Festschrift

LNCS 3895

Oded Goldreich
Arnold L. Rosenberg
Alan L. Selman (Eds.)

Theoretical Computer Science

Essays in Memory of Shimon Even

Springer

Oded Goldreich Arnold L. Rosenberg
Alan L. Selman (Eds.)

Theoretical Computer Science

Essays in Memory of Shimon Even

Springer

Volume Editors

Oded Goldreich
Weizmann Institute of Science
Department of Computer Science
Rehovot, Israel
E-mail: oded.goldreich@weizmann.ac.il

Arnold L. Rosenberg
University of Massachusetts Amherst
Department of Computer Science
Amherst, MA 01003, USA
E-mail: rsnbrg@cs.umass.edu

Alan L. Selman
University at Buffalo, The State University of New York
Department of Computer Science and Engineering
Buffalo, NY 14260-2000, USA
E-mail: selman@cse.buffalo.edu

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

Library of Congress Control Number: 2006922002

CR Subject Classification (1998): F.2.2, G.1.2, G.2.2, C.2.4, E.3

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

ISSN 0302-9743
ISBN-10 3-540-32880-7 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-32880-3 Springer Berlin Heidelberg New York

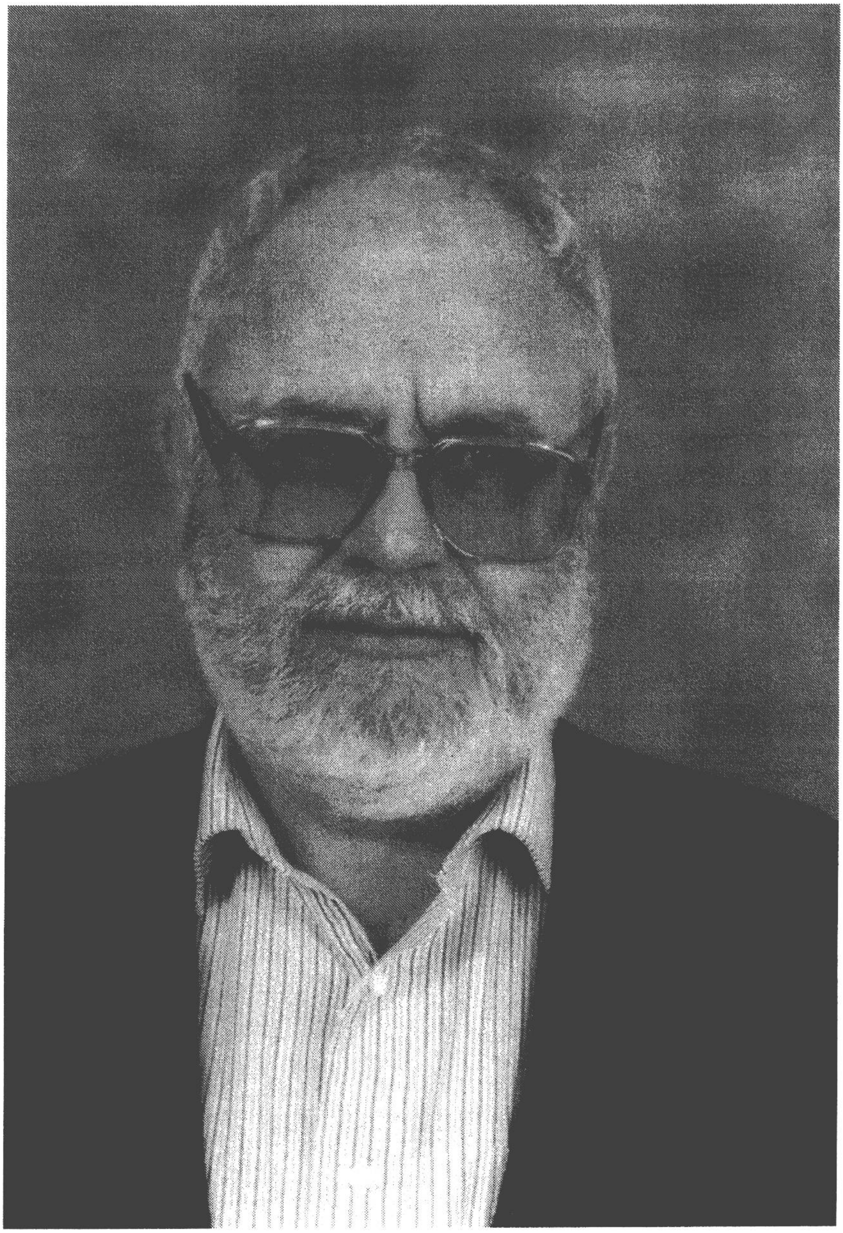
Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Boller Mediendesign
Printed on acid-free paper SPIN: 11685654 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 3895



Shimon Even (1935–2004)

Preface

On May 1, 2004, the world of theoretical computer science suffered a stunning loss: Shimon Even passed away. Few computer scientists have had as long, sustained, and influential a career as Shimon.

Shimon Even was born in Tel-Aviv in 1935. He received a B.Sc. in Electrical Engineering from the Technion in 1959, an M.A. in Mathematics from the University of Northern Carolina in 1961, and a Ph.D. in Applied Mathematics from Harvard University in 1963. He held positions at the Technion (1964–67 and 1974–2003), Harvard University (1967–69), the Weizmann Institute (1969–74), and the Tel-Aviv Academic College (2003-04). He visited many universities and research institutes, including Bell Laboratories, Boston University, Cornell, Duke, Lucent Technologies, MIT, Paderborn, Stanford, UC-Berkeley, USC and UT-Dallas.

Shimon Even played a major role in establishing computer science education in Israel and led the development of academic programs in two major institutions: the Weizmann Institute and the Technion. In 1969 he established at the Weizmann the first computer science education program in Israel, and led this program for five years. In 1974 he joined the newly formed computer science department at the Technion and shaped its academic development for several decades. These two academic programs turned out to have a lasting impact on the evolution of computer science in Israel.

Shimon Even was a superb teacher, and his courses deeply influenced many of the students attending them. His lectures, at numerous international workshops and schools, inspired a great number of students and researchers. His books, especially his celebrated *Graph Algorithms*, carried his educational message also to computer scientists who were not fortunate enough to meet him in person. As a mentor to aspiring researchers, Shimon was almost without peer, nurturing numerous junior researchers and advising many graduate students, who went on to have their own successful research careers.

Shimon Even was a pioneer in the areas of graph algorithms and cryptography, and his research contributions to these areas influenced the course of their development. Shimon was famous for not confining his interests to a few topics, but choosing rather to work in such diverse areas as switching and automata theory, coding theory, combinatorial algorithms, complexity theory, distributed computing, and circuit layout. In each of these areas, he produced high-quality, innovative research for more than four decades.

Shimon was the purest of pure theoreticians, following his nose toward research problems that were "the right" ones at the moment, not the faddish ones. His standards were impeccable, to the point where he would balk at employing any result whose proof he had not mastered himself. His integrity was unimpeachable: he would go to great lengths to defend any principle he believed in.

Shimon had a great passion for computer science as well as a great passion for truth. He valued simplicity, commitment to science, natural questions and carefully prepared expositions. By merely following his own way, Shimon influenced numerous researchers to adopt his passions and values. We hope that this is reflected in the current volume.

This volume contains research contributions and surveys by former students and close collaborators of Shimon. We are very pleased that Reuven Bar-Yehuda, Yefim Dinitz, Guy Even, Richard Karp, Ami Litman, Yehoshua Perl, Sergio Rajsbaum, Adi Shamir, and Yacov Yacobi agreed to send contributions. In accordance with Shimon's style and principles, the focus of these contributions is on addressing natural problems and being accessible to most researchers in theoretical computer science. The contributions are of three different types, reflecting three main scientific activities of Shimon: original research, technical surveys, and educational essays.

The Contributions

The contributions were written by former students and close collaborators of Shimon. In some cases the contributions are co-authored by researchers who were not fortunate enough to be close to Shimon or even to have met him in person. Below we comment on particular aspects of each contribution that we believe Shimon would have appreciated.

Original Research

Needless to say, everybody likes original research, and Shimon was no exception. We believe that Shimon would have been happy with the attempt to make these research contributions accessible to a wide range of researchers (rather than merely to experts in the area). In order to promote this goal, these contributions were reviewed both by experts and by non-experts.

- P. Fraigniaud, D. Ilcinkas, S. Rajsbaum and S. Tixeuil: *The Reduced Automata Technique for Graph Exploration Space Lower Bounds.* Shimon liked connections between areas, and the areas of graph algorithms and of automata theory were among his favorites.
- O. Goldreich: *Concurrent Zero-Knowledge with Timing, Revisited.* Shimon would have joked at Oded's tendency to write long papers.
- R.M. Karp: *Fair Bandwidth Allocation Without Per-Flow State.* Shimon would have like the fact that the starting point of this work is a practical problem, and that it proceeds by distilling a clear computational problem and resolving it optimally.
- R.M. Karp, T. Nierhoff and T. Tantau: *Optimal Flow Distribution Among Multiple Channels with Unknown Capacities.* This paper has the same flavor as the previous one, and Shimon would have liked it for the very same reason.

- A. Litman: *Parceling the Butterfly and the Batcher Sorting Network.* Shimon would have liked the attempt to present a new complexity measure that better reflects the actual cost of implementations.
- X. Zhou, J. Geller, Y. Perl, and M. Halper: *An Application Intersection Marketing Ontology.* Shimon would have liked the fact that simple insights of graph theory are used for a problem that is very remote from graph theory.
- R.L. Rivest, A. Shamir and Y. Tauman: *How to Leak a Secret: Theory and Applications of Ring Signatures.* Shimon would have like the natural ("daily") problem addressed in this paper as well as the elegant solution provided to it.
- O. Yacobi with Y. Yacobi: *A New Related Message Attack on RSA.* Shimon would have enjoyed seeing a father and son work together.

Technical Surveys

Shimon valued the willingness to take a step back, look at what was done (from a wider perspective), and provide a better perspective on it. We thus believe that he would have been happy to be commemorated by a volume that contains a fair number of surveys.

- R. Bar-Yehuda and D. Rawitz: *A Tale of Two Methods.* Shimon liked stories, and he also liked the techniques surveyed here. Furthermore, he would have been excited to learn that these two techniques are in some sense two sides of the same coin.
- Y. Dinitz: *Dinitz' Algorithm: The Original Version and Even's Version.* Shimon is reported to have tremendously enjoyed Dinitz's lecture that served as a skeleton to this survey.
- C. Glaßer, A.L. Selman, and L. Zhang: *Survey of Disjoint NP-pairs and Relations to Propositional Proof Systems.* This survey focuses on one of the applications of promise problems, which was certainly unexpected in 1984 when Shimon Even, together with Alan Selman and Yacov Yacobi, introduced this notion.
- O. Goldreich: *On Promise Problems.* This survey traces the numerous and diverse applications that the notion of promise problems found in the two decades that have elapsed since the invention of the notion.
- G. Malewicz and A.L. Rosenberg: *A Pebble Game for Internet-Based Computing.* Shimon liked elegant models, and would have been interested to see pebble games used to model an Internet-age problem.

Educational Essays

Shimon liked opinionated discussions and valued independent opinions that challenge traditional conventions. So we are sure he would have enjoyed reading these essays, and we regret that we cannot have his reaction to them.

- G. Even: *On Teaching Fast Adder Designs: Revisiting Ladner & Fischer.* Shimon would have been very proud of this insightful and opinionated exposition of hardware implementations of the most basic computational task.
- O. Goldreich: *On Teaching the Basics of Complexity Theory.* Shimon would have appreciated the attempt to present the basics of complexity theory in a way that appeals to the naive student.
- A.L. Rosenberg: *State.* Shimon would have supported the campaign, launched in this essay, in favor of the Myhill-Nerode Theorem.

December 2005

Oded Goldreich (Weizmann Institute of Science)
Arnold L. Rosenberg (University of Massachusetts Amherst)
Alan L. Selman (University at Buffalo)

Lecture Notes in Computer Science

For information about Vols. 1–3804

please contact your bookseller or Springer

Vol. 3901: P.M. Hill (Ed.), Logic Based Program Synthesis and Transformation. X, 179 pages. 2006.

Vol. 3899: S. Frintrop, VOCUS: A Visual Attention System for Object Detection and Goal-Directed Search. XIV, 216 pages. 2006. (Sublibrary LNAI).

Vol. 3895: O. Goldreich, A.L. Rosenberg, A.L. Selman (Eds.), Theoretical Computer Science. XII, 399 pages. 2006.

Vol. 3894: W. Grass, B. Sick, K. Waldschmidt (Eds.), Architecture of Computing Systems - ARCS 2006. XII, 496 pages. 2006.

Vol. 3890: S.G. Thompson, R. Ghanea-Hercock (Eds.), Defence Applications of Multi-Agent Systems. XII, 141 pages. 2006. (Sublibrary LNAI).

Vol. 3889: J. Rosca, D. Erdogmus, J.C. Príncipe, S. Haykin (Eds.), Independent Component Analysis and Blind Signal Separation. XXI, 980 pages. 2006.

Vol. 3888: D. Draheim, G. Weber (Eds.), Trends in Enterprise Application Architecture. IX, 145 pages. 2006.

Vol. 3887: J. Correa, A. Hevia, M. Kiwi (Eds.), LATIN 2006: Theoretical Informatics. XVI, 814 pages. 2006.

Vol. 3886: E.G. Bremer, J. Hakenberg, E.-H.(S.) Han, D. Berrar, W. Dubitzky (Eds.), Knowledge Discovery in Life Science Literature. XIV, 147 pages. 2006. (Sublibrary LNBI).

Vol. 3885: V. Torra, Y. Narukawa, A. Valls, J. Domingo-Ferrer (Eds.), Modeling Decisions for Artificial Intelligence. XII, 374 pages. 2006. (Sublibrary LNAI).

Vol. 3884: B. Durand, W. Thomas (Eds.), STACS 2006. XIV, 714 pages. 2006.

Vol. 3881: S. Gibet, N. Courty, J.-F. Kamp (Eds.), Gesture in Human-Computer Interaction and Simulation. XIII, 344 pages. 2006. (Sublibrary LNAI).

Vol. 3880: A. Rashid, M. Aksit (Eds.), Transactions on Aspect-Oriented Software Development I. IX, 335 pages. 2006.

Vol. 3879: T. Erlebach, G. Persinao (Eds.), Approximation and Online Algorithms. X, 349 pages. 2006.

Vol. 3878: A. Gelbukh (Ed.), Computational Linguistics and Intelligent Text Processing. XVII, 589 pages. 2006.

Vol. 3877: M. Detyniecki, J.M. Jose, A. Nürnberger, C. J. ‘. van Rijsbergen (Eds.), Adaptive Multimedia Retrieval: User, Context, and Feedback. XI, 279 pages. 2006.

Vol. 3876: S. Halevi, T. Rabin (Eds.), Theory of Cryptography. XI, 617 pages. 2006.

Vol. 3875: S. Ur, E. Bin, Y. Wolfsthal (Eds.), Haifa Verification Conference. X, 265 pages. 2006.

Vol. 3874: R. Missaoui, J. Schmidt (Eds.), Formal Concept Analysis. X, 309 pages. 2006. (Sublibrary LNAI).

Vol. 3873: L. Maicher, J. Park (Eds.), Charting the Topic Maps Research and Applications Landscape. VIII, 281 pages. 2006. (Sublibrary LNAI).

Vol. 3872: H. Bunke, A. L. Spitz (Eds.), Document Analysis Systems VII. XIII, 630 pages. 2006.

Vol. 3870: S. Spaccapietra, P. Atzeni, W.W. Chu, T. Catarci, K.P. Sycara (Eds.), Journal on Data Semantics V. XIII, 237 pages. 2006.

Vol. 3869: S. Renals, S. Bengio (Eds.), Machine Learning for Multimodal Interaction. XIII, 490 pages. 2006.

Vol. 3868: K. Römer, H. Karl, F. Mattern (Eds.), Wireless Sensor Networks. XI, 342 pages. 2006.

Vol. 3866: T. Dimitrakos, F. Martinelli, P.Y.A. Ryan, S. Schneider (Eds.), Formal Aspects in Security and Trust. X, 259 pages. 2006.

Vol. 3865: W. Shen, K.-M. Chao, Z. Lin, J.-P.A. Barthès (Eds.), Computer Supported Cooperative Work in Design II. XII, 359 pages. 2006.

Vol. 3863: M. Kohlhase (Ed.), Mathematical Knowledge Management. XI, 405 pages. 2006. (Sublibrary LNAI).

Vol. 3862: R.H. Bordini, M. Dastani, J. Dix, A.E.F. Seghrouchni (Eds.), Programming Multi-Agent Systems. XIV, 267 pages. 2006. (Sublibrary LNAI).

Vol. 3861: J. Dix, S.J. Hegner (Eds.), Foundations of Information and Knowledge Systems. X, 331 pages. 2006.

Vol. 3860: D. Pointcheval (Ed.), Topics in Cryptology – CT-RSA 2006. XI, 365 pages. 2006.

Vol. 3858: A. Valdes, D. Zamboni (Eds.), Recent Advances in Intrusion Detection. X, 351 pages. 2006.

Vol. 3857: M.P.C. Fossorier, H. Imai, S. Lin, A. Poli (Eds.), Applied Algebra, Algebraic Algorithms and Error-Correcting Codes. XI, 350 pages. 2006.

Vol. 3855: E. A. Emerson, K.S. Namjoshi (Eds.), Verification, Model Checking, and Abstract Interpretation. XI, 443 pages. 2005.

Vol. 3853: A.J. Ijspeert, T. Masuzawa, S. Kusumoto (Eds.), Biologically Inspired Approaches to Advanced Information Technology. XIV, 388 pages. 2006.

Vol. 3852: P.J. Narayanan, S.K. Nayar, H.-Y. Shum (Eds.), Computer Vision – ACCV 2006, Part II. XXXI, 977 pages. 2006.

Vol. 3851: P.J. Narayanan, S.K. Nayar, H.-Y. Shum (Eds.), Computer Vision – ACCV 2006, Part I. XXXI, 973 pages. 2006.

Vol. 3850: R. Freund, G. Păun, G. Rozenberg, A. Salomaa (Eds.), Membrane Computing. IX, 371 pages. 2006.

Vol. 3849: I. Bloch, A. Petrosino, A.G.B. Tettamanzi (Eds.), Fuzzy Logic and Applications. XIV, 438 pages. 2006. (Sublibrary LNAI).

Vol. 3848: J.-F. Boulicaut, L. De Raedt, H. Mannila (Eds.), Constraint-Based Mining and Inductive Databases. X, 401 pages. 2006. (Sublibrary LNAI).

Vol. 3847: K.P. Jantke, A. Lunzer, N. Spyratos, Y. Tanaka (Eds.), Federation over the Web. X, 215 pages. 2006. (Sublibrary LNAI).

Vol. 3846: H. J. van den Herik, Y. Björnsson, N.S. Netanyahu (Eds.), Computers and Games. XIV, 333 pages. 2006.

Vol. 3845: J. Farré, I. Litovsky, S. Schmitz (Eds.), Implementation and Application of Automata. XIII, 360 pages. 2006.

Vol. 3844: J.-M. Bruel (Ed.), Satellite Events at the MoDELS 2005 Conference. XIII, 360 pages. 2006.

Vol. 3843: P. Healy, N.S. Nikolov (Eds.), Graph Drawing. XVII, 536 pages. 2006.

Vol. 3842: H.T. Shen, J. Li, M. Li, J. Ni, W. Wang (Eds.), Advanced Web and Network Technologies, and Applications. XXVII, 1057 pages. 2006.

Vol. 3841: X. Zhou, J. Li, H.T. Shen, M. Kitsuregawa, Y. Zhang (Eds.), Frontiers of WWW Research and Development - APWeb 2006. XXIV, 1223 pages. 2006.

Vol. 3840: M. Li, B. Boehm, L.J. Osterweil (Eds.), Unifying the Software Process Spectrum. XVI, 522 pages. 2006.

Vol. 3839: J.-C. Filliâtre, C. Paulin-Mohring, B. Werner (Eds.), Types for Proofs and Programs. VIII, 275 pages. 2006.

Vol. 3838: A. Middeldorp, V. van Oostrom, F. van Raamsdonk, R. de Vrijer (Eds.), Processes, Terms and Cycles: Steps on the Road to Infinity. XVIII, 639 pages. 2005.

Vol. 3837: K. Cho, P. Jacquet (Eds.), Technologies for Advanced Heterogeneous Networks. IX, 307 pages. 2005.

Vol. 3836: J.-M. Pierson (Ed.), Data Management in Grids. X, 143 pages. 2006.

Vol. 3835: G. Sutcliffe, A. Voronkov (Eds.), Logic for Programming, Artificial Intelligence, and Reasoning. XIV, 744 pages. 2005. (Sublibrary LNAI).

Vol. 3834: D.G. Feitelson, E. Frachtenberg, L. Rudolph, U. Schwiegelshohn (Eds.), Job Scheduling Strategies for Parallel Processing. VIII, 283 pages. 2005.

Vol. 3833: K.-J. Li, C. Vangenot (Eds.), Web and Wireless Geographical Information Systems. XI, 309 pages. 2005.

Vol. 3832: D. Zhang, A.K. Jain (Eds.), Advances in Biometrics. XX, 796 pages. 2005.

Vol. 3831: J. Wiedermann, G. Tel, J. Pokorný, M. Bieliková, J. Štuller (Eds.), SOFSEM 2006: Theory and Practice of Computer Science. XV, 576 pages. 2006.

Vol. 3830: D. Weyns, H. V.D. Parunak, F. Michel (Eds.), Environments for Multi-Agent Systems II. VIII, 291 pages. 2006. (Sublibrary LNAI).

Vol. 3829: P. Pettersson, W. Yi (Eds.), Formal Modeling and Analysis of Timed Systems. IX, 305 pages. 2005.

Vol. 3828: X. Deng, Y. Ye (Eds.), Internet and Network Economics. XVII, 1106 pages. 2005.

Vol. 3827: X. Deng, D.-Z. Du (Eds.), Algorithms and Computation. XX, 1190 pages. 2005.

Vol. 3826: B. Benatallah, F. Casati, P. Traverso (Eds.), Service-Oriented Computing - ICSOC 2005. XVIII, 597 pages. 2005.

Vol. 3824: L.T. Yang, M. Amamiya, Z. Liu, M. Guo, F.J. Rammig (Eds.), Embedded and Ubiquitous Computing – EUC 2005. XXIII, 1204 pages. 2005.

Vol. 3823: T. Enokido, L. Yan, B. Xiao, D. Kim, Y. Dai, L.T. Yang (Eds.), Embedded and Ubiquitous Computing – EUC 2005 Workshops. XXXII, 1317 pages. 2005.

Vol. 3822: D. Feng, D. Lin, M. Yung (Eds.), Information Security and Cryptology. XII, 420 pages. 2005.

Vol. 3821: R. Ramanujam, S. Sen (Eds.), FSTTCS 2005: Foundations of Software Technology and Theoretical Computer Science. XIV, 566 pages. 2005.

Vol. 3820: L.T. Yang, X.-s. Zhou, W. Zhao, Z. Wu, Y. Zhu, M. Lin (Eds.), Embedded Software and Systems. XXVIII, 779 pages. 2005.

Vol. 3819: P. Van Hentenryck (Ed.), Practical Aspects of Declarative Languages. X, 231 pages. 2005.

Vol. 3818: S. Grumbach, L. Sui, V. Vianu (Eds.), Advances in Computer Science – ASIAN 2005. XIII, 294 pages. 2005.

Vol. 3817: M. Faundez-Zanuy, L. Janer, A. Esposito, A. Satue-Villar, J. Roure, V. Espinosa-Duro (Eds.), Nonlinear Analyses and Algorithms for Speech Processing. XII, 380 pages. 2006. (Sublibrary LNAI).

Vol. 3816: G. Chakraborty (Ed.), Distributed Computing and Internet Technology. XXI, 606 pages. 2005.

Vol. 3815: E.A. Fox, E.J. Neuhold, P. Premsmit, V. Wuwongse (Eds.), Digital Libraries: Implementing Strategies and Sharing Experiences. XVII, 529 pages. 2005.

Vol. 3814: M. Maybury, O. Stock, W. Wahlster (Eds.), Intelligent Technologies for Interactive Entertainment. XV, 342 pages. 2005. (Sublibrary LNAI).

Vol. 3813: R. Molva, G. Tsudik, D. Westhoff (Eds.), Security and Privacy in Ad-hoc and Sensor Networks. VIII, 219 pages. 2005.

Vol. 3812: C. Bussler, A. Haller (Eds.), Business Process Management Workshops. XIII, 520 pages. 2006.

Vol. 3811: C. Bussler, M.-C. Shan (Eds.), Technologies for E-Services. VIII, 127 pages. 2006.

Vol. 3810: Y.G. Desmedt, H. Wang, Y. Mu, Y. Li (Eds.), Cryptology and Network Security. XI, 349 pages. 2005.

Vol. 3809: S. Zhang, R. Jarvis (Eds.), AI 2005: Advances in Artificial Intelligence. XXVII, 1344 pages. 2005. (Sublibrary LNAI).

Vol. 3808: C. Bento, A. Cardoso, G. Dias (Eds.), Progress in Artificial Intelligence. XVIII, 704 pages. 2005. (Sublibrary LNAI).

Vol. 3807: M. Dean, Y. Guo, W. Jun, R. Kaschek, S. Krishnaswamy, Z. Pan, Q.Z. Sheng (Eds.), Web Information Systems Engineering – WISE 2005 Workshops. XV, 275 pages. 2005.

Vol. 3806: A.H. H. Ngu, M. Kitsuregawa, E.J. Neuhold, J.-Y. Chung, Q.Z. Sheng (Eds.), Web Information Systems Engineering – WISE 2005. XXI, 771 pages. 2005.

Vol. 3805: G. Subsol (Ed.), Virtual Storytelling. XII, 289 pages. 2005.

¥461.00元

Table of Contents

The Reduced Automata Technique for Graph Exploration Space Lower Bounds 1
Pierre Fraigniaud, David Ilcinkas, Sergio Rajsbaum, Sébastien Tixeuil

Concurrent Zero-Knowledge with Timing, Revisited 27
Oded Goldreich

Fair Bandwidth Allocation Without Per-Flow State 88
Richard M. Karp

Optimal Flow Distribution Among Multiple Channels with Unknown Capacities 111
Richard Karp, Till Nierhoff, Till Tantau

Parceling the Butterfly and the Batcher Sorting Network 129
Ami Litman

An Application Intersection Marketing Ontology 143
Xuan Zhou, James Geller, Yehoshua Perl, Michael Halper

How to Leak a Secret: Theory and Applications of Ring Signatures 164
Ronald L. Rivest, Adi Shamir, Yael Tauman

A New Related Message Attack on RSA 187
Oded Yacobi, Yacov Yacobi

A Tale of Two Methods 196
Reuven Bar-Yehuda, Dror Rawitz

Dinitz' Algorithm: The Original Version and Even's Version 218
Yefim Dinitz

Survey of Disjoint NP-pairs and Relations to Propositional Proof Systems 241
Christian Glaßer, Alan L. Selman, Liyu Zhang

On Promise Problems: A Survey 254
Oded Goldreich

A Pebble Game for Internet-Based Computing 291
Grzegorz Malewicz, Arnold L. Rosenberg

On Teaching Fast Adder Designs: Revisiting Ladner & Fischer 313
Guy Even

On Teaching the Basics of Complexity Theory 348
Oded Goldreich

State 375
Arnold L. Rosenberg

Author Index 399

The Reduced Automata Technique for Graph Exploration Space Lower Bounds*

Pierre Fraigniaud[1] **, David Ilcinkas[2] **, Sergio Rajsbaum[3] ***, and
Sébastien Tixeuil[4] †

[1] CNRS, LRI, Université Paris-Sud, France
pierre@lri.fr
[2] LRI, Université Paris-Sud, France
ilcinkas@lri.fr
[3] Instituto de Matemáticas, Univ. Nacional Autónoma de México, Mexico
rajsbaum@math.unam.mx
[4] LRI & INRIA, Université Paris-Sud, France
tixeuil@lri.fr

Abstract. We consider the task of exploring graphs with anonymous nodes by a team of non-cooperative robots, modeled as finite automata. For exploration to be completed, each edge of the graph has to be traversed by at least one robot. In this paper, the robots have no a priori knowledge of the topology of the graph, nor of its size, and we are interested in the amount of memory the robots need to accomplish exploration, We introduce the so-called *reduced automata technique*, and we show how to use this technique for deriving several space lower bounds for exploration. Informally speaking, the reduced automata technique consists in reducing a robot to a simpler form that preserves its "core" behavior on some graphs. Using this technique, we first show that any set of $q \geq 1$ non-cooperative robots, requires $\Omega(\log(\frac{n}{q}))$ memory bits to explore all n-node graphs. The proof implies that, for any set of q K-state robots, there exists a graph of size $O(qK)$ that no robot of this set can explore, which improves the $O(K^{O(q)})$ bound by Rollik (1980). Our main result is an application of this latter result, concerning *terminating* graph exploration with one robot, i.e., in which the robot is requested to stop after completing exploration. For this task, the robot is provided with a pebble, that it can use to mark nodes (without such a marker, even terminating exploration of cycles cannot be achieved). We prove that terminating exploration requires $\Omega(\log n)$ bits of memory for a robot achieving this task in all n-node graphs.

* A preliminary version of this paper appears in the proceedings of the 12th International Colloquium on Structural Information and Communication Complexity (SIROCCO), Mont Saint-Michel, France, May 24-26, 2005, as part of [13].

** Supported by the INRIA project "Grand Large", and the projects "PairAPair" of the ACI "Masses de Données", and "FRAGILE" of the ACI "Sécurité et Informatique".

*** Supported by LAFMI and PAPIIT projects. Part of this work was done while visiting LRI, Univ. Paris Sud, Orsay.

† Supported by the INRIA project "Grand Large". Additional support from the project "FRAGILE" of the ACI "Sécurité et Informatique".

O. Goldreich et al. (Eds.): Shimon Even Festschrift, LNCS 3895, pp. 1–26, 2006.

1 Introduction

The problem of exploring an unknown environment occurs in a variety of situations, like robot navigation, network maintenance, resource discovery, and WWW search. In these situations the entities performing exploration can be either a physical mobile device or a software agent. In this paper, we restrict our attention to the case where the environment in which the mobile entities are moving is modeled as a graph. At an abstract level, graph exploration is the task where one or more mobile entities, called *robots* in this paper, are trying to collectively traverse every edge of a graph. In addition to the aforementioned applications, graph exploration is important due to its strong relation to complexity theory, and in particular to the undirected *st*-connectivity (USTCON) problem (cf., e.g., [6]). Given an undirected graph G and two vertices s and t, the USTCON problem is to decide whether s and t are in the same connected component of G. The directed version of the problem is denoted STCON. It is known that STCON is complete for NL, the class of non-deterministic log-space solvable problems. Whether USTCON is complete for L, the class of problems solvable by deterministic log-space algorithms, has been a challenging open problem for quite a long time, and it is only very recently that Reingold proved that USTCON is indeed complete for L [15]. Note that the existence of a finite set of finite-state automata able to explore all graphs would have put USTCON in L, and proving or disproving this existence had therefore motivated quite a long sequence of studies. Cook and Rackoff [6] eventually proved that even a more powerful machine, called JAG, for "Jumping Automaton for Graphs", cannot explore all graphs (a JAG is a finite set of globally cooperative finite-state automata enhanced with the ability, for every automaton, to "jump" from its current position to any node occupied by another automaton). Since this latter result, the exploration graph problem is focussing on determining the space complexity of robots able to explore all graphs.

As far as upper bounds in concerned, Reingold showed in [15] that his log-space algorithm for USTCON implies the existence of log-space constructible *universal exploration sequences* (UXS) of polynomial length. Roughly speaking, a UXS [14] is a sequence of integers that (1) tell a robot how to move from node to node in a graph (the exit port at the kth step of the traversal is obtained by adding the kth integer of the UXS to the entry port), and (2) guarantee to explore every node of a graph of appropriate size (a UXS is defined for a given size, and a given degree). Rephrasing this latter result, there is a $O(\log n)$-space robot that explores all the graphs of size n. The extend to which this bound can be decreased by using a set of $q > 1$ cooperative robots remains open. Also, the question of the existence of log-space constructible *universal traversal sequences* (UTS) [1] remains open (a UTS is a sequence of port-numbers so that the output port at the kth step of the traversal is the kth element of the sequence).

As far as lower bounds are concerned, most papers are dealing with the design of small *traps* for arbitrary teams of robots, i.e., small graphs that no robot of the team can explore. (Formally, a trap consists of a graph and a node from where the robots start the exploration.) The first trap for a finite state robot is generally attributed to Budach [5] (the trap is actually a planar graph). The trap constructed by Budach is however of large size, and a much smaller trap was described in [12] which proved that, for any K-state robot, there exists a trap of at most $K + 1$ nodes. In [16], Rollik proved that no finite set of finite locally-cooperative automata, i.e., automata that exchange information only when they meet at a node, can explore all graphs. In the proof of this result, the author uses as a tool a trap for a set of q non-cooperative K-state robots (such robots may have different transition functions, hence they will follow different paths in the explored graph). This latter trap is of size $O(K^{O(q)})$ nodes. Rollik's trap for cooperative robots is even larger: $\tilde{O}(K^{K^{\cdot^{\cdot^{\cdot^{K}}}}})$ nodes, with $2q + 1$ levels of exponentials where the $\tilde{O}$ notation hides logarithmic factors. In this paper, we present a new lower bound technique for graph exploration, called *reduced automata technique*. Roughly, this technique consists in reducing a robot to a simpler form that preserves its "core" behavior on some graphs: except for some easily described closed paths, the reduced robot follows the path of the original robot, on any such graph.

The interested reader can find other pointers to the literature in, e.g., [3–5, 7, 8, 12]. To complete the picture, and before describing our results in more details, let us point out that Shimon Even, whom this book is dedicated to, was interested in graph exploration problems early on in his career. In particular, in his 1976 seminal paper with Tarjan [11], he presented a way of numbering nodes during a DFS traversal that proved to be useful in many algorithms. In collaboration with A. Litman and P. Winkler [10], he then studied traversal in directed networks. With G. Itkis and S. Rajsbaum [9], he described a traversal strategy for undirected graphs that constructs a subgraph with good connectivity but few edges. And recently, in collaboration with S. Bhatt, D. Greenberg, and R. Tayard [2], he studied the problem of using a robot as simple as possible (with access to some local memory stored in the vertices) to find an Eulerian cycle in mazes and graphs.

1.1 Problem Settings

As in [6, 16], we are interested in exploration of undirected graphs where nodes are not uniquely labeled. Note that, besides the theoretical interest of understanding when or at what cost such graphs can be explored, the unlabeled-node setting can occur in practice, due to, e.g., privacy concerns, limited capabilities of the robots, or simply anonymous edge intersections. The robots, modeled as a deterministic automata, can however identify the edges incident to a node through unique port labels, from 1 to the degree of the node. We consider two types of exploration:

- Perpetual exploration, in which the task of the robots is to, eventually, traverse all edges.
- Terminating exploration, in which the robots, after completing exploration, must eventually stop.

In acyclic graphs, terminating exploration is strictly more difficult than perpetual exploration. In particular, it is shown in [7] that terminating exploration in n-node bounded degree trees requires a robot with memory size $\Omega(\log\log\log n)$, whereas perpetual exploration is possible with $O(1)$ bits. In arbitrary graphs, terminating exploration cannot be achieved. Indeed, it is easy to see that a robot can traverse all edges of some graphs, say a cycle, but that it cannot recognize when it has visited a node twice, because nodes are not uniquely labeled. That is, there are graphs that a robot can explore perpetually, but it can never stops. Thus, as in previous work in this setting, e.g., [3,4,8], we assume that, for terminating exploration, robots can mark nodes: a robot can drop a pebble in a node and later identify it and pick it up.

Following the common conventions in the literature, the robots aiming at performing perpetual exploration are not given pebbles, whereas robots aiming at performing terminating exploration are given one or more pebbles. As a consequence, the two problems becomes incomparable. Indeed, terminating exploration is more demanding than perpetual exploration, but the "machines" designed for these two tasks do not have the same power.

A *team* of robots is a set of deterministic automata with possibly different transition functions, all starting from the same starting point. When sets or teams of robots are considered, the robots of a team can communicate in various manners. Four cases are considered in the literature:

- Non-cooperative robots: the robots are oblivious of each other, and each of them acts independently from the others.
- Locally cooperative robots: robots meeting at a node can exchange information, including their identities and their current states.
- Globally cooperative robots: the robots are perpetually aware of the states of the others, of whether they meet and who they meet, and of the degrees of the nodes occupied by the robots.
- Jumping Automaton: the robots are globally cooperative, and any robot is able to jump from the node it is currently occupying to a node currently occupied by any other robot.

In this paper, we restrict our attention to the two weakest models: non-cooperative robots, and locally cooperative robots.

1.2 Our Results

In this paper, we present a new lower bound technique for graph exploration, called *reduced automata technique*. Based on this technique, the lower bounds presented in this paper are obtained as follows. Assume a set of q robots is given. Then construct the smallest possible graph, called a *trap* for this set of robots, such that if the robots are placed in some specified nodes of the graphs, then there is at least one edge that is not traversed by any of the robots. If the q robots have K states each, and the trap has $f_q(K)$ nodes, then the space lower bound for a set of q robots exploring all n-node graphs is $\Omega(\log f_q^{-1}(n))$ bits.