammars, acceptors and transducers for strings, trees, graphs, arrays; efficient text algorithms; algebraic theories for automata and languages; combinatorial and algebraic properties of words and languages; variable-length codes; symbolic dynamics; decision problems; relations to complexity theory and logic; picture description and analysis; polyominoes and bidimensional patterns; cryptography; concurrency; bio-inspired computing; quantum computing.

This volume of *Lecture Notes in Computer Science* contains the papers that were presented at DLT 2006, including the abstracts or full papers of four invited lectures presented by Rajeev Alur, Yuri Gurevich, Gheorghe Paun, and Grzegorz Rozenberg.

The 36 contributed papers were selected from 63 submissions. Each submitted paper was reviewed by three Program Committee members, with the assistance of external referees. The authors of the papers came from the following countries and regions: Canada, Czech Republic, Finland, France, Germany, Greece, Hong Kong, Hungary, India, Italy, Japan, Korea, The Netherlands, Poland, Romania, Russia, South Africa, Spain, Sweden, the UK, and the USA.

A great many contributed to the success of DLT 2006. We extend our sincere thanks to the authors who submitted papers and to all those who gave presentations. We express our appreciation to the members of the Program Committee and their colleagues who assisted in the review process. To our invited speakers, we thank you for sharing your insights and expertise. We would like to acknowledge the work of the Organizing Committee, who thoughtfully and energetically planned the event over the course of months. To members of the Steering Committee, we are grateful for your counsel. Finally, we wish to recognize the kind support of the conference sponsors: Ask.com, Citrix, Google, and UCSB's Department of Computer Science, College of Engineering, and Graduate Division.

June 2006

Oscar H. Ibarra

Organization

Program Committee

Jean Berstel (Marne-la-Vallée, France)
Cristian Calude (Auckland, New Zealand)
Erzsebet Csuhaj-Varju (Budapest, Hungary)
Zhe Dang (Pullman, USA)
Volker Diekert (Stuttgart, Germany)
Omer Egecioglu (Santa Barbara, USA)
Zoltan Esik (Szeged, Hungary & Tarragona, Spain)
Juraj Hromkovic (Zurich, Switzerland)
Oscar H. Ibarra, Chair (Santa Barbara, USA)
Masami Ito (Kyoto, Japan)
Natasha Jonoska (Tampa, USA)
Juhani Karhumaki (Turku, Finland)
Lila Kari (London, Canada)
Werner Kuich (Vienna, Austria)
Giancarlo Mauri (Milan, Italy)
Jean-Eric Pin (Paris, France)
Bala Ravikumar (Sonoma, USA)
Antonio Restivo (Palermo, Italy)
Kai Salomaa (Kingston, Canada)
Jeffrey Shallit (Waterloo, Canada)
Wolfgang Thomas (Aachen, Germany)
Hsu-Chun Yen (Taipei, Taiwan)
Sheng Yu (London, Canada)

Organizing Committee

Omer Egecioglu, Co-chair (Santa Barbara, USA)
Bee Jay Estalilla (Santa Barbara, USA)
Cagdas Gerede (Santa Barbara, USA)
Oscar H. Ibarra, Co-chair (Santa Barbara, USA)
Jianwen Su (Santa Barbara, USA)
Sara Woodworth (Santa Barbara, USA)

Proceedings Committee

Zhe Dang (Pullman, USA)
Oscar H. Ibarra (Santa Barbara, USA)

Publicity Committee

Kai Salomaa (Kingston, Canada)

Sheng Yu (London, Canada)

Steering Committee

Jean Berstel (Marne-la-Vallée, France)

Cristian Calude (Auckland, New Zealand)

Volker Diekert (Stuttgart, Germany)

Juraj Hromkovic (Zurich, Switzerland)

Oscar H. Ibarra (Santa Barbara, USA)

Masami Ito (Kyoto, Japan)

Werner Kuich (Vienna, Austria)

Gheorghe Paun (Bucharest, Romania & Seville, Spain)

Antonio Restivo (Palermo, Italy)

Grzegorz Rozenberg, Chair (Leiden, The Netherlands)

Arto Salomaa (Turku, Finland)

Sheng Yu (London, Canada)

External Referees

Marcella Anselmo, Marie-Pierre Béal, Franziska Biegler, Hans-Joachim Boeckenhauer, Bernd Borchert, Henning Bordihn, Giusi Castiglione, Mark Daley, Jürgen Dassow, Jörg Desel, Mike Domaratzki, Pal Domosi, Claudio Ferretti, Wan Fokkink, Zoltan Fulop, Stefan Göller, Valentin Goranko, Vesa Halava, Tero Harju, Markus Holzer, Geza Horvath, Lucian Ilie, Jarkko Kari, Joachim Kupke, Martin Kutrib, A. Lepistö, Christof Löding, Markus Lohrey, Sylvain Lombardy, Robert Lorenz, Marina Madonia, Stuart W. Margolis, Tobias Moemke, Anca Muscholl, Loránd Muzamel, Benedek Nagy, Zoltan L. Nemeth, Alexander Okhotin, Friedrich Otto, Holger Petersen, Narad Rampersad, Grigore Rosu, Nicolae Santean, Stefan Schwoon, Marinella Sciortino, Sebastian Seibert, Olivier Serre, Petr Sosik, Howard Straubing, Simayk Taati, Siegbert Tiga, Sandor Vagvolgyi, György Vaszil, Bow-Yaw Wang, Pascal Weil, Gaoyan Xie, Claudio Zandron

Sponsors

Computer Science Department, University of California, Santa Barbara

College of Engineering, University of California, Santa Barbara

Graduate Division, University of California, Santa Barbara

Ask.com

Citrix

Google

Lecture Notes in Computer Science

For information about Vols. 1–3956

please contact your bookseller or Springer

- Vol. 4060: K. Futatsugi, J.-P. Jouannaud, J. Meseguer (Eds.), *Algebra, Meaning and Computation*. XXXVIII, 643 pages. 2006.
- Vol. 4058: L.M. Batten, R. Safavi-Naini (Eds.), *Information Security and Privacy*. XII, 446 pages. 2006.
- Vol. 4055: J. Lee, J. Shim, S.-g. Lee, C. Bussler, S. Shim (Eds.), *Data Engineering Issues in E-Commerce and Services*. IX, 290 pages. 2006.
- Vol. 4054: A. Horváth, M. Telek (Eds.), *Formal Methods and Stochastic Models for Performance Evaluation*. VIII, 239 pages. 2006.
- Vol. 4053: M. Ikeda, K.D. Ashley, T.-W. Chan (Eds.), *Intelligent Tutoring Systems*. XXVI, 821 pages. 2006.
- Vol. 4044: P. Abrahamsson, M. Marchesi, G. Succì (Eds.), *Extreme Programming and Agile Processes in Software Engineering*. XII, 230 pages. 2006.
- Vol. 4043: A.S. Atzeni, A. Lioy (Eds.), *Public Key Infrastructure*. XI, 261 pages. 2006.
- Vol. 4041: S.-W. Cheng, C.K. Poon (Eds.), *Algorithmic Aspects in Information and Management*. XI, 395 pages. 2006.
- Vol. 4040: R. Reulke, U. Eckardt, B. Flach, U. Knauer, K. Polthier (Eds.), *Combinatorial Image Analysis*. XII, 482 pages. 2006.
- Vol. 4039: M. Morisio (Ed.), *Reuse of Off-the-Shelf Components*. XIII, 444 pages. 2006.
- Vol. 4038: P. Ciancarini, H. Wiklicky (Eds.), *Coordination Models and Languages*. VIII, 299 pages. 2006.
- Vol. 4037: R. Gorrieri, H. Wehrheim (Eds.), *Formal Methods for Open Object-Based Distributed Systems*. XVII, 474 pages. 2006.
- Vol. 4036: O. H. Ibarra, Z. Dang (Eds.), *Developments in Language Theory*. XII, 456 pages. 2006.
- Vol. 4034: J. Münch, M. Vierimaa (Eds.), *Product-Focused Software Process Improvement*. XVII, 474 pages. 2006.
- Vol. 4033: B. Stiller, P. Reichl, B. Tuffin (Eds.), *Performability Has its Price*. X, 103 pages. 2006.
- Vol. 4031: M. Ali, R. Dapoigny (Eds.), *Innovations in Applied Artificial Intelligence*. XXIII, 1353 pages. 2006. (Sublibrary LNAI).
- Vol. 4027: H.L. Larsen, G. Pasi, D. Ortiz-Arroyo, T. Andreassen, H. Christiansen (Eds.), *Flexible Query Answering Systems*. XVIII, 714 pages. 2006. (Sublibrary LNAI).
- Vol. 4026: P. Gibbons, T. Abdelzaher, J. Aspnes, R. Rao (Eds.), *Distributed Computing in Sensor Systems*. XIV, 566 pages. 2006.
- Vol. 4025: F. Eliassen, A. Montresor (Eds.), *Distributed Applications and Interoperable Systems*. XI, 355 pages. 2006.
- Vol. 4024: S. Donatelli, P. S. Thiagarajan (Eds.), *Petri Nets and Other Models of Concurrency - ICATPN 2006*. XI, 441 pages. 2006.
- Vol. 4021: E. André, L. Dybkjær, W. Minker, H. Neumann, M. Weber (Eds.), *Perception and Interactive Technologies*. XI, 217 pages. 2006. (Sublibrary LNAI).
- Vol. 4020: A. Bredenfeld, A. Jacoff, I. Noda, Y. Taka-hashi, RoboCup 2005: Robot Soccer World Cup IX. XVII, 727 pages. 2006. (Sublibrary LNAI).
- Vol. 4018: V. Wade, H. Ashman, B. Smyth (Eds.), *Adaptive Hypermedia and Adaptive Web-Based Systems*. XVI, 474 pages. 2006.
- Vol. 4016: J.X. Yu, M. Kitsuregawa, H.V. Leong (Eds.), *Advances in Web-Age Information Management*. XVII, 606 pages. 2006.
- Vol. 4013: L. Lamontagne, M. Marchand (Eds.), *Advances in Artificial Intelligence*. XIII, 564 pages. 2006. (Sublibrary LNAI).
- Vol. 4012: T. Washio, A. Sakurai, K. Nakashima, H. Takeda, S. Tojo, M. Yokoo (Eds.), *New Frontiers in Artificial Intelligence*. XIII, 484 pages. 2006. (Sublibrary LNAI).
- Vol. 4011: Y. Sure, J. Domingue (Eds.), *The Semantic Web: Research and Applications*. XIX, 726 pages. 2006.
- Vol. 4010: S. Dunne, B. Stoddart (Eds.), *Unifying Theories of Programming*. VIII, 257 pages. 2006.
- Vol. 4009: M. Lewenstein, G. Valiente (Eds.), *Combinatorial Pattern Matching*. XII, 414 pages. 2006.
- Vol. 4007: C. Álvarez, M. Serna (Eds.), *Experimental Algorithms*. XI, 329 pages. 2006.
- Vol. 4006: L.M. Pinho, M. González Harbour (Eds.), *Reliable Software Technologies – Ada-Europe 2006*. XII, 241 pages. 2006.
- Vol. 4005: G. Lugosi, H.U. Simon (Eds.), *Learning Theory*. XI, 656 pages. 2006. (Sublibrary LNAI).
- Vol. 4004: S. Vaudenay (Ed.), *Advances in Cryptology - EUROCRYPT 2006*. XIV, 613 pages. 2006.
- Vol. 4003: Y. Koucheryav, J. Harju, V.B. Iversen (Eds.), *Next Generation Teletraffic and Wired/Wireless Advanced Networking*. XVI, 582 pages. 2006.
- Vol. 4001: E. Dubois, K. Pohl (Eds.), *Advanced Information Systems Engineering*. XVI, 560 pages. 2006.
- Vol. 3999: C. Kop, G. Fliedl, H.C. Mayr, E. Métais (Eds.), *Natural Language Processing and Information Systems*. XIII, 227 pages. 2006.

- Vol. 3998: T. Calamoneri, I. Finocchi, G.F. Italiano (Eds.), Algorithms and Complexity. XII, 394 pages. 2006.
- Vol. 3997: W. Grieskamp, C. Weise (Eds.), Formal Approaches to Software Testing. XII, 219 pages. 2006.
- Vol. 3996: A. Keller, J.-P. Martin-Flatin (Eds.), Self-Managed Networks, Systems, and Services. X, 185 pages. 2006.
- Vol. 3995: G. Müller (Ed.), Emerging Trends in Information and Communication Security. XX, 524 pages. 2006.
- Vol. 3994: V.N. Alexandrov, G.D. van Albada, P.M.A. Sloot, J. Dongarra (Eds.), Computational Science – ICCS 2006, Part IV. XXXV, 1096 pages. 2006.
- Vol. 3993: V.N. Alexandrov, G.D. van Albada, P.M.A. Sloot, J. Dongarra (Eds.), Computational Science – ICCS 2006, Part III. XXXVI, 1136 pages. 2006.
- Vol. 3992: V.N. Alexandrov, G.D. van Albada, P.M.A. Sloot, J. Dongarra (Eds.), Computational Science – ICCS 2006, Part II. XXXV, 1122 pages. 2006.
- Vol. 3991: V.N. Alexandrov, G.D. van Albada, P.M.A. Sloot, J. Dongarra (Eds.), Computational Science – ICCS 2006, Part I. LXXXI, 1096 pages. 2006.
- Vol. 3990: J. C. Beck, B.M. Smith (Eds.), Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. X, 301 pages. 2006.
- Vol. 3989: J. Zhou, M. Yung, F. Bao, Applied Cryptography and Network Security. XIV, 488 pages. 2006.
- Vol. 3987: M. Hazas, J. Krumm, T. Strang (Eds.), Location- and Context-Awareness. X, 289 pages. 2006.
- Vol. 3986: K. Stølen, W.H. Winsborough, F. Martinelli, F. Massacci (Eds.), Trust Management. XIV, 474 pages. 2006.
- Vol. 3984: M. Gavrilova, O. Gervasi, V. Kumar, C.J. K. Tan, D. Taniar, A. Laganà, Y. Mun, H. Choo (Eds.), Computational Science and Its Applications - ICCSA 2006, Part V. XXV, 1045 pages. 2006.
- Vol. 3983: M. Gavrilova, O. Gervasi, V. Kumar, C.J. K. Tan, D. Taniar, A. Laganà, Y. Mun, H. Choo (Eds.), Computational Science and Its Applications - ICCSA 2006, Part IV. XXVI, 1191 pages. 2006.
- Vol. 3982: M. Gavrilova, O. Gervasi, V. Kumar, C.J. K. Tan, D. Taniar, A. Laganà, Y. Mun, H. Choo (Eds.), Computational Science and Its Applications - ICCSA 2006, Part III. XXV, 1243 pages. 2006.
- Vol. 3981: M. Gavrilova, O. Gervasi, V. Kumar, C.J. K. Tan, D. Taniar, A. Laganà, Y. Mun, H. Choo (Eds.), Computational Science and Its Applications - ICCSA 2006, Part II. XXVI, 1255 pages. 2006.
- Vol. 3980: M. Gavrilova, O. Gervasi, V. Kumar, C.J. K. Tan, D. Taniar, A. Laganà, Y. Mun, H. Choo (Eds.), Computational Science and Its Applications - ICCSA 2006, Part I. LXXV, 1199 pages. 2006.
- Vol. 3979: T.S. Huang, N. Sebe, M.S. Lew, V. Pavlović, M. Kölsch, A. Galata, B. Kisačanin (Eds.), Computer Vision in Human-Computer Interaction. XII, 121 pages. 2006.
- Vol. 3978: B. Hnich, M. Carlsson, F. Fages, F. Rossi (Eds.), Recent Advances in Constraints. VIII, 179 pages. 2006. (Sublibrary LNAI).
- Vol. 3977: N. Fuhr, M. Lalmas, S. Malik, G. Kazai (Eds.), Advances in XML Information Retrieval and Evaluation. XII, 556 pages. 2006.
- Vol. 3976: F. Boavida, T. Plagemann, B. Stiller, C. Westphal, E. Monteiro (Eds.), Networking 2006. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems. XXVI, 1276 pages. 2006.
- Vol. 3975: S. Mehrotra, D.D. Zeng, H. Chen, B.M. Thuraisingham, F.-Y. Wang (Eds.), Intelligence and Security Informatics. XXII, 772 pages. 2006.
- Vol. 3973: J. Wang, Z. Yi, J.M. Zurada, B.-L. Lu, H. Yin (Eds.), Advances in Neural Networks - ISNN 2006, Part III. XXIX, 1402 pages. 2006.
- Vol. 3972: J. Wang, Z. Yi, J.M. Zurada, B.-L. Lu, H. Yin (Eds.), Advances in Neural Networks - ISNN 2006, Part II. XXVII, 1444 pages. 2006.
- Vol. 3971: J. Wang, Z. Yi, J.M. Zurada, B.-L. Lu, H. Yin (Eds.), Advances in Neural Networks - ISNN 2006, Part I. LXVII, 1442 pages. 2006.
- Vol. 3970: T. Braun, G. Carle, S. Fahmy, Y. Koucheryavy (Eds.), Wired/Wireless Internet Communications. XIV, 350 pages. 2006.
- Vol. 3969: Ø. Ytrehus (Ed.), Coding and Cryptography. XI, 443 pages. 2006.
- Vol. 3968: K.P. Fishkin, B. Schiele, P. Nixon, A. Quigley (Eds.), Pervasive Computing. XV, 402 pages. 2006.
- Vol. 3967: D. Grigoriev, J. Harrison, E.A. Hirsch (Eds.), Computer Science – Theory and Applications. XVI, 684 pages. 2006.
- Vol. 3966: Q. Wang, D. Pfahl, D.M. Raffo, P. Wernick (Eds.), Software Process Change. XIV, 356 pages. 2006.
- Vol. 3965: M. Bernardo, A. Cimatti (Eds.), Formal Methods for Hardware Verification. VII, 243 pages. 2006.
- Vol. 3964: M. Ü. Uyar, A.Y. Duale, M.A. Fecko (Eds.), Testing of Communicating Systems. XI, 373 pages. 2006.
- Vol. 3963: O. Dikenelli, M.-P. Gleizes, A. Ricci (Eds.), Engineering Societies in the Agents World VI. XII, 303 pages. 2006. (Sublibrary LNAI).
- Vol. 3962: W. IJsselstein, Y. de Kort, C. Midden, B. Eggen, E. van den Hoven (Eds.), Persuasive Technology. XII, 216 pages. 2006.
- Vol. 3960: R. Vieira, P. Quaresma, M.d.G.V. Nunes, N.J. Mamede, C. Oliveira, M.C. Dias (Eds.), Computational Processing of the Portuguese Language. XII, 274 pages. 2006. (Sublibrary LNAI).
- Vol. 3959: J.-Y. Cai, S. B. Cooper, A. Li (Eds.), Theory and Applications of Models of Computation. XV, 794 pages. 2006.
- Vol. 3958: M. Yung, Y. Dodis, A. Kiayias, T. Malkin (Eds.), Public Key Cryptography - PKC 2006. XIV, 543 pages. 2006.

Table of Contents

Invited Lectures

Adding Nesting Structure to Words <i>Rajeev Alur, P. Madhusudan</i>	1
Can Abstract State Machines Be Useful in Language Theory? <i>Yuri Gurevich, Charles Wallace</i>	14
Languages in Membrane Computing: Some Details for Spiking Neural P Systems <i>Gheorghe Păun</i>	20
Computational Nature of Biochemical Reactions <i>A. Ehrenfeucht, G. Rozenberg</i>	36

Papers

Polynomials, Fragments of Temporal Logic and the Variety DA over Traces <i>Manfred Kufleitner</i>	37
Weighted Automata and Weighted Logics on Infinite Words <i>Manfred Droste, George Rahonis</i>	49
Simulation Relations for Alternating Parity Automata and Parity Games <i>Carsten Fritz, Thomas Wilke</i>	59
Equivalence of Functions Represented by Simple Context-Free Grammars with Output <i>Cédric Bastien, Jurek Czyzowicz, Wojciech Fraczak, Wojciech Rytter</i>	71
On the Gap-Complexity of Simple RL-Automata <i>F. Mráz, F. Otto, M. Plátek</i>	83
Noncanonical LALR(1) Parsing <i>Sylvain Schmitz</i>	95
Context-Free Grammars and XML Languages <i>Alberto Bertoni, Christian Choffrut, Beatrice Palano</i>	108

Synchronization of Pushdown Automata <i>Didier Caucal</i>	120
Context-Dependent Nondeterminism for Pushdown Automata <i>Martin Kutrib, Andreas Malcher</i>	133
Prime Decompositions of Regular Languages <i>Yo-Sub Han, Kai Salomaa, Derick Wood</i>	145
On Weakly Ambiguous Finite Transducers <i>Nicolae Santean, Sheng Yu</i>	156
Ciliate Bio-operations on Finite String Multisets <i>Jürgen Dassow, György Vaszil</i>	168
Characterizing DNA Bond Shapes Using Trajectories <i>Michael Domaratzki</i>	180
Involution Solid and Join Codes <i>Nataša Jonoska, Lila Kari, Kalpana Mahalingam</i>	192
Well-Founded Semantics for Boolean Grammars <i>Vassilis Kountouriotis, Christos Nomikos, Panos Rondogiannis</i>	203
Hierarchies of Tree Series Transformations Revisited <i>Andreas Maletti</i>	215
Bag Context Tree Grammars <i>Frank Drewes, Christine du Toit, Sigrid Ewert, Brink van der Merwe, Andries P.J. van der Walt</i>	226
Closure of Language Classes Under Bounded Duplication <i>Masami Ito, Peter Leupold, Kayoko Shikishima-Tsuji</i>	238
The Boolean Closure of Growing Context-Sensitive Languages <i>Tomasz Jurdziński</i>	248
Well Quasi Orders and the Shuffle Closure of Finite Sets <i>Flavio D'Alessandro, Gwénaél Richomme, Stefano Varricchio</i>	260
The Growth Ratio of Synchronous Rational Relations Is Unique <i>Olivier Carton</i>	270
On Critical Exponents in Fixed Points of Non-erasing Morphisms <i>Dalia Krieger</i>	280

P Systems with Proteins on Membranes and Membrane Division <i>Andrei Păun, Bianca Popa</i>	292
Computing by Only Observing <i>Matteo Cavaliere, Pierluigi Frisco, Hendrik Jan Hoogeboom</i>	304
A Decision Procedure for Reflexive Regular Splicing Languages <i>Paola Bonizzoni, Giancarlo Mauri</i>	315
Contextual Hypergraph Grammars – A New Approach to the Generation of Hypergraph Languages <i>Adrian-Horia Dediu, Renate Klempien-Hinrichs, Hans-Jörg Kreowski, Benedek Nagy</i>	327
End-Marked Maximal Depth-First Contextual Grammars <i>K. Lakshmanan</i>	339
Some Examples of Semi-rational DAG Languages <i>Jan Robert Menzel, Lutz Priese, Monika Schuth</i>	351
Finding Lower Bounds for Nondeterministic State Complexity Is Hard <i>Hermann Gruber, Markus Holzer</i>	363
Lowering Undecidability Bounds for Decision Questions in Matrices <i>Paul Bell, Igor Potapov</i>	375
Complexity of Degenerated Three Dimensional Billiard Words <i>J.-P. Borel</i>	386
Factorial Languages of Low Combinatorial Complexity <i>Arseny M. Shur</i>	397
Perfect Correspondences Between Dot-Depth and Polynomial-Time Hierarchy <i>Christian Glaßer, Stephen Travers, Klaus W. Wagner</i>	408
Language Equations with Complementations <i>Alexander Okhotin, Oksana Yakimova</i>	420
Synchronizing Automata with a Letter of Deficiency 2 <i>D.S. Ananichev, M.V. Volkov, Yu I. Zaks</i>	433

On Some Variations of Two-Way Probabilistic Finite Automata Models
 Bala Ravikumar 443

Author Index 455

Adding Nesting Structure to Words

Rajeev Alur¹ and P. Madhusudan²

¹ University of Pennsylvania, USA

² University of Illinois at Urbana-Champaign, USA

1 Introduction

We propose *nested words* to capture models where there is *both* a natural linear sequencing of positions and a hierarchically nested matching of positions. Such dual structure exists for executions of structured programs where there is a natural well-nested correspondence among entries to and exits from program components such as functions and procedures, and for XML documents where each open-tag is matched with a closing tag in a well-nested manner.

We define and study finite-state automata as acceptors of nested words. A nested-word automaton is similar to a classical finite-state word automaton, and reads the input from left to right according to the linear sequence. However, at a position with two predecessors, one due to linear sequencing and one due to a hierarchical nesting edge, the next state depends on states of the run at both these predecessors. The resulting class of *regular* languages of nested words has all the appealing theoretical properties that the class of classical regular word languages enjoys: deterministic nested word automata are as expressive as their nondeterministic counterparts; the class is closed under operations such as union, intersection, complementation, concatenation, and Kleene- $*$; decision problems such as membership, emptiness, language inclusion, and language equivalence are all decidable; definability in monadic second order logic of nested words corresponds exactly to finite-state recognizability; and finiteness of the congruence induced by a language of nested words is a necessary and sufficient condition for regularity.

The motivating application area for our results has been software verification. Given a sequential program P with stack-based control flow, the execution of P is modeled as a nested word with nesting edges from calls to returns. Specification of the program is given as a nested word automaton A , and verification corresponds to checking whether every nested word generated by P is accepted by A . Nested-word automata can express a variety of requirements such as stack-inspection properties, pre-post conditions, and interprocedural data-flow properties. If we were to model program executions as words, all of these properties are non-regular, and hence inexpressible in classical specification languages based on temporal logics, automata, and fixpoint calculi (recall that context-free languages cannot be used as specification languages due to nonclosure under intersection and undecidability of key decision problems such as language inclusion). In finite-state software model checking, the data variables in the program are abstracted into a set of boolean variables, and in that case, the set of nested words generated by the abstracted program is regular. This implies that algorithmic software verification is possible for all regular specifications of nested

words. We believe that the nested-word view will provide a unifying basis for the next generation of specification logics for program analysis, software model checking, and runtime monitoring. As explained in Section 3, another potential area of application is XML document processing.

1.1 Related Work

The finite automata on nested words that we study here have been motivated by our recent work on *visibly pushdown automata* [6]. A visibly pushdown automaton is one in which the input alphabet Σ is partitioned into three parts, $\langle \Sigma_c, \Sigma_i, \Sigma_r \rangle$ such that the automaton pushes exactly one symbol when reading symbols from Σ_c , pops one symbol from the stack when reading a symbol in Σ_r , and does not touch the stack when reading letters of Σ_i . The input word hence has an implicit nesting structure defined by matching occurrences of symbols in Σ_c with symbols in Σ_r . In nested words, this nesting is given explicitly, and this lets us define an automaton without a stack¹. We believe that nested words is a more appealing and simpler formulation of the insights in the theory of visibly pushdown languages. However, in terms of technical results, this paper only reformulates the corresponding results for visibly pushdown languages in [6].

Visibly pushdown languages are obviously related to Dyck languages, which is the class of languages with well-bracketed structure. The class of *parenthesis* languages studied by McNaughton comes closest to our notion of visibly pushdown languages [16]. A parenthesis language is one generated by a context free grammar where every production introduces a pair of parentheses that delimit the scope of the production. Viewing the nesting relation as that defined by the parentheses, parenthesis languages are a subclass of visibly pushdown languages. In [16, 11], it was shown that parenthesis languages are closed under union, intersection and difference, and that the equivalence problem for them is decidable. However, parenthesis languages are a strict subclass of visibly pushdown languages, and are not closed under Kleene-*

The class of visibly pushdown languages, was considered in papers related to parsing *input-driven languages* [22, 9]. Input-driven languages are precisely visibly pushdown languages (the stack operations are *driven* by the input). However, the papers considered only the membership problem for these languages (namely showing that membership is easier for these languages than for general context-free languages) and did not systematically study the *class* of languages defined by such automata.

2 Nested Words

2.1 Definition

A *nested relation* ν of width k , for $k \geq 0$, is a binary relation over $\{1, 2 \dots k\}$ such that (1) if $\nu(i, j)$ then $i < j$; (2) if $\nu(i, j)$ and $\nu(i, j')$ then $j = j'$, and if

¹ It is worth noting that most of the algorithms for inter-procedural program analysis and context-free reachability compute summary edges between control locations to capture the computation of the called procedure (see, for example [18]).

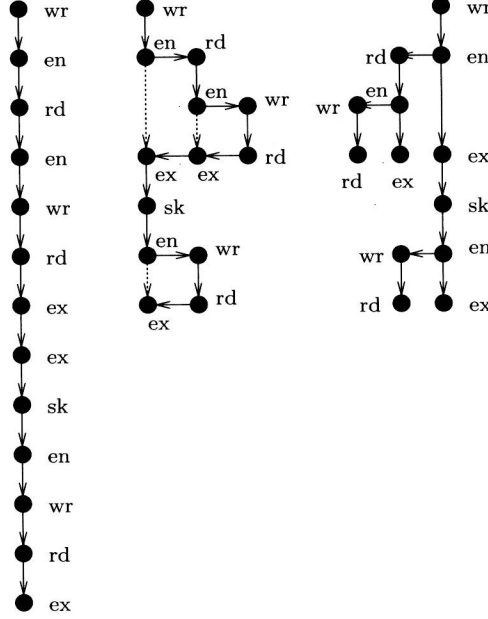


Fig. 1. Execution as a word, as a nested word, and as a tree

$\nu(i, j)$ and $\nu(i', j)$ then $i = i'$; (3) if $\nu(i, j)$ and $\nu(i', j')$ and $i < i'$ then either $j < i'$ or $j' < j$.

Let ν be a nested relation. When $\nu(i, j)$ holds, the position j is called a *return-successor* of the position i , and the position i is called a *call-predecessor* of the position j . Our definition requires that a position has at most one return-successor and at most one call-predecessor, and a position cannot have both a return-successor and a call-predecessor. A position is called a *call* position if it has a return successor, a *return* position if it has a call-predecessor, and an *internal* position otherwise.

A *nested word* nw over an alphabet Σ is a pair $(a_1 \dots a_k, \nu)$, for $k \geq 0$, such that a_i , for each $1 \leq i \leq k$, is a symbol in Σ , and ν is a nested relation of width k . Let us denote the set of nested words over Σ as $NW(\Sigma)$. A language of nested words over Σ is a subset of $NW(\Sigma)$.

2.2 Example: Program Executions as Nested Words

Execution of a program is typically modeled as a word over an alphabet Σ . The choice of Σ depends on the desired level of detail. As an example, suppose we are interested in tracking read/write accesses to a program variable x . The variable x may get redefined, for example, due to a declaration of a local variable within a called procedure, and we need to track the scope of these definitions. For simplicity, let's assume every change in context redefines the variable. Then,

we can choose the following set of symbols: rd to denote a read access to x , wr to denote a write access to x , en to denote beginning of a new scope (such as a call to a function or a procedure), ex to denote the ending of the current scope, and sk to denote all other actions of the program. Note that in any structured programming language, in a given execution, there is a natural nested matching of the symbols en and ex . Figure 1 shows a possible execution as a word as well as a nested word. The nesting edges are shown as dotted edges. A vertical path can be interpreted as a local path through a procedure. There is a natural connection between nested words and binary trees, and is also depicted in Figure 1. In this view, at a call node, the left subtree encodes the computation within the called procedure, while a path along the right children gives the local computation within a procedure.

In modeling the execution as a word, the matching between calls and returns is only implicit, and a pushdown automaton is needed to reconstruct the matching. The tree view makes the hierarchical structure explicit: every matching exit is a right-child of the corresponding entry node. However, this view loses linearity: the left and right subtrees of an entry node are disconnected, and (top-down) tree automata need nondeterminism to relate the properties of the subtrees². Our hypothesis is that the nested-word view is the most suitable view for program verification. In this view, a program will be a *generator* of nested words, and will be modeled as a language of nested words. For acceptors, linearity is used to obtain a left-to-right deterministic acceptor, while nesting is exploited to keep the acceptor finite state.

2.3 Operations on Nested Words

Analogous of a variety of operations on words and word languages can be defined for nested words and corresponding languages. We describe a few of the interesting ones here.

Given two nested words $nw_1 = (w_1, \nu_1)$ and $nw_2 = (w_2, \nu_2)$, of lengths k_1 and k_2 , respectively, the *concatenation* of nw_1 and nw_2 is the nested word $nw_1.nw_2 = (w_1.w_2, \nu)$ of length $k_1 + k_2$, where ν is the nested relation $\nu_1 \cup \{(k_1 + i, k_1 + j) \mid (i, j) \in \nu_2\}$. The concatenation extends to languages of nested words. The Kleene-* operation is defined as usual: if L is a language of nested words over Σ , then L^* is the set of nested words $nw_1.nw_2 \dots nw_i$, where $i \in \mathbb{N}$, and each $w_j \in L$.

Given a nested word $nw = (a_1 \dots a_k, \nu)$ of length k , its *reverse* is $nw^r = (a_k \dots a_1, \nu^r)$ where $\nu^r = \{(i, j) \mid (k + 1 - j, k + 1 - i) \in \nu\}$.

Finally, we define a notion of *insertion* for nested words. A *context* is a pair (nw, i) where nw is a nested word of length k , and $0 \leq i \leq k$. Given a context (nw, i) , for $nw = (a_1 \dots a_k, \nu)$, and a nested word nw' , with $nw' = (w', \nu')$, $(nw, i) \oplus nw'$ is the nested word obtained by inserting the

² It is worth mentioning that in program verification, trees are used for a different purpose: an execution tree encodes all possible executions of a program, and branching corresponds to the choice within the program. It is possible to define *nested trees* in which each path encodes a structured execution as a nested word [3].