COUNCIL OF THE EUROPEAN ASSOCIATION FOR THEORETICAL COMPUTER SCIENCE

BOARD:

PRESIDENT: M. NIELSEN Denmark
VICE PRESIDENTS: J. VAN LEEUWEN The Netherlands
P. SPIRAKIS Greece
TREASURER: D. JANSSENS Belgium
SECRETARY: B. ROVAN Slovakia
BULLETIN EDITOR: G. ROZENBERG The Netherlands
TCS EDITORS: G. AUZIELLO Italy
M.W. MISLOWE U.S.A.
D. SANNELLA United Kingdom
MONOGRAPHS EDITORS: W. BRAUER Germany
G. ROZENBERG The Netherlands
A. SALOMAA Finland
WEB COORDINATOR: M. NIELSEN Denmark

OTHER COUNCIL MEMBERS

S. ARNBORG Sweden J. KARHUMÄKI Finland
R. BAEZA-YATES Chile K. MEHLHORN Germany
C. CALUDE New Zealand M. NIVAT France
P. DEGANO Italy G. PAUN Romania
M. DEZANI-CIANCAGLINI Italy J. ROLIM Switzerland
J. DÍAZ Spain J. SGALL Czech Republic
H. EHRIG Germany P.S. THIAGARAJAN India
J. ESPARZA United Kingdom W. THOMAS Germany
Z. FULÖP Hungary D. WAGNER Germany
H. GABOW U.S.A. O. WATANABE Japan
A. GIBBONS United Kingdom E. WELZL Switzerland
R. GORRIERI Italy J. WIEDERMANN Czech Republic
M. HENNESSY United Kingdom G. WOEGERINGER Austria
K. IWAMA Japan S. YU Canada
J.-P. JOUANNAUD France

PAST PRESIDENTS:

M. NIVAT France (1972-1977)
M. PATERSON England (1977-1979)
A. SALOMAA Finland (1979-1985)
G. ROZENBERG The Netherlands (1985-1994)
W. BRAUER Germany (1994-1997)
J. DÍAZ Spain (1997-2002)
EATCS Council Members

e-mail addresses

S. Arnborg ...................................................... stefan@nada.kth.se
G. Ausiello ........................................ ausiello@disparcs.ing.uniroma1.it
R. Baeza-Yates ........................................... rbaeza@dcc.uchile.cl
W. Brauer ........................................ brauer@informatik.tu-muenchen.de
C. Calude .................................................. cristian@cs.auckland.ac.nz
P. Degano .................................................. degano@di.unipi.it
M. Dezani-Ciancaglini .................................. dezani@di.unito.it
J. Diaz ...................................................... diaz@lsi.upc.es
H. Ehrig ..................................................... ehrig@cs.tu-berlin.de
J. Esparza ................................................ j.ev@dcs.ed.ac.uk
Z. Fülöp .................................................... fulop@inf.u-szeged.hu
H. Gabow .................................................. hal@cs.colorado.edu
A. Gibbons ............................................... amg@dcs.kcl.ac.uk
R. Gorrieri ............................................... gorrieri@cs.unibo.it
M. Hennessy .............................................. matthewh@cogs.sussex.ac.uk
K. Iwama .................................................. iwama@kuis.kyoto-u.ac.jp
D. Janssens .............................................. dirk.janssens@ua.ac.be
J.-P. Jouannaud .......................................... jouannau@iri.fr
J. Karhumäki ............................................. karhumak@cs.utu.fi
J. van Leeuwen ......................................... jan@cs.uu.nl
K. Mehlhorn ............................................. mehlhorn@mpi-sb.mpg.de
M.W. Mislove ........................................... mwm@math.tulane.edu
M. Nielsen ............................................... mnielsen@daimi.aau.dk
M. Nivat ................................................ tcsmn@club-internet.fr
G. Paun ................................................ gpaun@imar.ro
J. Rolim ..................................................... jose.rolim@cui.unige.ch
G. Rozenberg .......................................... rozenber@iaics.nl
A. Salomaa ............................................... asalomaa@utu.fi
D. Sannella ............................................... dts@dcs.ed.ac.uk
J. Sgall ...................................................... sgall@math.cas.cz
P. Spirakis ............................................... spirakis@cti.gr
P.S. Thiagarajan ........................................ pst@ssf.emet.in
W. Thomas ................................................. thomas@informatik.rwth-aachen.de
D. Wagner ............................................... dorothea.wagner@uni-konstanz.de
O. Watanabe .............................................. watanabe@is.titech.ac.jp
J. Wiedermann ......................................... wiedere@uivt1.cas.cz
E. Welzl ................................................ emo@inf.ethz.ch
G. Woeginger .......................................... gwoegi@opt.math.tu-graz.ac.at
S. Yu ........................................................ syu@csd.uwo.ca
# TABLE OF CONTENTS

1. EATCS Matters
   - Letter from the President ........................................ 1
   - Letter from the Bulletin Editor ................................. 2
   - The Japanese Chapter (by O. Watanabe) ....................... 3
     - Nippon from South to North – NII (by N. Arai) .......... 4

2. ICALP 2003 ................................................................. 6

3. Gödel Prize
   - 2003 GP awarded to Yoav Freund and Robert Schapire ...... 14

4. Institutional Sponsors .................................................. 16
   - BRICS - Computer Science Departments of University of Aarhus and Aalborg
     University (by U.H. Engberg, M. Nielsen) ................. 17
   - IPA - Institute for Programming research and Algorithmics (by T. Borghuis) .... 22

5. Contents of TCS .......................................................... 24

6. EATCS Monographs and Texts ......................................... 39

7. Sister Organizations
   - Contents of SIGACT News .......................................... 51

8. News from Australia (by C.J. Fidge) .............................. 52

9. News from New Zealand (by C.S. Calude) ......................... 54

10. News from Latin America (by A. Viola) ........................... 55

11. News from India (by M. Mukund) .................................. 59

12. Columns
   - J. Diaz: The Algorithmics Column
     - Net Transformations for Petri Net Technology (by M. Urbášek) .......... 77
   - L. Fortnow: The Computational Complexity Column
     - A Short History of Computational Complexity (by L. Fortnow, S. Homer) .... 95
   - G. Rozenberg: The Natural Computing Column
     - Quantum Computation explained to my Mother (by P. Arrighi) .......... 134
   - A. Salomaa: The Formal Language Theory Column
     - The DF0L language equivalence problem (by J. Honkala) ............... 143
13. Technical Contributions

M. Dietzfelbinger, M. Kunde: A case against using Stirling’s formula (unless you really need it) ........................................ 153
M. Domaratzki: On Iterated Scattered Deletion ........................................ 159
T. Harju, D. Nowotka: Periodicity and Unbordered Segments of Words ............. 162
K. Krithivasan, S.V. Varma: On Minimising P Finite State Automata ................. 168

14. Surveys and Tutorials

Challenges in Business Process Management: Verification of business processes using Petri nets (by W.M.P. van der Aalst) ........................................ 174
Threshold Phenomena: The computer Scientist’s Viewpoint (by Y.C. Stamatiou) ......................... 199

15. Education Matters (by H.-J. Kreowski) ........................................ 235
Education(al) matters: teaching P versus NP (by H. Fernau) .......................... 237

16. Reports on Conferences

Workshop on Communication, Specification & Programming (CS&P 2002), Berlin, Germany, 7-9 October 2002 (report by M. Kudlek) ........................................ 247

17. Conference Announcements

DIMACS Workshops and Events Calendar ........................................ 251
Workshop of the UML 2003 Conference on Compositional Verification of UML Models, San Francisco, California, USA, 20 October 2003 ......................... 259
Second International Symposium on Formal Methods for Components and Objects (FMCO 2003), Leiden, The Netherlands, 4-7 November 2003 ......................... 260
Thirty-First International Colloquium on Automata, Languages and Programming (ICALP 2004), Turku, Finland, 12-16 July 2004 ......................... 262

18. Agenda of Events ........................................ 263

19. Abstracts of Ph.D. Theses

J. Crampton: Authorization and Antichains ........................................ 269
M. Koutčý: On Traversal Sequences, Exploration Sequences and Completeness of Kolmogorov Random Strings ........................................ 272

20. Puzzle Corner (by L. Rosaz) ........................................ 275

21. EATCS Leaflet ........................................ 278

22. Cartoons ........................................ 13, 51, 60, 173, 277
Bulletin Editor: G. Rozenberg, Leiden, The Netherlands
Production and distribution: B. Rovan, Bratislava, Slovakia
Picture Editor: M. Kudlek, Hamburg, Germany
Cartoons: DADARA, Amsterdam, The Netherlands

The Editor is grateful to Ms. M. van der Nat and Dr. H.J. Hoogeboom for their help in the tedious task of editing this issue.

All contributions should be sent to the Bulletin Editor:

Prof. Dr. G. Rozenberg
Leiden University, Leiden Institute for Advanced Computer Science (LIACS)
Niels Bohrweg 1, NL-2333 CA Leiden, The Netherlands

Electronic submissions are accepted only in (La)TeX or Postscript. They should be sent to: rozenber@liacs.nl. Please prepare full-size A4 pages.

Deadlines when submissions have to be in the hands of the Bulletin Editor are:
15 January, 15 May and 15 September, for the February, June and October issue respectively.

All pictures (preferable black and white) including text of what they are showing should be sent to the Picture Editor:

Prof. Dr. M. Kudlek
University of Hamburg, Department of Computer Science
Vogt-Köln Straße 30, D-22527 Hamburg, Germany

Deadlines are 2 weeks before those indicated above for written contributions

The EATCS home page is: http://www.eatcs.org
Letter from the President

Dear EATCS member.

Our next ICALP in Eindhoven is approaching. You can get detailed information from www.win.tue.nl/icalp2003 or inside this Bulletin.

During ICALP we shall have our annual General Assembly. Following the procedures from the past, an EATCS activity report will be made available on our web pages www.eatcs.org, in order for members to read it before the General Assembly.

As mentioned in my letter from the February issue of the Bulletin, one of the important issues on the agenda is the selection of candidates for the election of ten new members of the EATCS Council. Following the election we shall finally have fully implemented our new statutes approved in 2000. The election itself will be organized following ICALP as an electronic ballot from our web pages. All members will get detailed information on the election procedures, which will also be available on our web pages.

The composition of our Council is vital to EATCS, and hence I encourage all members, who are prepared to allocate some of their time to EATCS, to propose their names for the ballot. Also, I hope that all members will actively take part in the election by voting.

At our General Assembly in Eindhoven, we shall also decide on the venue for ICALP in 2005. As you will know ICALP will go to Turku in 2004, co-located with LICS, the IEEE Symposium on Logic in Computer Science. For ICALP 2005, so far we have a strong candidate in Lisbon, Portugal.

One particular item of importance for EATCS at the moment is the role of basic research in computer science as part of the European Sixth Framework Programme. Any suggestion for EATCS initiatives on this issue is most welcomed.

Finally, let me remind you that you are always encouraged to send comments, suggestions or criticism to me or any other council member.

Aarhus, May 2003

Mogens Nielsen
Letter from the Bulletin Editor

We all look forward to ICALP in Eindhoven, but in the meantime, with this issue of the Bulletin, you get a lot of interesting material to read.

This includes all the columns, and news from our correspondents from all over the world. As the special ICALP feature of this issue, you get two really stimulating survey/tutorial papers on (theoretical) challenges in business process management (by van der Aalst), and on the threshold phenomena from the point of view of computer science (by Stamatiou).

This issue will be also very handy to you during ICALP - it contains the detailed scientific and social program of our meeting as well as interesting information about the meeting venue.

Finally, please read the letter from our President - it contains a lot of useful and important information about EATCS.

See you in Eindhoven.

Tot ziens.

G. Rozenberg

Leiden, May 2003
Report from Japanese Chapter

O. Watanabe (Tokyo Inst. of Tech.)

Report on the 1st EATCS/LA Workshop on TCS (Feb. 3 ~ 5, 2003)
The first EATCS Japan chapter workshop was held at Univ. Kyoto, Research Institute of Mathematical Sciences, February 3rd - 5th, 2003.

The workshop was jointly organized with LA Symposium, the Japanese association of theoretical computer scientists that has been organizing workshops twice a year since 1970. The purpose of this workshop is to give a place for discussing topics on all aspects of theoretical computer science. Please check our web page http://www.is.titech.ac.jp/~watanabe/eatcs-jp/eatcs-la02/ for the list of presented papers. You may be able to find something interesting!

EATCS/LA Presentation Award to Dr. ZhiZhou Chen
For the above workshop, we started the EATCS/LA Presentation Award, which is given to one or two young speakers who make an excellent presentation at the workshop. The awardee is selected by votes from those attended the workshop.

This year we selected Dr. ZhiZhou Chen (Tokyo Denki Univ.) for his presentation of the following paper:

Improved Algorithms for 2-Interval Scheduling and NMR Spectral Peak Assignment
by Z.Z. Chen, T. Jiang, G. Lin, and J. Wen

An award plate and two year EATCS membership were awarded to Dr. Chen. Congratulations!

First Business Meeting
The first business meeting of the EATCS Japan Chapter was also held at the workshop. After the opening talk by Prof. Inagaki (Chair of EATCS Japan Chapter), the following items have been reported / discussed.

- Introduction/Invitation to EATCS and Japanese Chapter.
- Explanation on the EATCS/LA Presentation Award
- The name of the workshop; we have decided to call it EATCS/LA Workshop on TCS.

Next our regular article — Nippon from South to North — follows. This time, we asked Prof. Arai at NII to introduce our renewed center of informatics, located in Takebashi, Tokyo.
Nippon from South to North

Research Activities in National Institute of Informatics (NII)

1. Historical Overview of National Institute of Informatics
The National Institute of Informatics (formerly called the National Center for Science Information Systems (NACSIS)) was founded as an inter-university research center of informatics in 1986. It started Ph.D program in the frame of the Department of Informatics of the University of Graduated Studies (Sokendai) in 2002. Now, 36 (13 from overseas) Ph.D students are studying at NII.

2. Missions of National Institute of Informatics
The main mission of NII is to function as an inter-university collaborative organization for comprehensive research on informatics and development and application of an advanced infrastructure for disseminating scientific information. To meet this goal, NII encourages partnership with private sector and public sector both national and international: every year, NII invites researchers for joint research programs. NII Ph.D program covers diverse aspects of information sciences from purely theoretical computer science to multimedia information processing and information sociology. NII also provides inter-university information services such as scholarly information networks, catalog and information retrieval services, and a variety of projects including educational and training programs for university libraries.

3. Foundation of Informatics Research Division
One of eight divisions of NII devotes for studies of theoretical aspects of information science; Foundation of Informatics Research Division. The following are examples of research areas covered by Foundation of Informatics Research Division.

- Research on mathematical informatics
- Research on agents having adductive reasoning capability
- Computational and proof complexity
- Machine learning for natural language processing
- Discrete algorithm
- Quantum computing
- Coordination of speech and gesture
- Bio-informatics for comparative genomics

4. Staffs in Foundation of Informatics Research Division

- Foundation of Algorithm
  - Makoto Tatsuta (type theory, program synthesis using constructive sets and coinductive definitions)
- Takeaki Uno (discrete algorithms, combinatorial optimization, enumeration algorithms)
- Kazushige Terui (constructive logics and computational complexity, proof theory and semantics of linear logic)
- Research on Mathematical Informatics
  - Ken Hayami (numerical analysis, numerical linear algebra)
  - Noriko Arai (computational and proof complexity, automated theorem proving, distance learning)
- Research on Symbolic Reasoning
  - Ken Satoh (construction of multiagent systems with speculative computation)
  - Nigel Collier (machine learning for semantic annotation of Web pages, natural language processing, ontology engineering)
  - Ken Kaneiwa (ontology-oriented logical reasoning systems, logic with structural expressions for the meaning of information)
- Cognitive Science
  - Tsuyoshi Murata (Web mining, machine discovery, diagrammatic reasoning)
  - Nobuhiko Furuyama (a study on coordination of speech and gesture)
- Quantum Computing
  - Keiji Matsumoto (theory of quantum computing)

Noriko Arai
Foundation of Informatics Research Division,
National Institute of Informatics, Tokyo
arai@nii.ac.jp
ICALP 2003

Thirtieth International Colloquium on Automata Languages and Programming
http://www.win.tue.nl/icalp2003/

For the second time in its 30 year history, the annual meeting of the European Association of Theoretical Computer Science will be held in The Netherlands, at the Technische Universiteit Eindhoven.

Conference program

<table>
<thead>
<tr>
<th>Time</th>
<th>Session</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>09.15 - 19.30</td>
<td>OPENING - Blauwe zaal</td>
<td></td>
</tr>
<tr>
<td>09.30 - 10.30</td>
<td>INVITED LECTURE Jan Bergstra Polarized process algebra and program equivalence</td>
<td>Blauwe zaal</td>
</tr>
<tr>
<td>10.30 - 11.00</td>
<td>Coffee and tea break - Voorhof/Senaatszaal</td>
<td></td>
</tr>
<tr>
<td>11.00 - 12.30</td>
<td>CZ4 ALGORITHMS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Juraj Hromkovic, Georg Schnitger Pushdown automata and multicycle machines, a comparison of computation modes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stefan Blom, Wan Fokkink, Sumit Nain</td>
<td>On the axiomatizability of ready traces, ready simulation and failure traces</td>
</tr>
<tr>
<td></td>
<td>Annalisa De Bonis, Leszek Gasieniec, Ugo Vaccaro Generalized framework for selectors with applications in optimal group testing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Daniele Gorla, Rosario Fuglise Resource access and mobility control with dynamic privileges acquisition</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Daniel Bleichenbacher, Aggelos Kiayias, Moti Yung Decoding of interleaved Reed Solomon codes over noisy data</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Nadia Busi, Maurizio Gabbrielli, Gianluigi Zavattaro Replication vs. recursive definitions in channel based calculi</td>
<td></td>
</tr>
<tr>
<td>12.30 - 14.00</td>
<td>Lunch - Voorhof/Senaatszaal</td>
<td></td>
</tr>
<tr>
<td>14.00 - 15.30</td>
<td>CZ4 APPROXIMATION ALGORITHMS 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Alexander Ageev, Yinyu Ye, Jiawei Zhang Improved combinatorial approximation algorithms for the k-level facility location problem</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Davide Ancona, Sonia Fagorzi, Eugenio Moggi, Elena Zucca Mixin modules and computational effects</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CZ5 LANGUAGES AND PROGRAMMING</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Davide Ancona, Sonia Fagorzi, Eugenio Moggi, Elena Zucca Mixin modules and computational effects</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>Event</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>09.00 - 10.00</td>
<td><strong>INVITED LECTURE</strong> Petra Mutzel: The SPQR-tree data structure in graph drawing - Blauwe zaal</td>
<td></td>
</tr>
<tr>
<td>10.00 - 10.30</td>
<td>Coffee and tea break - Voorhof/Senaatszaal</td>
<td></td>
</tr>
<tr>
<td>10.30 - 12.30</td>
<td><strong>CZ4 GRAPH ALGORITHMS</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Amos Korman, David Peleg: Labeling schemes for weighted dynamic trees</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Surender Baswana, Sandeep Sen: A simple linear time algorithm for computing a ((2k - 1))-Spanner of (O(n^{k+1/4})) size in weighted graphs</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Alex Hall, Steffen Hippler, Martin Skutella: Multicommodity flows over time: efficient algorithms and complexity</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Chandra Chekuri, Marcelo Mydlarz, Bruce Shepherd: Multicommodity demand flow in a tree</td>
<td></td>
</tr>
<tr>
<td>12.30 - 14.00</td>
<td>Lunch - Voorhof/Senaatszaal</td>
<td></td>
</tr>
<tr>
<td>14.00 - 16.00</td>
<td><strong>CZ4 OPTIMIZATION AND GAMES</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Eyal Even-Dar, Alex Kesselman, Yishay Mansour: Convergence time to Nash equilibrium</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Thierry Cachat: Higher order pushdown automata, the Causal hierarchy of graphs and parity games</td>
<td></td>
</tr>
</tbody>
</table>

**Tuesday, July 1**

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.00 - 10.30</td>
<td>Coffee and tea break - Voorhof/Senaatszaal</td>
</tr>
<tr>
<td>10.30 - 12.30</td>
<td><strong>CZ4 AUTOMATA I</strong></td>
</tr>
<tr>
<td></td>
<td>Luis Antunes, Lance Fortnow: Sophistication revisited</td>
</tr>
<tr>
<td></td>
<td>John Hitchcock, Jack Lutz, Elvira Mayordomo: Scaled dimension and nonuniform complexity</td>
</tr>
<tr>
<td></td>
<td>Peter Heyer, Michele Mosca, Ronald de Wolf: Quantum search on bounded-error inputs</td>
</tr>
<tr>
<td></td>
<td>Rahul Jain, Jaikumar Radhakrishnan, Pranab Sen: A direct sum theorem in communication complexity via message compression</td>
</tr>
<tr>
<td>12.30 - 14.00</td>
<td>Lunch - Voorhof/Senaatszaal</td>
</tr>
<tr>
<td>14.00 - 16.00</td>
<td><strong>CZ4 GRAPH ALGORITHMS</strong></td>
</tr>
<tr>
<td></td>
<td>Markus Bläser: An improved approximation algorithm for the asymmetric TSP with strengthened triangle inequality</td>
</tr>
<tr>
<td></td>
<td>Alexander Okhotin: Decision problems for language equations with Boolean operations</td>
</tr>
<tr>
<td></td>
<td>Rajiv Gandhi, Eran Halperin, Samir Khuller, Guy Kortsarz, Aravind Srinivasan: An improved approximation algorithm for vertex cover with hard capacities</td>
</tr>
<tr>
<td></td>
<td>Roberto Bruni, José Meseguer: Generalized rewrite theories</td>
</tr>
</tbody>
</table>

**15.30 - 16.00** Coffee and tea break - Voorhof/Senaatszaal
<table>
<thead>
<tr>
<th>Time</th>
<th>Session</th>
<th>Speaker(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>09.00 - 10.00</td>
<td>Invited Lecture Amos Fiat: Some issues regarding</td>
<td>Richard Mayr</td>
</tr>
<tr>
<td></td>
<td>search, censorship, and anonymity in peer to peer networks</td>
<td>Undecidability of weak bisimulation equivalence for 1-counter processes</td>
</tr>
<tr>
<td>10.00 - 10.30</td>
<td>Coffee and tea break - Voorhof/Senaatszaal</td>
<td>Vipul Bansal, Aseem Agrawal, Varun Malhotra</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stable marriages with multiple partners: efficient search for an optimal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>solution</td>
</tr>
<tr>
<td>10.30 - 12.30</td>
<td>C24 Online Problems</td>
<td>Arnul Carayol, Thomas Colcombet</td>
</tr>
<tr>
<td></td>
<td></td>
<td>On equivalent representations of infinite structures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Endre Boros, Khaled Elbassioni, Vladimir Gurvich, Leonid Khachiyan, Kazuhisa</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Makino</td>
</tr>
<tr>
<td>16.00 - 18.00</td>
<td>Excursion to and reception at Philips High Tech Campus</td>
<td>Rainer Feldmann, Martin Gaering, Thomas Lücking, Burkhard Monien, Manuel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Roic Nashifcintwn and the coordination ratio for a selfish routing game</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Richard Mayr</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Undecidability of weak bisimulation equivalence for 1-counter processes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vipul Bansal, Aseem Agrawal, Varun Malhotra</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stable marriages with multiple partners: efficient search for an optimal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>solution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Endre Boros, Khaled Elbassioni, Vladimir Gurvich, Leonid Khachiyan, Kazuhisa</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Makino</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rainer Feldmann, Martin Gaering, Thomas Lücking, Burkhard Monien, Manuel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Roic Nashifcintwn and the coordination ratio for a selfish routing game</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Richard Mayr</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Undecidability of weak bisimulation equivalence for 1-counter processes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vipul Bansal, Aseem Agrawal, Varun Malhotra</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stable marriages with multiple partners: efficient search for an optimal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>solution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Endre Boros, Khaled Elbassioni, Vladimir Gurvich, Leonid Khachiyan, Kazuhisa</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Makino</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rainer Feldmann, Martin Gaering, Thomas Lücking, Burkhard Monien, Manuel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Roic Nashifcintwn and the coordination ratio for a selfish routing game</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Richard Mayr</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Undecidability of weak bisimulation equivalence for 1-counter processes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vipul Bansal, Aseem Agrawal, Varun Malhotra</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stable marriages with multiple partners: efficient search for an optimal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>solution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Endre Boros, Khaled Elbassioni, Vladimir Gurvich, Leonid Khachiyan, Kazuhisa</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Makino</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rainer Feldmann, Martin Gaering, Thomas Lücking, Burkhard Monien, Manuel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Roic Nashifcintwn and the coordination ratio for a selfish routing game</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Richard Mayr</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Undecidability of weak bisimulation equivalence for 1-counter processes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vipul Bansal, Aseem Agrawal, Varun Malhotra</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stable marriages with multiple partners: efficient search for an optimal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>solution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Endre Boros, Khaled Elbassioni, Vladimir Gurvich, Leonid Khachiyan, Kazuhisa</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Makino</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rainer Feldmann, Martin Gaering, Thomas Lücking, Burkhard Monien, Manuel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Roic Nashifcintwn and the coordination ratio for a selfish routing game</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Richard Mayr</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Undecidability of weak bisimulation equivalence for 1-counter processes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vipul Bansal, Aseem Agrawal, Varun Malhotra</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stable marriages with multiple partners: efficient search for an optimal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>solution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Endre Boros, Khaled Elbassioni, Vladimir Gurvich, Leonid Khachiyan, Kazuhisa</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Makino</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rainer Feldmann, Martin Gaering, Thomas Lücking, Burkhard Monien, Manuel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Roic Nashifcintwn and the coordination ratio for a selfish routing game</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Richard Mayr</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Undecidability of weak bisimulation equivalence for 1-counter processes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vipul Bansal, Aseem Agrawal, Varun Malhotra</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stable marriages with multiple partners: efficient search for an optimal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>solution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Endre Boros, Khaled Elbassioni, Vladimir Gurvich, Leonid Khachiyan, Kazuhisa</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Makino</td>
</tr>
</tbody>
</table>

**Wednesday, July 2**

<table>
<thead>
<tr>
<th>Time</th>
<th>Session</th>
<th>Speaker(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>09.00 - 10.00</td>
<td>Invited Lecture Amos Fiat: Some issues regarding</td>
<td>Richard Mayr</td>
</tr>
<tr>
<td></td>
<td>search, censorship, and anonymity in peer to peer networks - Blauwe zaal</td>
<td>Undecidability of weak bisimulation equivalence for 1-counter processes</td>
</tr>
<tr>
<td>10.00 - 10.30</td>
<td>Coffee and tea break - Voorhof/Senaatszaal</td>
<td>Vipul Bansal, Aseem Agrawal, Varun Malhotra</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stable marriages with multiple partners: efficient search for an optimal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>solution</td>
</tr>
<tr>
<td>10.30 - 12.30</td>
<td>C24 Online Problems</td>
<td>Arnul Carayol, Thomas Colcombet</td>
</tr>
<tr>
<td></td>
<td></td>
<td>On equivalent representations of infinite structures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Endre Boros, Khaled Elbassioni, Vladimir Gurvich, Leonid Khachiyan, Kazuhisa</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Makino</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rainer Feldmann, Martin Gaering, Thomas Lücking, Burkhard Monien, Manuel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Roic Nashifcintwn and the coordination ratio for a selfish routing game</td>
</tr>
<tr>
<td>12.30 - 14.00</td>
<td>Lunch - Voorhof/Senaatszaal</td>
<td>Noam Berger, Béla Bollobás, Christian Borgs, Jennifer Chayes, Oliver Riordan</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Degree distribution of the FKP network model</td>
</tr>
<tr>
<td>14.00 - 15.00</td>
<td>Invited Lecture Doron Peled: Model checking and testing combined - Blauwe zaal</td>
<td>Jan Johannsen, Martin Lange CTLe+ is complete for double exponential time</td>
</tr>
<tr>
<td>15.00 - 15.30</td>
<td>Coffee and tea break - Voorhof/Senaatszaal</td>
<td>Noam Berger, Béla Bollobás, Christian Borgs, Jennifer Chayes, Oliver Riordan</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Degree distribution of the FKP network model</td>
</tr>
<tr>
<td>15.30 - 17.00</td>
<td>C24 Around the Internet</td>
<td>Salvatore La Torre, Margherita Napoli, Mimmo Parente, Gennaro Parlato</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hierarchical and recursive state machines with context-dependent properties</td>
</tr>
<tr>
<td>Time</td>
<td>Event</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>09.00 - 10.00</td>
<td><strong>Invited Lecture</strong> Moshe Vardi: Logic and automata, a match made in heaven - Blauwe zaal</td>
<td></td>
</tr>
<tr>
<td>10.00 - 10.30</td>
<td>Coffee and tea break - Voorhof/Senaatszaal</td>
<td></td>
</tr>
<tr>
<td>10.30 - 12.30</td>
<td><strong>CZ4 Graph Problems</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Luisa Gargano, Mikael Hammar There are spanning spiders in dense graphs (and we know how to find them)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Jiri Fiala, Daniel Paulusma The computational complexity of the role assignment problem</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Erik Demaine, Fedor Fomin, MohammadTaghi Hajiaghayi, Dimitrios Thilikos Fixed-parameter algorithms for the (k,r)-center in planar graphs and map graphs</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Jianer Chen, Iyad Kanj, Ljubomir Perkovic, Eric Sedgwick, Ge Xia Genus characterizes the complexity of graph problems: some tight results</td>
<td></td>
</tr>
<tr>
<td>12.30 - 14.00</td>
<td>Lunch - Voorhof/Senaatszaan</td>
<td></td>
</tr>
<tr>
<td>14.00 - 15.30</td>
<td><strong>CZ4 Data Structures and Algorithms</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Yossi Matias, Ely Porat Efficient pebbling for list traversal synopsis</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Amihood Amir, Yonatan Aumann, Richard Cole, Mohe Lewnestein, Ely Porat Function matching: algorithms, applications, and a lower bound</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Juha Kärkkäinen, Peter Sanders Simple linear work suffix array construction</td>
<td></td>
</tr>
<tr>
<td>15.30 - 16.00</td>
<td>Coffee and tea break - Voorhof/Senaatszaal</td>
<td></td>
</tr>
<tr>
<td>16.00 - 18.00</td>
<td><strong>CZ4 Approximation Algorithms II</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sanjeev Arora, Kevin Chang Approximation schemes for degree-restricted MST and red-blue separation problem</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Chandra Chekuri, Sudipto Guha, Joseph Naor Approximating Steiner k-cuts</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Amin Coja-Oghlan, Christoph Moore, Vishal Sanwalali MAX k-CUT and approximating the chromatic number of random graphs</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Philipp Schnoebelen Oracle circuits for branching-time model checking</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Jo Hannay Axiomatic criteria for quotients and subobjects for higher-order data types</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Thomas Henzinger, Rujit Jhala, Rupak Majumdar Counterexample guided control</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Michele Bugliesi, Silvia Crafa, Amelia Prelic, Vladimir Sassone Secrecy in untrusted networks</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Arabdev Chattopadhyay, Denia Thérien Locally commutative categories</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ernst-Erich Doberkat Semi-pullbacks and simulations in categories of stochastic relations</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Alexander Rabinovich Quantitative analysis of probabilistic lossy channel systems</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Luca de Alfaro, Thomas A. Henzinger, Rupak Majumdar Discounting the future in systems theory</td>
<td></td>
</tr>
</tbody>
</table>
Workshops

In the weekends preceding and following ICALP2003, a number of satellite workshops of interest to the EATCS community will be held.

Pre-conference workshops (June 28-29)

- **Algorithms for Massive Data Sets**
  Organisers: Gerth Brodal, and Rolf Fagerberg
  [http://www.brics.dk/massive03/](http://www.brics.dk/massive03/)

- **FGC Foundations of Global Computing**
  Organiser: Vladimiro Sassone
  [http://www.cogs.susx.ac.uk/users/vs/fgc/](http://www.cogs.susx.ac.uk/users/vs/fgc/)

- **LCMAS Logic and Communication in Multi-Agent Systems**
  Organisers: Wiebe van der Hoek, Mike Wooldridge, Alessio Lomuscio, and Erik de Vink
  [http://www.win.tue.nl/~evink/lcmas03.html](http://www.win.tue.nl/~evink/lcmas03.html)
• Quantum Computing
Organiser: Harry Buhrman
http://www.cwi.nl/themes/ins4/icalp03-quantum/

• SecCo Security Issues in Coordination Models, Languages and Systems
Organisers: Gianluigi Zavattaro, and Riccardo Focardi
http://cs.unibo.it/secco03/

• Stochastic Petri-Nets
Organiser: Peter Kemper

Post-conference workshops (July 4-5)
• Evolutionary Algorithms
Organisers: Thomas Baeck, and Marc Schoenauer

• FUNN 2003 1st International Workshop on the Future of Neural Networks
Organisers: Sander Bohte, Joost Kok, Erkki Oja, and Michiel van Wezel
http://www.cwi.nl/~sbohte/funn/funn.html

• Mathematics, Logic and Computation workshop in honour of N.G. de Bruijn's 85th anniversary. Organisers: Fairouz Kamareddine, and Herman Geuvers
http://www.macs.hw.ac.uk/~fairouz/Bruijn03/

Surrounding events
ICALP2003 is surrounded by a broad program of scientific and social events. Ordered by date, these are:

• June 23-27 Co-located conferences In the week preceding ICALP2003, two international conferences in computer science are hosted by the TU/e: the 24th International Conference on Application and Theory of Petri Nets (ATPN 2003), and the Conference on Business Process Management (BPM 2003)

• June 29 Education matters “Education Matters - The Challenge of Teaching Theoretical Computer Science" is a column in this Bulletin that offers a forum to discuss educational matters with the emphasis on higher education in computer science. ICALP 2003 provides the opportunity for an Education Matters panel discussion on Sunday afternoon (16.00 - 17.30) to which all ICALP participants are invited. The event will be moderated by Hans-Joerg Kreowski, among the panelists are Jozef Gruska (Brno), and Markus Roggenbach (Swansea).

• June 30 Award presentations On Monday evening, the EATCS award and the NVITI Life Time Achievement Award will be presented to the winners at a festive occasion hosted by the city of Eindhoven in its modern city hall. The program features lectures by the winners and a reception.

• July 1 Excursion to the High Tech Campus After end of program on Tuesday, Philips Research invites participants of ICALP for an excursion to the new High Tech Campus that Philips is building in the southwest of Eindhoven. The program includes a tour of the facility, presentations on work done at Philips Research, and a reception.

• July 3 Conference dinner/social event For ICALP2003, the conference dinner and the social event are combined into one unique experience: A Midsummernight's Dream theme party at the Efteling, one of the leading family attraction parks in Europe. The Midsummernight's Dream is a party telling the well-known story by means of actors, musicians, and one of the
most beautiful rides of the Efteling. King Oberon invites you to his Dream Flight Palace were you will be treated to food, drink, song, dance and drama.

City and venue

Eindhoven is located in the south of the Netherlands. It has a very convenient rail connection to the main airport of the Netherlands, Schiphol near Amsterdam. From Schiphol it takes only 1.5 hours by train to reach the centre of Eindhoven. Eindhoven itself has a small international airport, with connections to a number of major European cities. Eindhoven is also easy to reach over land, it is part of the intercity network of the Dutch railways (which makes it well-connected to international trains) and it is situated at the junction of two major Dutch motorways. For detailed information on how to get to Eindhoven, see the Travel section of the ICALP2003 website.

The venue for ICALP2003 is the Auditorium of the Technische Universiteit Eindhoven. All activities related to the conference will take place here, including workshops and lunches. The Technische Universiteit Eindhoven was founded in the 1950s on a patch of uncultivated land near the centre of town. Due to this fortunate circumstance, the university campus is now right in the middle of the fifth largest city in the Netherlands. This means that the railway station, conference hotels, and other facilities are all within walking distance of the venue.

During your stay, you will have you easy access to everything Eindhoven has to offer. It is known for its high tech industry and design institutes, but also has a rich cultural life with theatres, a concert hall, and (many) cinemas. The internationally renowned Van Abbe museum has just reopened after a spectacular remodelling, which allows it to present more of its leading collection of modern and contemporary art. There are over 150 restaurants in Eindhoven, serving food from all corners of the world, and ranging from causal to sophisticated. Due to the mild temperatures in the summer months (around 20 degrees Celsius in July), Eindhoven is a lively city with many outdoor cafes and street festivals. In the city’s green surroundings, there plenty of possibilities for outdoor activities like bicycling, hiking and canoeing. For those with a taste for more radical exercise, Eindhoven has one the premier skate bowls in Europe: Area 51.

Registration and conference fees

ICALP2003 and the associated workshops have three registration categories: Early up to and including Wednesday May 28, Late, from Wednesday May 29 up to and including June 20, and On-site, from June 23 up to and including July 4. There are three categories of participants:

- **EATCS-members**: access to all sessions of the conference, proceedings, coffee&tea, lunches, reception, etc.
- **Students**: as EATCS-members, but without proceedings and without the possibility for on-site registration.
- **Others**: as EATCS-members, but with a year’s membership of EATCS as part of the registration.

The registration procedure for EATCS-member is slightly different from that in previous years, details can be found in the Registration section of the ICALP2003 website. Also, there is a discount of 25 Euros off the registration fee for ICALP2003 for all participants who also register for the co-located conference ATPN2003.

Conference fees (in Euros)

<table>
<thead>
<tr>
<th></th>
<th>Early</th>
<th>Late</th>
<th>On-site</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EATCS Members</strong></td>
<td>325</td>
<td>400</td>
<td>550</td>
</tr>
<tr>
<td><strong>Students</strong></td>
<td>225</td>
<td>300</td>
<td>-</td>
</tr>
<tr>
<td><strong>Others</strong></td>
<td>375</td>
<td>450</td>
<td>600</td>
</tr>
</tbody>
</table>

The combined conference dinner and social event ("Midsumernight’s party") is not included in the registration fee. Participation in the workshops is also charged separately. Fees are equal
for all participant categories: 75 Euros per workshop (100 Euros on-site), and for one-day and two-day workshops. There is one exception: the Quantum Computing workshop is sponsored, which means that participation is free of charge (however, you must register to participate).

Registration

You can register for ICALP 2003, workshops, and co-located events, by means of an online registration form on the ICALP 2003 website. This form also offers the possibility to make reservations at one of the ICALP 2003 conference hotels. The Congress Office of the Technische Universiteit Eindhoven handles hotel reservations and accommodation for ICALP 2003, and offers special arrangements for the conference hotels through the registration form.

Contact information

Visiting address
ICALP 2003
Technische Universiteit Eindhoven
Main Building HG 7.22
Den Dolech 2
5612 AZ Eindhoven
The Netherlands

tel. (+31)-40-2474124 (ICALP2003 Secretariat)
fax (+31)-40-2475361
e-mail icalp2003@tue.nl
url http://www.win.tue.nl/icalp2003/

Postal address
ICALP2003
Dep. of Math. and Comp. Sci.
Technische Universiteit Eindhoven
P.O. Box 513
5600 MB Eindhoven
The Netherlands
2003 Gödel Prize

Awarded to
Yoav Freund and Robert Schapire

The 2003 Gödel Prize for an outstanding journal article or articles in theoretical computer science is awarded to Yoav Freund and Robert Schapire for their paper


The Gödel Prize is sponsored jointly by the European Association for Theoretical Computer Science (EATCS) and the Special Interest Group on Algorithms and Computing Theory of the Association of Computing Machinery (ACM-SIGACT). It includes a monetary award of $5000. The Prize is named in honor of the mathematician Kurt Gödel in recognition of his major contributions to mathematical logic and of his having first posed what has now become the famous "P versus NP" question.

This paper introduced AdaBoost, an adaptive algorithm to improve the accuracy of hypotheses in machine learning. The algorithm demonstrated novel possibilities in analysing data and is a permanent contribution to science even beyond computer science.

Because of a combination of features, including its elegance, the simplicity of its implementation, its wide applicability, and its striking success in reducing errors in benchmark applications even while its theoretical assumptions are not known to hold, the algorithm set off an explosion of research in the fields of statistics, artificial intelligence, experimental machine learning, and data mining. The algorithm is now widely used in practice.

The paper highlights the fact that theoretical computer science continues to be a fount of powerful and entirely novel ideas with significant and direct impact even in areas, such as data analysis, that have been studied extensively by other communities.

This award is presented annually, with the presentation taking place alternately at the International Colloquium on Automata, Languages, and Programming (ICALP) and ACM Symposium on the Theory of Computing (STOC). The eleventh presentation will take place during the 2003 STOC, June 9-11, 2003 in San Diego, California. The winners were chosen by an award committee consisting of Giorgio Ausiello (Roma), László Babai (Chicago), Zvi Galil (Columbia), Juhani Karhumäki (Turku), Dexter Kozen (Cornell), chair, and Ugo Montanari (Pisa).
Past winners:


INSTITUTIONAL SPONSORS

BRICS, Basic Research in Computer Science, Aarhus, Denmark

IPA, Institute for Programming research and Algorithmics, Eindhoven, The Netherlands

Microsoft, Cambridge, United Kingdom

PWS Publishing Company, Boston, U.S.A.

TUeS, Turku Centre for Computer Science, Turku, Finland

UNU/IIST, UN University, International Institute for Software Technology, Macau
Coming Events

In the period July 7–10, 2003, BRICS in Aarhus, organises 18th Annual IEEE Conference on Computational Complexity. Further information can be found by visiting www.brics.dk/Complexity2003/.

For details on coming events, please see the BRICS Activities web page: www.brics.dk/Activities.

Dissertation Abstracts

Domain Specific Languages for Interactive Web Services

Claus Brabrand

This dissertation shows how domain specific languages may be applied to the domain of interactive Web services to obtain flexible, safe, and efficient solutions. We show how each of four key aspects of interactive Web services involving sessions, dynamic creation of HTML/XML documents, form field input validation, and concurrency control, may benefit from the design of a dedicated language.

Also, we show how a notion of metamorphic syntax macros facilitates integration of these individual domain specific languages into a complete language.

The result is a domain specific language, \langle \text{bigwig} \rangle, that supports virtually all aspects of the development of interactive Web services and provides flexible, safe, and efficient solutions. See [DS-03-1].

Hashing, Randomness and Dictionaries

Rasmus Pagh

This thesis is centered around one of the most basic information retrieval problems, namely that of storing and accessing the elements of a set. Each element in the set has some associated information that is returned along with it. The problem is referred to as the dictionary problem, due to the similarity to a bookshelf dictionary, which contains a set of words and has an explanation associated with each word. In the static version of the problem the set is fixed, whereas in the dynamic version, insertions and deletions of elements are possible.

The approach taken is that of the theoretical algorithms community. We work (almost) exclusively with a model, a mathematical object that is meant to capture essential aspects of a real computer. The main model considered here (and in most of the literature on dictionaries) is a unit cost RAM with a word size that allows a set element to be stored in one word.
We consider several variants of the dictionary problem, as well as some related problems. The problems are studied mainly from an upper bound perspective, i.e., we try to come up with algorithms that are as efficient as possible with respect to various computing resources, mainly computation time and memory space. To some extent we also consider lower bounds, i.e., we attempt to show limitations on how efficient algorithms are possible. A central theme in the thesis is randomness. Randomized algorithms play an important role, in particular through the key technique of hashing. Additionally, probabilistic methods are used in several proofs. See [DS-02-5].

Program Verification with Monadic Second-Order Logic & Languages for Web Service Development

Anders Moller

Domain-specific formal languages are an essential part of computer science, combining theory and practice. Such languages are characterized by being tailor-made for specific application domains and thereby providing expressiveness on high abstraction levels and allowing specialized analysis and verification techniques. This dissertation describes two projects, each exploring one particular instance of such languages: monadic second-order logic and its application to program verification, and programming languages for construction of interactive Web services. Both program verification and Web service development are areas of programming language research that have received increased attention during the last years.

We first show how the logic Weak monadic Second-order Logic on Strings and Trees can be implemented efficiently despite an intractable theoretical worst-case complexity. Among several other applications, this implementation forms the basis of a verification technique for imperative programs that perform data-type operations using pointers. To achieve this, the basic logic is extended with layers of language abstractions. Also, a language for expressing data structures and operations along with correctness specifications is designed. Using Hoare logic, programs are split into loop-free fragments which can be encoded in the logic. The technique is described for recursive data types and later extended to the whole class of graph types. As an example application, we verify correctness properties of an implementation of the insert procedure for red-black search trees.

We then show how Web service development can benefit from high-level language support. Existing programming languages for Web services are typically general-purpose languages that provide only low-level primitives for common problems, such as maintaining session state and dynamically producing HTML or XML documents. By introducing explicit language-based mechanisms for those issues, we liberate the Web service programmer from the tedious and error-prone alternatives. Specialized program analyses aid the programmer by verifying at compile time that only valid HTML documents are ever shown to the clients at runtime and that the documents are constructed consistently. In addition, the language design provides support for declarative form-field validation, caching of dynamic documents, concurrency control based on temporal-logic specifications, and syntax-level macros for making additional language extensions. In its newest version, the programming language is designed as an extension of Java. To describe classes of XML documents, we introduce a novel XML schema language aiming to both simplify and generalize existing proposals. All parts are implemented and tested in practice.

Both projects involve design of high-level languages and specialized analysis and verification techniques, supporting the thesis that the domain-specific paradigm can provide a versatile and productive approach to development of formal languages. See [DS-02-4].
Dynamic Planar Convex hull

Riko Jacob

We determine the computational complexity of the dynamic convex hull problem in the planar case. We present a data structure that maintains a finite set of points in the plane under insertion and deletion of points in amortized $O(\log n)$ time. Here $n$ denotes the number of points in the set. The data structure supports the reporting of the extreme point of the set in some direction in worst-case and amortized $O(\log n)$ time. The space usage of the data structure is $O(n)$. We give a lower bound on the amortized asymptotic time complexity that matches the performance of this data structure. See [DS-02-3].

On Resolution Complexity of Matching Principles

Stefan Dantchev

Studying the complexity of mathematical proofs is important not only for automated theorem proving, but also for Mathematics as a whole. Each significant result in this direction would potentially have a great impact on Foundations of Mathematics.

Surprisingly enough, the general Proof Complexity is closely related to Propositional Proof Complexity. The latter area was founded by Cook and Reckhow in 1979, and enjoyed quite a fast development since then. One of the main research directions is finding the precise complexity of some natural combinatorial principle within a relatively weak propositional proof system. The results in the thesis fall in this category. We study the Resolution complexity of some Matching Principles. The three major contributions of the thesis are as follows.

Firstly, we develop a general technique of proving resolution lower bounds for the perfect matching principles based on regular planar graphs. No lower bounds for these were known prior to our work. As a matter of fact, one such problem, the Mutilated Chessboard, was suggested as hard to automated theorem provers in 1964, and remained open since then. Our technique proves a tight resolution lower bound for the Mutilated Chessboard as well as for Tseitin tautologies based on rectangular grid graph. We reduce these problems to Tiling games, a concept introduced by us, which may be of interest on its own.

Secondly, we find the exact Tree-Resolution complexity of the Weak Pigeon-Hole Principle. It is the most studied combinatorial principle, but even its Tree-Resolution complexity was unknown prior to our work. We develop a new, more general method for proving Tree-Resolution lower bounds. We also define and prove non-trivial upper bounds on worst-case proofs of the Weak Pigeon-Hole Principle. The worst-case proofs are first introduced by us, as a concept opposite to the optimal proofs.

Thirdly, we prove Resolution width-size trade-offs for the Pigeon-Hole Principle. Proving the size lower bounds via the width lower bounds was known since the seminal paper of Haken, who first proved an exponential lower bound for the ordinary Pigeon-Hole Principle. The width-size trade-offs however were not studied at all prior to our work. Our result gives an optimal width-size trade-off for resolution in general. See [DS-02-2].

New in the BRICS Report Series, 2002 and 2003


19 Christian Kirkegaard, Anders Møller, and


Mads Sig Ager, Dariusz Biemacki, Olivier Danvy, and Jan Midtgaard. A Functional Correspondence between Evaluators and Abstract Machines. March 2003. 28 pp.


New in the BRICS Notes Series, 2002


New in the BRICS Dissertation Series, 2002 and 2003


Institute for Programming research and Algorithmics

IPA had a busy spring staging its Lentedagen on Bioinformatics and the EEF Summerschool on Concurrency. After supporting the organisation of ICALP2003, which takes place in the Netherlands this summer, IPA will dedicate its Herfstdagen to Compositional Programming Methods.

For more IPA news we refer you to our website.

Coming events

IPA Herfstdagen on Compositional Programming Methods


Compositional Programming Methods is one of the four application areas chosen by IPA as a focus for its research in the period 2002 - 2006. In this area, issues from design, engineering, validation, and verification come together. The leading theme is compositionality: obtaining larger software systems from smaller ones by well-understood composition rules. As the Herfstdagen program develops, more information will become available at: http://www.win.tue.nl/ipa/activities/falldays2003/

IPA sponsors ICALP2003


The 30th annual meeting of the European Association of Theoretical Computer Science, ICALP 2003 will be held at the Technische Universiteit Eindhoven. As usual there will two tracks to the conference: Track A of the meeting covers Algorithms, Automata, Complexity and Games, while Track B covers Logic, Semantics and Theory of Programming.

In the weekends surrounding the conference there is an extensive program of pre- and post-conference workshops. ICALP2003 is collocated with two international conferences on computer science, which will be staged at the Technische Universiteit Eindhoven from June 23-27: the 24th International Conference on Application and Theory of Petri Nets (ATPN 2003), and the Conference on Business Process Management (BPM 2003). See: http://www.win.tue.nl/icalp2003/
Past events

IPA Lentedagen on Bioinformatics
This year’s Lentedagen were dedicated to Bioinformatics, an exciting new scientific field that combines molecular biology with methods and approaches from computer science and statistics. The program provided an overview of Bioinformatics research in and around IPA with sessions dedicated to different parts of the "cycle of life" as it is studied in the life sciences: DNA → RNA → Protein → Cell → Organism → Population → DNA. For abstracts, hand-outs and papers see: http://www.win.tue.nl/ipa/activities/springdays2003/

EEF Foundations School on Concurrency,
Working together with fellow research schools BRICS, TUCS, and UKII in the European Educational Forum (EEF), IPA organises a series of four summer schools on the Foundations of Computer Science. The last event of this series was dedicated to Concurrency and hosted by IPA. The aim of the school was to provide in-depth knowledge on the foundations of Concurrency from a number of approaches. For each approach, training in theoretical foundations was combined with hands-on experience with tools that have been developed within the approach:

- Process algebra, lectures: Jos Baeten and Jan Friso Groote (TU/e), tool: μCRL toolset
- Modelchecking, lecturers: Dennis Dams (Bell Labs) and Dragan Bošnački (TU/e), tool: Spin
- Verification of non-functional properties, lecturer: Kim Larsen (Aalborg University), tool: Uppaal
- Theorem proving, lecturer: Jozef Hooman (University of Nijmegen), tool: PVS
- Petri nets, lecturer: Wil van der Aalst (TU/e), tool: WoFlan

See: http://www.win.tue.nl/ipa/activities/EEFschool/

Addresses

<table>
<thead>
<tr>
<th>Visiting address</th>
<th>Postal address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eindhoven University of Technology</td>
<td>IPA, Fac. of Math. and Comp. Sci.</td>
</tr>
<tr>
<td>Main Building HG 7.17</td>
<td>Eindhoven University of Technology</td>
</tr>
<tr>
<td>Den Dolech 2</td>
<td>P.O. Box 513</td>
</tr>
<tr>
<td>5612 AZ Eindhoven</td>
<td>5600 MB Eindhoven</td>
</tr>
<tr>
<td>The Netherlands</td>
<td>The Netherlands</td>
</tr>
<tr>
<td>tel. (+31)-40-2474124 (IPA Secretariat)</td>
<td></td>
</tr>
<tr>
<td>fax (+31)-40-2475361</td>
<td></td>
</tr>
<tr>
<td>e-mail <a href="mailto:ipa@tue.nl">ipa@tue.nl</a> url <a href="http://www.win.tue.nl/ipa/">http://www.win.tue.nl/ipa/</a></td>
<td></td>
</tr>
</tbody>
</table>
CONTENTS OF TCS

Volume 293, Issue 1, 3 February 2003

Max-PLUS Algebras
Vendee, France, 4 - 7 May 1998

Editorial board
Stephane Gaubert, Jean-Jacques Loiseau, Jean Mairesse, Maurice Nivat, Jean-Eric Pin
Foreword

R.A. Cuninghame-Green, P. Butkovic
The equation $A^x = B^y$ over $(\max, +)$

Jacob van der Woude, Subiono
Conditions for the structural existence of an eigenvalue of a bipartite $(\min, \max, +)$-system

Pierre Bernhard
Minimax—or feared value—LL-control

Karel Zimmermann
Disjunctive optimization, max-separable problems and extremal algebras

Flavio d’Alessandro, Jacques Sakarovitch
The finite power property in free groups

Nami Kobayashi
Some properties of recognizable Z-subsets

Ines Klimann
A solution to the problem of $(A,B)$-invariance for series

Jeremy Gunawardena
From max-plus algebra to nonexpansive mappings: a nonlinear theory for discrete event systems

Luca Aceto, Zoltán Ésik, Anna Ingólfsdóttir
The max-plus algebra of the natural numbers has no finite equational basis

J.-P. Comet
Application of Max-Plus algebra to biological sequence comparisons

Bertrand Ducourthial and Sébastien Tixeuil
Self-stabilization with path algebra
Volume 293, Issue 2, 6 February 2003

Algebraic Methods in Language Processing
Iowa City, USA, 20 – 22 May 2000
Edited by A. Nijholt, G. Scollo and D. Heylen

A. Nijholt  G. Scollo and D. Heylen
Editorial  237-241

Aravind K. Joshi
A note on the strong and weak generative powers of formal systems  243-259

Hans-Peter Kolb, Jens Michaelis, Uwe Mönich, Frank Morawietz
An operational and denotational approach to non-context-freeness  261-289

James Rogers
wMSO theories as grammar formalisms  291-320

Denys Duchier
Dominance constraints with Boolean connectives: a model-eliminative treatment  321-343

Edward P. Stabler, Edward L. Keenan
Structural similarity within and among languages  345-363

Karl-Michael Schneider
Tabular parsing and algebraic transformations  365-389

P. Boullier
Counting with range concatenation grammars  391-416

Peter R.J. Asveld
Algebraic aspects of families of fuzzy languages  417-445

D. Cantone, A. Formisano, E.G. Omodeo and C.G. Zarba
Compiling dyadic first-order specifications into map algebra  447-475

Volume 293, Issue 3, 9 February 2003

Australasian Computer Science
Edited by C. Fidge

Colin Fidge
Foreword  477

Salvatore La Torre, Margherita Napoli
Finite automata on timed omega-trees  479-505

Annabelle McIver and Carroll Morgan
Almost-certain eventualities and abstract probabilities in the quantitative temporal logic qTL  507-534
Shane Saunders, Tadao Takaoka
Improved shortest path algorithms for nearly acyclic graphs 535-556

Peter Schachte
Precise goal-independent abstract interpretation of constraint logic programs 557-577

Author index 583-584

Volume 294, Issue 1-2, 15 February 2003

Category Theory and Computer Science
Edinburgh, 10 - 12 September 1999
Edited by J. Adamek, M. Escardo and M. Hofmann

Editorial board v-ix

Jiri Adamek, Martin Escardó, Martin Hofmann
Preface 1

Jiri Adamek
On final coalgebras of continuous functors 3-29

Anna Bucalo, Carsten Führmann, Alex Simpson
An equational notion of lifting monad 31-60

J.R.B. Cockett, Stephen Lack
Restriction categories II: partial map classification 61-102

Camillo Fiorentini, Silvio Ghilardi
Combining word problems through rewriting in categories with products 103-149

Thomas T. Hildebrandt
Towards categorical models for fairness: fully abstract presheaf semantics of SCCS with finite delay 151-181

Martin Hyland, Andrea Schalk
Glueing and orthogonality for models of linear logic 183-231

Lawrence S. Moss
Recursion and corecursion have the same equational logic 233-267

A.S. Murawski, C.-H. L. Ong
Exhausting strategies, joker games and full completeness for IMLL with Unit 269-305

Hideki Tsuiki
A domain-theoretic semantics of lax generic functions 307-331
Volume 294, Issue 3, 18 February 2003

Linear Logic
Tokyo, 28 March - 2 April 1996
Edited by J.-Y. Girard, M. Okada and A. Scedrov

Jean-Yves Girard, Mitsuhiro Okada, Andre Scedrov
Preface

V. Michele Abrusci
Towards a semantics of proofs for non-commutative logic: multiplicatives and additives

Vincent Danos, Jean-Baptiste Joinet, Harold Schellinx
Computational isomorphisms in classical logic

Stefano Guerrini, Simone Martini, Andrea Masini
Coherence for sharing proof-nets

Raymond McDowell, Dale Miller, Catuscia Palamidessi
Encoding transition systems in sequent calculus

Vaughan Pratt
Chu spaces as a semantic bridge between linear logic and mathematics

Christian Retoré
Handsome proof-nets: perfect matchings and cographs

Lorenzo Tortora de Falco
Additives of linear logic and normalization--Part I: a (restricted) Church-Rosser property

Max I. Kanovich, Mitsuhiro Okada, Andre Scedrov
Phase semantics for light linear logic

Misao Nagayama, Mitsuhiro Okada
A graph-theoretic characterization theorem for multiplicative fragment of non-commutative linear logic

Author index

Volume 295, Issue 1-3, 24 February 2003

Mathematical Foundations of Computer Science
Marianske Lazne, 27 - 31 August 2001
Edited by J. Sgall and A. Pultr

Editorial board

v-ix
<table>
<thead>
<tr>
<th>Authors/Title</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jiri Sgall, Ales Pultr</td>
<td>1</td>
</tr>
<tr>
<td><strong>Foreword</strong></td>
<td></td>
</tr>
<tr>
<td>Andris Ambainis, Arnolds Kikusts</td>
<td>3-25</td>
</tr>
<tr>
<td>Exact results for accepting probabilities of quantum automata</td>
<td></td>
</tr>
<tr>
<td>Albert Atserias</td>
<td>27-39</td>
</tr>
<tr>
<td>Improved bounds on the Weak Pigeonhole Principle and infinitely many primes</td>
<td></td>
</tr>
<tr>
<td>from weaker axioms</td>
<td></td>
</tr>
<tr>
<td>Chris Barrett et al.</td>
<td>41-64</td>
</tr>
<tr>
<td>Reachability problems for sequential dynamical systems with threshold functions</td>
<td></td>
</tr>
<tr>
<td>Markus Bläser</td>
<td>65-83</td>
</tr>
<tr>
<td>The complexity of bivariate power series arithmetic</td>
<td></td>
</tr>
<tr>
<td>Ahmed Bouajjani, Peter Habermehl, Richard Mayr</td>
<td>85-106</td>
</tr>
<tr>
<td>Automatic verification of recursive procedures with one integer parameter</td>
<td></td>
</tr>
<tr>
<td>Sabin Cautis, Filippo Mignosi, Jeffrey Shallit, Ming-wei Wang, Soroosh Yazdani</td>
<td>107-121</td>
</tr>
<tr>
<td>Periodicity, morphisms, and matrices</td>
<td></td>
</tr>
<tr>
<td>Giovanni Di Crescenzo</td>
<td>123-140</td>
</tr>
<tr>
<td>Sharing one secret vs. sharing many secrets</td>
<td></td>
</tr>
<tr>
<td>Pavol Duris, Jan Manuch</td>
<td>141-151</td>
</tr>
<tr>
<td>On the computational complexity of infinite words</td>
<td></td>
</tr>
<tr>
<td>Hervé Fournier</td>
<td>153-169</td>
</tr>
<tr>
<td>Quantifier rank for parity of embedded finite models</td>
<td></td>
</tr>
<tr>
<td>Viliam Geffert</td>
<td>171-187</td>
</tr>
<tr>
<td>Space hierarchy theorem revised</td>
<td></td>
</tr>
<tr>
<td>Viliam Geffert, Carlo Mereghetti, Giovanni Pighizzini</td>
<td>189-203</td>
</tr>
<tr>
<td>Converting two-way nondeterministic unary automata into simpler automata</td>
<td></td>
</tr>
<tr>
<td>Thanh Minh Hoang, Thomas Thierauf</td>
<td>205-222</td>
</tr>
<tr>
<td>The complexity of the characteristic and the minimal polynomial</td>
<td></td>
</tr>
<tr>
<td>Jarkko Kari</td>
<td>223-232</td>
</tr>
<tr>
<td>Synchronizing finite automata on Eulerian digraphs</td>
<td></td>
</tr>
<tr>
<td>Andreas Klein and Martin Kutrib</td>
<td>233-250</td>
</tr>
<tr>
<td>Fast one-way cellular automata</td>
<td></td>
</tr>
<tr>
<td>Chiu-Yuen Koo, Tak-Wah Lam, Tsuen-Wan Ngan, Kunihiko Sadakane, Kar-Keung To</td>
<td>251-261</td>
</tr>
<tr>
<td>On-line scheduling with tight deadlines</td>
<td></td>
</tr>
</tbody>
</table>
Daniel Král', Jan Kratochvíl, Heinz-Jürgen Voss
Mixed hypergraphs with bounded degree: edge-coloring of mixed multigraphs 263-278

Sven O. Krumke, Willem E. de Paepe, Diana Poensgen, Leen Stougie
News from the online traveling repairman 279-294

Nir Piterman, Moshe Y. Vardi
From bidirectionality to alternation 295-321

Pavel Pudlák
On reducibility and symmetry of disjoint NP pairs 323-339

Luigi Santocanale
On the equational definition of the least prefixed point 341-370

K. Sutner
The size of power automata 371-386

Martin Thimm
On the approximability of the Steiner tree problem 387-402

Author index 403-404

Volume 296, Issue 1, 4 March 2003

Computing and Combinatorics
Guilin, China, 20 - 23 August 2001
Edited by J. Wang

Editorial board  v-ix

Jie Wang
Preface  1-2

Oswin Aichholzer, Franz Aurenhammer, Ferran Hurtado, Hannes Krasser
Towards compatible triangulations 3-13

Jin-Yi Cai, Eric Bach
On testing for zero polynomials by a set of points with bounded precision 15-25

Wun-Tat Chan, Tak-Wah Lam, Hing-Fung Ting and Prudence W.H. Wong
On-line stream merging in a general setting 27-46

Otfried Cheong, Chan-Su Shin, Antoine Vigneron
Computing farthest neighbors on a convex polytope 47-58

Zhe Dang, Oscar H. Ibarra, Richard A. Kemmerer
Generalized discrete timed automata: decidable approximations for safety verification 59-74
Rob Duncan, Jianbo Qian, Antoine Vigneron, Binhai Zhu
Polynomial time algorithms for three-label point labeling 75-87

Liying Kang, Hong Qiao, Erfang Shan, Dingzhu Du
Lower bounds on the minus domination and k-subdomination numbers 89-98

Wen-Lian Hsu, Ross M. McConnell
PC trees and circular-ones arrangements 99-116

Michal Koucký
Log-space constructible universal traversal sequences for cycles of length \( O(n^{403}) \) 117-144

Xiang-Yang Li
Generating well-shaped d-dimensional Delaunay Meshes 145-165

Enrico Nardelli, Guido Proietti, Peter Widmayer
Finding the most vital node of a shortest path 167-177

Bodo Manthey
Non-approximability of weighted multiple sequence alignment 179-192

Volume 296, Issue 2, 8 March 2003

Machines, Computations and Universality
Chisinau, Moldova, 2001
Edited by M. Margenstern

Maurice Margenstern
Foreword 193-194

Didier Caucal
On the transition graphs of turing machines 195-223

Henning Fernau
Nonterminal complexity of programmed grammars 225-251

D. Besozzi, G. Mauri, G. Paun, C. Zandron
Gemmating P systems: collapsing hierarchies 253-267

Pierluigi Frisco
Direct constructions of universal extended H systems 269-293

Carlos Martin-Vide, Gheorghe Paun, Juan Pazos and Alfonso Rodriguez-Patón
Tissue P systems 295-326

Francine Herrmann and Maurice Margenstern
A universal cellular automaton in the hyperbolic plane 327-364
K. Sutner
Cellular automata and intermediate degrees 365-375

Volume 297, Issue 1-3, 17 March 2003

Latin American Theoretical Informatics
Punta del Este, Uruguay, 10 - 14 April 2000
Edited by G. Gonnet, D. Panario and A. Viola

Editorial board v-ix

A. Viola Preface 1

Ali Akhavi
The optimal LLL algorithm is still polynomial in fixed dimension 3-23

Pedro Berrizbeitia, Mauricio Odreman, Juan Tena Ayuso
Primality test for numbers M with a large power of 5 dividing M^4-1 25-36

Olivier Carton, Max Michel
Unambiguous Büchi automata 37-81

Serafino Cicerone, Gabriele Di Stefano, Daniele Frigioni, Umberto Nanni
A fully dynamic algorithm for distributed shortest paths 83-102

Myra B. Cohen, Charles J. Colbourn
Optimal and pessimal orderings of Steiner triple systems in disk arrays 103-117

Sylvie Corteel, Mario Valencia-Pabon, Danièle Gardy, Dominique Barth, Alain Denise
The permutation-path coloring problem on trees 119-143

Celina M.H. de Figueiredo, João Meidanis, Célia Picinin de Mello, Carmen Ortiz
Decompositions for the edge colouring of reduced indifference graphs 145-155

Maribel Fernández, Ian Mackie
Operational equivalence for interaction nets 157-181

David Fernández-Baca
Decomposable multi-parameter matroid optimization problems 183-198

Joachim von zur Gathen, Thomas Lücking
Subresultants revisited 199-239

Andreas Goerdt, Mike Molloy
Analysis of edge deletion processes on faulty random regular graphs 241-260

Peter J. Grabner, Arnold Knopfmacher, Helmut Prodinger
Combinatorics of geometrically distributed random variables: run statistics 261-270
Claudio Gutiérrez
Equations in free semigroups with involution and their relation to equations in free groups 271-280

Valentine Kabanets
Almost k-wise independence and hard Boolean functions 281-295

F. Laroussinie, Ph. Schnoebelen, M. Turuani
On the expressivity and complexity of quantitative branching-time temporal logics 297-315

Guy Louchard, John W. Turner
Generalized covariances of multi-dimensional Brownian excursion local times 317-336

Richard Mayr
Undecidable problems in unreliable computations 337-354

F.K. Miyazawa, Y. Wakabayashi
Cube packing 355-366

Lucia Moura
Rank inequalities and separation algorithms for packing designs and sparse triple systems 367-384

Jaroslav Opatrny
Uniform multi-hop all-to-all optical routings in rings 385-397

Brett Stevens
The anti-Oberwolfach solution: pancyclic 2-factorizations of complete graphs 399-424

Stephen Taylor and Marianne Durand
Emerging behavior as binary search trees are symmetrically updated 425-445

Brigitte Vallée
Dynamical analysis of a class of Euclidean algorithms 447-486

Michele Zito
Small maximal matchings in random graphs 487-507

Author index 509-510

Volume 298, Issue 1, 4 April 2003

Selected Papers in honour of Setsuo Arikawa
Edited by T. Shinohara, C.H. Smith and T. Zeugmann

Editorial board v-ix
Takeshi Shinohara, Carl H. Smith, Thomas Zeugmann
Foreword 1-4

Akihiro Yamamoto
Hypothesis finding based on upward refinement of residue hypotheses 5-19

Hiroshi Sakamoto, Kouichi Hirata, Hiroki Arimura
Learning elementary formal systems with queries 21-50

Steffen Lange, Gunter Grieser, Klaus P. Jantke
Advanced elementary formal systems 51-70

Steffen Lange, Jochen Nessel
Decision lists over regular patterns 71-87

Yasuhito Mukouchi, Masako Sato
Refutable language learning with a neighbor system 89-110

Sanjay Jain, Efim Kinber, Rolf Wiehagen, Thomas Zeugmann
On learning of functions refutably 111-143

Wolfgang Merkle, Frank Stephan
Refuting learning revisited 145-177

Takashi Yokomori
Polynomial-time identification of very simple grammars from positive data 179-206

Seishi Okamoto, Nobuhiro Yugami
Effects of domain characteristics on instance-based learning algorithms 207-233

Tatsuya Akutsu, Satoru Kuhara, Osamu Maruyama, Satoru Miyano
Identification of genetic networks by strategic gene disruptions and gene overexpressions under a boolean model 235-251

Takuya Kida et al.
Collage system: a unifying framework for compressed pattern matching 253-272

Volume 298, Issue 2, 8 April 2003

Australasian Computing Theory
Canberra, Australia, February 2000
Edited by D. Wolfram

David Wolfram
Preface 273

Mariangiola Dezani-Ciancaglini, Paula Severi, Fer-Jan de Vries
Infinitary lambda calculus and discrimination of Berarducci trees 275-302
Rod Downey, Lance Fortnow
Uniformly hard languages 303-315

Michael R. Fellows, Catherine McCartin
On the parametric complexity of schedules to minimize tardy tasks 317-324

Bakhadyr Khoussainov
On algebraic and logical specifications of classes of regular languages 325-346

Padmanabhan Krishnan
Automatic synthesis of a subclass of schedulers in timed systems 347-363

Eric Martin, Arun Sharma and Frank Stephan
Learning power and language expressiveness 365-383

Volume 298, Issue 3, 11 April 2003

Foundations of Software Science and Computation Structures
Genova, Italy, 2 - 4 April 2001
Edited by F. Honsell and M. Miculan

Furio Honsell and Marino Miculan
Preface 385-386

Martin Abadi and Bruno Blanchet
Secrecy types for asymmetric communication 387-415

Luca Aceto, Zoltán Ésik and Anna Ingólfsdóttir
Equational theories of tropical semirings 417-469

Michel Bidoit, Rolf Hennicker and Alexander Kurz
Observational logic, constructor-based logic, and their duality 471-510

Mikolaj Bojanczyk
The finite graph problem for two-way alternating automata 511-528

Nadia Busi and Gianluigi Zavattaro
Expired data collection in shared dataspaces 529-556

Cristiano Calcagno, Peter O'Hearn and Richard Bornat
Program logic and equivalence in the presence of garbage collection 557-581

Gerwin Klein and Tobias Nipkow
Verified bytecode verifiers 583-626

Author index 627-628
Volume 299, Issue 1-3, 18 April 2003

Editorial board
v-ix

Fundamental Study
Bruno Courcelle
The monadic second-order logic of graphs XIV: uniformly sparse graphs and edge set quantifications
1-36

Letizia Magnoni, Massimo Mirolli, Franco Montagna and Giulia Simi
PAC learning of probability distributions over a discrete domain
37-63

Julio Aracena and Eric Goles
Complexity of perceptron recognition for a class of geometric patterns
65-79

M. Kiwi
Algebraic testing and weight distributions of codes
81-106

F.K. Hwang
A survey on multi-loop networks
107-121

Chiara Epifanio, Michel Koskas and Filippo Mignosi
On a conjecture on bidimensional words
123-150

David R. Wood
Optimal three-dimensional orthogonal graph drawing in the general position model
151-178

Miklós Bartha and Miklós Krész
Structuring the elementary components of graphs having a perfect internal matching
179-210

Zhi-Zhong Chen et al.
Approximation algorithms for NMR spectral peak assignment
211-229

R. Baeza-Yates, J. Gabarró and X. Messeguer
Fringe analysis of synchronized parallel insertion algorithms in 2-3 Trees
231-271

Subhas C. Nandy, Sandip Das and Partha P. Goswami
An efficient k nearest neighbors searching algorithm for a query line
273-288

Dahlia Malkhi, Yishay Mansour and Michael K. Reiter
Diffusion without false rumors: on propagating updates in a Byzantine environment
289-306

Markus Holzer and Pierre McKenzie
Alternating and empty alternating auxiliary stack automata
307-326

Olivier Finkel
On omega context free languages which are Borel sets of infinite rank
327-346

Dietrich Kuske
Towards a language theory for infinite N-free pomsets
347-386
Kjell Lemström and Lauri Hella
Approximate pattern matching and transitive closure logics 387-412

Zhe Dang, Pierluigi San Pietro and Richard A. Kemmerer
Presburger liveness verification of discrete timed automata 413-438

Leah Epstein and Rob van Stee
Lower bounds for on-line single-machine scheduling 439-450

Michael Rusinowitch and Mathieu Turuani
Protocol insecurity with a finite number of sessions and composed keys is NP-complete 451-475

Francesca Fiorenzi
Cellular automata and strongly irreducible shifts of finite type 477-493

Shankara Narayanan Krishna and Raghavan Rama
Breaking DES using P systems 495-508

Bruce W. Watson
A new regular grammar pattern matching algorithm 509-521

Bruno Durand, Enrico Formenti and Zsuzsanna Róka
Number-conserving cellular automata I: decidability 523-535

Lars Engebretsen and Jonas Holmerin
Towards optimal lower bounds for clique and chromatic number 537-584

Alain Giorgetti
An asymptotic study for path reversal 585-602

Marcel Wild
Idempotent and co-idempotent stack filters and min-max operators 603-631

H. Fernau, M. Holzer and R. Freund
Hybrid modes in cooperating distributed grammar systems: combining the t-mode with the modes ≤ k and = k 633-662

Alexander Okhotin
On the closure properties of linear conjunctive languages 663-685

Oscar H. Ibarra and Zhe Dang
Eliminating the storage tape in reachability constructions 687-706

Adam L. Buchsbaum, Raffaele Giancarlo and Jeffery R. Westbrook
On finding common neighborhoods in massive graphs 707-718

Michael U. Gerber and Daniel Kobler
Algorithms for vertex-partitioning problems on graphs with fixed clique-width 719-734
Note
P.-C. Héam
Some complexity results for polynomial rational expressions 735-741

Note
Y.J. Liu
Regular component decomposition of regular languages 743-749

Note
Andrea E.F. Clementi, Miriam Di Ianni and Riccardo Silvestri
The minimum broadcast range assignment problem on linear multi-hop wireless networks 751-761

Note
Wojciech Rytter
On maximal suffixes and constant-space linear-time versions of KMP algorithm 763-774

Note
Pedro Garcia, José Ruiz and Manuel Vazquez de Parga
Bilateral locally testable languages 775-783

Note
Erzsebet Csuha-Varjú, Gheorghe Paun and György Vaszil
PC*grammar systems with five context-free components generate all recursively enumerable languages 785-794

Corrigendum
Leslie G. Valiant

Author index 797-799

Volume 300, Issue 1-3, 7 May 2003

Editorial board v-ix

Fundamental Study
Peter Aczel, Jiri Adamek, Stefan Milius and Jiri Velebil
Infinite trees and completely iterative theories: a coalgebraic view 1-45

Gian Luca Cattani and Glynn Winskel
Presheaf models for CCS-like languages 47-89

Stacy E. Finkelstein, Peter Freyd and James Lipton
A new framework for declarative programming 91-160

Otmane Ait Mohamed, Xiaoyu Song and Eduard Cerny
On the non-termination of MDG-based abstract state enumeration 161-179
This is a series of monographs published by Springer-Verlag and launched during ICALP 1984: nearly 40 volumes appeared over 13 years. The series includes monographs in all areas of theoretical computer science, such as the areas considered for ICALPs. Books published in this series present original research or material of interest to the research community and graduate students. Each volume is normally a uniform monograph rather than a compendium of articles. The series also contains high-level presentations of special topics. Nevertheless, as research and teaching usually go hand in hand, these volumes may still be useful as textbooks, too.

Texts published in this series are intended mostly for the graduate level. Typically, an undergraduate background in computer science is assumed. However, the background required may vary from topic to topic, and some books may be self-contained. The texts cover both modern and classical areas with an innovative approach that may give them additional value as monographs. Most books in this series will have examples and exercises.

The editors of both series are W. Brauer (Munich), G. Rozenberg (Leiden), and A. Salomaa (Turku). Potential authors should contact one of the editors. The advisory board consists of G. Ausiello (Rome), S. Even (Haifa), J. Hartmanis (Ithaca), N. Jones (Copenhagen), M. Nivat (Paris), C. Papadimitriou (Athens and San Diego), and D. Scott (Pittsburgh).

EATCS Monographs and Texts is a very important EATCS activity and its success depends largely on our members. If you are a potential author or know one please contact one of the editors.

An EATCS member can purchase books from the series with 25% discount. Your order should be sent to:

G. Rozenberg
Leiden University
Leiden Institute of Advanced Computer Science (LIACS)
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

who after acknowledging your EATCS membership forwards the order for further processing to Springer Verlag - they handle the whole matter further on.

Here is the listing of available volumes that have appeared, and are still available from the publisher.

**Volume 6:**
**FUNDAMENTALS OF ALGEBRAIC SPECIFICATION I**
Equations and Initial Semantics
H. Ehrig, B. Mahr

The aim of this book is to present fundamentals of algebraic specifications with respect to the following three aspects, fundamentals in the sense of a carefully motivated introduction to algebraic specifications, which is easy to understand for computer scientists and mathematicians; fundamentals in the sense of mathematical theories which are the basis for precise definitions, constructions, results, and correctness proofs; and fundamentals in the sense of concepts, which are introduced on a conceptual level and formalized in mathematical terms. The book is equally suitable as a text book for graduate courses and as a reference for researchers and system developers.

**Volume 7:**
**PRODUCTS OF AUTOMATA**
F. Gécseg

Both theoretical and practical considerations motivate the representation of objects as certain compositions of simpler ones. In the theory of automata this observation led to the concepts of products and complete systems of automata. The monograph gives a systematic summary of results concerning the hierarchy of compositions of automata determined by the lengths of feedbacks. The first chapter provides an exposition of necessary topics of universal algebra, automata and sequential machines to make the book self-contained. The remaining four chapters discuss the basic properties of the hierarchy under different representations (homomorphic representation, isomorphic representation, generalized products and simulations, representation of automaton mappings in finite length).

To aid the reader in understanding the ideas behind the constructions and proofs, several examples and figures are given. A bibliography and an index are also included.

The text is written for researchers and graduates with interest in automata or switching theory, and may also be useful for those working in computer design and in universal algebra.

**Volume 10:**
**ALGORITHMS IN COMBINATORIAL GEOMETRY**
H. Edelsbrunner

Computational geometry emerged in the early seventies as an area of research in its own right. From the beginning, it was obvious that there were strong connections to questions studied in the considerably older field of combinatorial geometry. For example, the combinatorial structure of a geometric problem usually decides which algorithmic method solves the problem most efficiently. Furthermore, the analysis of an algorithm often requires a great deal of combinatorial knowledge. This book consists of three parts, a combinatorial part, a computational part, and one that presents applications of the results of the first two parts. The choice of the topics covered in this book was guided by the attempt to describe the most fundamental algorithms in computational geometry that have an interesting combinatorial structure. The approach to the subject is new. The book is also an encyclopedic collection of results in the field. Each chapter includes a set of exercises of various degrees of difficulty whose purpose is to give results that extend the material presented and to point out related open problems. Each chapter also contains bibliographic notes which point out solutions in the existing literature to some of the more difficult problems from the exercise section.
**Volume 14:**

**CONFLUENT STRING REWRITING**

M. Jantzen


This monograph treats comprehensively central aspects of string rewriting systems in the form of semi-Thue systems. These are so general as to enable the discussion of all the basic notions and questions that arise in arbitrary replacement systems as used in various areas of computer science. The Church-Rosser property is used in its original meaning and the existence of complete monoid and group presentations is the central point of discussion. Decidability problems with their complexity are surveyed and congruential languages including the deterministic context-free NTS languages are discussed. The book contains a number of generalizations of results published elsewhere, e.g., the uniqueness of complete string rewriting systems with respect of the underlying order. Completely new and unpublished results which serve as an exposition of techniques and new methods are discussed in detail. With the help of semi-Thue systems it is shown in which situations the famous Knuth-Bendix completion method does not terminate and why, and that in general complete replacement systems cannot always be used as algorithms to solve the word problem. It is suggested how these situations can be stated by using a certain control under which the rewriting is to be performed. This monograph is a reference for graduate students and active researchers in theoretical computer science. The reader is led to the forefront of current research in the area of string rewriting and monoid presentations.

**Volume 15:**

**PARSING THEORY**

**Volume I: Languages and Parsing**

S. Sippu, E. Soisalon-Soininen


The entire work "Parsing Theory" appears in two volumes, "Volume I: Languages and Parsing" and "Volume II: LR(k) and LL(k) Parsing". The two volumes form an integral work. Volume I is an introduction to the basic concepts of languages and parsing, and it also contains the relevant mathematical and computer scientific background needed in the development of the theory of deterministic parsing; it deals with topics such as algorithms on relations and graphs, regular languages and lexical analysis, context-free languages, left parsers and right parsers, strong LL(k) parsers and their implementation, simple precedence parsers. Volume II contains a thorough treatment of the theory of LR(k) and LL(k) parsing. "Parsing Theory" is a contemporary reference work on the theory of deterministic parsing of context-free languages. It emphasizes the LR(k) and LL(k) methods, which are developed in a uniform manner and pays special attention to their efficient implementation. Construction algorithms for parsers are derived from general graph-theoretic methods. Complexity questions about parsable grammars are analyzed. The work can be used as a textbook for graduate-level and senior undergraduate-level courses on parsing theory and compiler design. A one-semester course on the basic theory of languages and parsing can be taught from Volume I.

**Volume 17:**

**THE STRUCTURE OF THE RELATIONAL DATABASE MODEL**

J. Paredaens, P. de Bra, M. Gyssens, D. van Gucht


The book presents an overview of the most fundamental aspects of the theory that underlies the Relational Database Model. Most recent books on databases describe the use and the construction of database applications. The present volume focuses on the theoretical background which is necessary for understanding relational databases. The book presents a formal framework in which the concepts can be described unambiguously, an environment for discussing database problems, exact definitions of widely accepted concepts, and exercises that make the book especially useful as a textbook for mathematically oriented disciplines. Besides describing the fundamentals of the relational database model the book
contains four chapters describing assorted research topics, which handle various issues of database design in the context of the relational model.

**Volume 19:**

**COMPILER GENERATORS**

*What they can do, what they might do and what they will probably never do*

M. Tofte


This monograph is concerned with the problem of getting computers to transform formal language definitions into compilers. Its purpose is to demonstrate how certain simple theoretical ideas can be used to generate compilers and even compiler generators. As the title complexity of realistic compilation and the relative simplicity studied in theoretical work is attempted.

The monograph contains an overview of existing compiler generators. The CERES'83 compiler generator, developed by Neil D. Jones and the author, is described in detail. The CERES system is based on the idea of composing language definitions and it serves as an example of a powerful novel "bootstrapping" technique by which one can generate compiler generators as well as compilers by considering a compiler generator to be, in a sense which is made mathematically precise, a special kind of compiler. The core of the CERES system is a two-page-long machine generated compiler generator. The approach uses ideas from denotational semantics and many-sorted algebra and connects them with novel ideas about how to treat programs and language definitions as data. Considerable effort has been made to present the necessary theory in a manner suitable for readers who have some practical experience but not necessarily a theoretical background in semantics.

**Volume 20:**

**PARSING THEORY**

*Volume II: LR(k) and LL(k) Parsing*

S. Sippu, E. Soisalon-Soininen


This is the second volume of a two-volume set representing an up-to-date reference work on the theory of deterministic parsing of context-free grammars. Volume I is an introduction to the basic concepts of formal language theory and context-free parsing. Volume II contains a thorough treatment of the theory of the two most important deterministic parsing methods. The two volumes together form an integrated work with chapters, theorems, lemmas, etc. numbered consecutively. The emphasis is on LR(k) and LL(k) methods, and special attention is paid to the efficient implementation of LR(k) and LL(k) parsers. Construction algorithms for parsers are derived from general graph-theoretic methods. Complexity questions about parsable grammars are analyzed. The work can be used as a textbook in graduate and senior undergraduate courses on parsing theory and compiler design.

**Volume 21:**

**FUNDAMENTALS OF ALGEBRAIC SPECIFICATION II**

*Module Specifications and Constraints*

H. Ehrig, B. Mahr


Two important new concepts, module specifications and constraints, are introduced in this second volume of a three-volume set on fundamentals of algebraic specification. These concepts are motivated by problems in practical software development and are studied here from a theoretical point of view. Modularization is one of the main structuring principles in software development. Modules and module specifications can be seen as the basic building blocks which are used in modularization of software systems and software system specification. Constraints are introduced to increase the expressive power of algebraic specifications in order to make them more useful for practical applications. The book is suitable both as a textbook for graduate courses in formal specification of data types and software systems and as a reference for researchers and system developers.
Volume 22:
STRUCTURAL COMPLEXITY II
J.L. Balcazar, J. Diaz, J. Gabarro
This is the second volume of a systematic two-volume presentation of the various areas of research in
the field of structural complexity. The mathematical theory of computation has developed into a broad
and rich discipline within which the theory of algorithmic complexity can be approached from several
points of view. This volume is addressed to graduate students and researchers and assumes knowledge
of the topics treated in the first volume but is otherwise nearly self-contained. Topics covered include
vector machines, parallel computation, alternation, uniform circuit complexity, isomorphism, biimmunity
and complexity cores, relativization and positive relativization, the low and high hierarchies.
Kolmogorov complexity and probability classes. Numerous exercises and references are given.

Volume 24:
FIRST-ORDER PROGRAMMING THEORIES
T. Gergely, L. Ury
This book proposes a purely classical first-order logical approach to the theory of programming. The
authors, leading members of the famous "Hungarian school", use this approach to give a unified and
systematic presentation of the theory. This approach provides formal methods and tools for reasoning
about computer programs and programming languages by allowing the syntactic and semantic
characterization of programs, the description of program properties, and ways to check whether a given
program satisfies certain properties. The basic methods are logical extension, inductive definition and
their combination, all of which admit an appropriate first-order representation of data and time. The
framework proposed by the authors allows the investigation and development of different programming
theories and logics from a unified point of view. Dynamic and temporal logics, for example, are
investigated and compared with respect to their expressive and proof-theoretic powers. The book should
appeal to both theoretical researchers and students. For researchers in computer science the book
provides a coherent presentation of a new approach which permits the solution of various problems in
programming theory in a unified manner by the use of first-order logical tools. The book may serve as a
basis for graduate courses in programming theory and logic as it covers all important questions arising
between the theory of computation and formal descriptive languages and presents an appropriate
derivation system.

COLOURED PETRI NETS
Basic Concepts, Analysis Methods and Practical Use, Volume 1
K. Jensen
2nd corr. printing 1997, 84 figures, XII, 234 pages, ISBN 3-540-60943-1
This book presents a coherent description of the theoretical and practical aspects of coloured Petri nets
(CP-nets). It shows how CP-nets have been developed, from being a promising theoretical model to
being a full-fledged language for the design, specification, simulation, validation and implementation of
large software systems. The book contains the formal definition of CP-nets and the mathematical theory
behind their analysis methods. It is also intended to attract readers who are more interested in
applications than in the underlying mathematics, and a large part is written in a style which is closer to
an engineering textbook or a user manual than a typical textbook in theoretical computer science. The
book consists of three separate volumes. The first volume defines the net model and the basic concepts.
It gives a detailed presentation of many small examples and a brief overview of some industrial
applications. It introduces the formal analysis methods. Finally, it contains a description of a set of CP-
net tools which support the practical use of CP-nets. Most of the material in this volume is application
oriented. Its purpose is to teach the reader how to construct CP-net models and analyse them by means
of simulation.
RELATIONS AND GRAPHS
Discrete Mathematics for Computer Scientists
G. Schmidt, T. Ströhlein
1993, 203 figures, IX, 301 pages, ISBN 3-540-56254-0
Relational methods can be found at various places in computer science, notably in data base theory, relational semantics of concurrency, relational type theory, analysis of rewriting systems, and modern programming language design. In addition, they appear in algorithms analysis and in the bulk of discrete mathematics taught to computer scientists. This book is devoted to the background of these methods. It is the first to explain how to use relational and graph-theoretic methods systematically in computer science. The powerful calculus of relational algebra is developed with respect to applications to a diverse range of problem areas. Results are first motivated by practical examples, often visualized by both Boolean 0-1 matrices and graphs, and then derived algebraically.

ITERATION THEORIES
The Equational Logic of Iterative Processes
S.L. Bloom, Z. Ésik
Written both for graduate students and research scientists in theoretical computer science and mathematics, this book provides a detailed investigation of the properties of the fixed point or iteration operation. Iteration plays a fundamental role in the theory of computation: for example, in the theory of automata, in formal language theory, in the study of formal power series, in the semantics of flowchart algorithms and programming languages, and in circular data type definitions. It is shown that in all structures that have been used as semantic models, the equational properties of the fixed point operation are captured by the axioms describing iteration theories. These structures include ordered algebras, partial functions, relations, finitary and infinitary regular languages, trees, synchronization trees, 2-categories, and others. The book begins with a gentle introduction to the study of universal algebra in the framework of algebraic theories. A remarkably useful calculus is developed for manipulating algebraic theory terms. The reader then is guided through a vast terrain of theorems and applications by means of detailed proofs, examples, and exercises. The emphasis is on equational proofs, as the title suggests. Of particular interest, the last chapter shows that the familiar topic of correctness logic is a special case of the equational logic of iteration theories. Several significant open problems are scattered throughout the text.

COLOURED PETRI NETS
Basic Concepts, Analysis Methods and Practical Use, Volume 2
K. Jensen
This work presents a coherent description of the theoretical and practical aspects of coloured Petri nets (CP-nets). It consists of three separate volumes. The second volume contains a detailed presentation of the analysis methods for CP-nets. The analysis methods allow the modeller to investigate dynamic properties of CP-nets. The main ideas behind the analysis methods are described as well as the mathematics on which they are based and also how the methods are supported by computer tools. Some parts of the volume are theoretical while others are application oriented. The purpose of the volume is to teach the reader how to use the formal analysis methods, which does not require a deep understanding of the underlying mathematical theory.

COLOURED PETRI NETS
Basic Concepts, Analysis Methods and Practical Use, Volume 3
K. Jensen
This is the third volume of a definitive work on coloured Petri nets. It contains a detailed presentation of 19 applications of CP-nets across a broad range of application areas, including a security system, ATM
networks, audio/video systems, transaction processing, ISDN services, VLSI chips, document storage, distributed programming, electronic funds transfer, a naval vessel, chemical processing, nuclear waste management, and many more. Most of the projects were carried out in an industrial setting, and in each case the original authors have cooperated with the author and approved the new presentation. The author has taken care to unify the terminology and the CPN diagrams and to ensure that the background knowledge required has been provided in the first two volumes of the work.

SYNTAX-DIRECTED SEMANTICS
Formal Models Based on Tree Transducers
Z. Fülöp, H. Vogler
The subject of this monograph is the use of tree transducers to study general properties of formal models for giving semantics to context-free languages. Such formal models include attribute grammars with synthesized attributes only, denotational semantics, and attribute grammars with synthesized and inherited attributes. The authors consider top-down tree transducers, macro tree transducers, attributed tree transducers, and macro attributed tree transducers. A unified terminology is used to define them, and their transformational capacities are compared. Composition and decomposition of the tree transformations induced by the transducers are investigated intensively. This handbook on tree transducers is a base for further research as well as for lecturing the subject to graduate students.

Finiteness and Regularity in Semigroups and Formal Languages
A. de Luca, S. Varricchio
1999, X, 240 pages, 1 figure, ISBN 3-540-63771-0
This is a rigorous and self-contained monograph on a central topic in theoretical computer science: finiteness conditions for semigroups and regularity conditions for formal languages. For the first time in book form, original results from the last ten years are presented, some previously unpublished, using combinatorial and algebraic methods. These are mainly based on combinatorics on words and especially on the theory of "unavoidable regularities" in free monoids. Many finiteness conditions are considered, formulated in terms of such concepts as: permutability, iteration, repetitivity, and chain conditions. These give rise to regularity conditions for formal languages. Non-algebraic regularity conditions are also investigated. A background in mathematics and computer science is required.

PETRI NET ALGEBRA
E. Best, R. Devillers, M. Koutny
This book presents a step-by-step development of a rigorous framework for the specification and verification of concurrent systems. Petri Net Algebra takes its inspiration and methods from Process Algebra and Petri Nets, two prominent branches of concurrency theory. Added synergistic benefit is derived from combining their respective advantages. Petri nets are treated as composable objects, and as such they are embedded in a general process algebra. On the other hand, a generic process algebra is given an automatic Petri net semantics so that net-based verification techniques, based on structural invariants and causal partial orders, can be applied to the process algebra. The book contains full proofs, carefully chosen examples and several possible directions for further research. A unique aspect is that the development of the Petri net algebra is handled so as to allow for further application-oriented extensions and modifications.

INCOMPLETE INFORMATION: STRUCTURE, INFERENCE, COMPLEXITY
S.P. Demri, E.S. Orlowska
This monograph presents a systematic, exhaustive and up-to-date overview of formal methods and theories for data analysis and inference inspired by the concept of rough set. The book studies structures with incomplete information from the logical, algebraic and computational perspective. The formalisms
developed are non-invasive in that only the actual information is needed in the process of analysis without external sources of information being required. The book is intended for researchers, lecturers and graduate students who wish to get acquainted with the rough set style approach to information systems with incomplete information.

**PROCESS ALGEBRA WITH TIMING**

*J.C.M. Baeten, C.A. Middelburg*


Timing issues are of growing importance for the conceptualization and design of computer-based systems. Timing may simply be essential for the correct behaviour of a system, e.g. of a controller. Even if timing is not essential for the correct behaviour of a system, there may be good reasons to introduce it in such a way that suitable timing becomes relevant for the correct behaviour of a complex system. This book is unique in presenting four algebraic theories about processes, each dealing with timing from a different point of view, in a coherent and systematic way. The timing of actions is either relative or absolute and the underlying time scale is either discrete or continuous. All presented theories are extensions of the algebra of communicating processes. The book is essential reading for researchers and advanced students interested in timing issues in the context of the design and analysis of concurrent and communicating processes.

**THEORY OF SEMI-FEASIBLE ALGORITHMS**

*L.A. Hemaspaandra, L. Torenvliet*


This book presents a consolidated survey of the vibrant field of research known as the theory of semi-feasible algorithms. This research stream perfectly showcases the richness of, and contrasts between, the central notions of complexity: running time, nonuniform complexity, lowness, and NP-hardness. Research into semi-feasible computation has already developed a rich set of tools, yet is young enough to have an abundance of fresh, open issues. Being essentially self-contained, the book requires neither great mathematical maturity nor an extensive background in computational complexity theory or in computer science in general. Newcomers are introduced to the field systematically and guided to the frontiers of current research. Researchers already active in the field will appreciate the book as a valuable source of reference.

**STRUCTURAL COMPLEXITY I**

*J.L. Balcazar, J. Diaz, J. Gabarró*

(Former volume 11, see also volume 22)


This is the first volume of a systematic two-volume presentation of the various areas of research on structural complexity. The theory of algorithmic complexity, a part of the mathematical theory of computation, can be approached from several points of view, one of which is the structural one. This volume is written for undergraduate students who have taken a first course in Formal Language Theory. It presents the basic concepts of structural complexity, thus providing the background necessary for the understanding of complexity theory. The second corrected edition has been extended by an appendix with results on nondeterministic space classes and updated with regard to the bibliographical remarks and the references. Note: The first edition was originally published as volume 11 in the series EATCS Monographs on Theoretical Computer Science.
INFORMATION AND RANDOMNESS
An Algorithmic Perspective
C.S. Calude
The book presents in a mathematically clear way the fundamentals of algorithmic information theory and a few selected applications to mathematical logic. Chaitin's Omega Numbers play a significant role: Chaitin classical results are complemented with very recent ones, e.g., the characterization of computable enumerable random reals, the construction of an Omega Number for which ZFC cannot determine any digits and the first successful attempt to compute the exact values of 64 bits of a specific Omega Number. Finally, the book contains a discussion of some interesting philosophical questions related to randomness and mathematical knowledge. In this 2nd edition, new and important results obtained in the last years are described, e.g., the complete characterization of computably enumerable random reals.

BOOLEAN FUNCTIONS AND COMPUTATION MODELS
P. Clote, E. Kranakis
This textbook presents a survey of research on boolean functions, circuits, parallel computation models, function algebras, and proof systems. Its main aim is to elucidate the structure of "fast" parallel computation. The complexity of parallel computation is emphasized through a variety of techniques ranging from finite combinatorics, probability theory and finite group theory to finite model theory and proof theory. Nonuniform computation models are studied in the form of boolean circuits; uniform ones in a variety of forms. Steps in the investigation of non-deterministic polynomial time are surveyed as is the complexity of various proof systems. The book will benefit advanced undergraduates and graduate students as well as researchers in the field of complexity theory.

INTRODUCTION TO PROCESS ALGEBRA
W. Fokkink
Automated and semi-automated manipulation of so-called labelled transition systems has become an important means in discovering flaws in software and hardware systems. Process algebra has been developed to express such labelled transition systems algebraically, which enhances the ways of manipulation by means of equational logic and term rewriting. The theory of process algebra has developed rapidly over the last twenty years, and verification tools have been developed on the basis of process algebra, often in cooperation with techniques related to model checking. This textbook gives a thorough introduction into the basics of process algebra and its applications.

MODELS OF MASSIVE PARALLELLISM
Analysis of Cellular Automata and Neural Networks
M. Garzon
1995, 14 figures, 10 tables, XIV, 272 pages, ISBN 3-540-56149-8
This monograph presents a coherent exposition of analytic methods and results for the exploration and understanding of cellular automata and discrete neural networks as computational and dynamical systems. The methods include exact descriptions of additive cellular automata, and the results concern the computational complexity of totalistic and more general rules of homogeneous and neural network type with discrete activations. The book will be useful both as a textbook and as a reference manual to the scattered literature in the field. Each chapter includes a separate bibliography, as well as pointers to historically relevant papers, and gives exercise problems for the reader.
THE COMPLEXITY THEORY COMPANION
L.A. Hemaspaandra, M. Ogihara
This book provides a companion for students and professionals who seek an accessible, algorithmically oriented, research-centered, up-to-date guide to some of the most interesting techniques of complexity theory. The book's theme is that simple algorithms are at the heart of complexity theory. From the tree-pruning algorithms that dominate the first chapter to the hashing procedures that dominate the last chapter, algorithmic techniques provide the central proof methods of the book. In fact, to most clearly highlight the role of algorithmic techniques in complexity theory, the book is organized by technique rather than by topic. In particular, and in contrast to the organization of previous textbooks on complexity, each chapter of this book focuses on one technique - what it is, and what results and applications it has yielded in complexity theory.

ALGORITHMS FOR HARD PROBLEMS
Introduction to Combinatorial Optimization, Randomized Approximation, and Heuristic Algorithms
J. Hromkovic
This textbook is an introduction to the methods of designing algorithms for hard computing tasks. This area has developed very dynamically in the last years and is one of the kernels of current research in algorithm and complexity theory. The book mainly concentrates on approximate, randomized and heuristic algorithms, and on the theoretical and experimental comparison of these approaches according to the requirements of the practice. There exist several monographs specializing in some of these methods, but no book systematically explains and compares all main possibilities of attacking hard computing problems. Since the topic is fundamental for the university study in computer science and essential for the transfer of theoretical knowledge to the practice, the book tries to close this gap by providing at once a textbook for graduate students and a handbook for practitioners dealing with hard computing problems.

COMMUNICATION COMPLEXITY AND PARALLEL COMPUTING
J. Hromkovic
This book is devoted to the investigation of a special topic in theoretical computer science - communication complexity as an abstract measure of the complexity of computing problems. Its main aim is to show how the theoretical study of communication complexity can be useful in the process of designing effective parallel algorithms. The author shows how to get important information about the parallel complexity (parallel time, the number of processors, the descriptional complexity of the topology of the parallel architecture) of specific computing problems from knowledge of their communication complexity. The book is written as a textbook for undergraduate and graduate students, and provides a careful explanation of the subject as well as motivation for further research.

EXTREME COMBINATORICS WITH APPLICATIONS IN COMPUTER SCIENCE
S. Jukna
The book is a concise, self-contained and up-to-date introduction to extremal combinatorics for non-specialists. Strong emphasis is made on theorems with particularly elegant and informative proofs which may be called gems of the theory. A wide spectrum of most powerful combinatorial tools is presented: methods of extremal set theory, the linear algebra method, the probabilistic method and fragments of Ramsey theory. A thorough discussion of some recent applications to computer science motivates the liveliness and inherent usefulness of these methods to approach problems outside combinatorics. No special combinatorial or algebraic background is assumed. All necessary elements of linear algebra and discrete probability are introduced before their combinatorial applications. Aimed
primarily as an introductory text for graduates, it provides also a compact source of modern extremal
combinatorics for researchers in computer science and other fields of discrete mathematics.

**THE RESOLUTION CALCULUS**  
*A. Leitsch*  
This is a completely new presentation of resolution as a logical calculus and as a basis for
computational algorithms and decision procedures. The first part deals with the traditional topics
(Herbrand's theorem, completeness of resolution, refinements and deletion) but with many new features
and concepts like normalization of clauses, resolution operators and search complexity. The second part
gives a systematic treatment of recent research topics. It is shown how resolution decision procedures
can be applied to solve the decision problem for some important first-order classes. The complexity of
resolution is analyzed in terms of Herbrand complexity, new concepts are used to classify the
complexity of refinements, and functional extension is introduced with resolution to give a strong
calculus.

**DNA COMPUTING**  
*New Computing Paradigms*  
*G. Paun, G. Rozenberg, A. Salomaa*  
This is the first text and monograph about DNA computing, a molecular approach that might
revolutionize our thinking and ideas about computing. Although it is too soon to predict whether
computer hardware to change from silicon to carbon and from microchips to DNA molecules,
the theoretical premises have already been studied extensively. This book starts with an introduction to
DNA-related matters, the basics of biochemistry and language and computation theory, and progresses
to the most advanced mathematical theory developed so far in the area. All three authors are pioneers in
the theory of DNA computing. Apart from being well-known scientists, they are known for their lucid
writing. Many of their previous books have become classics in their field, and this book too is sure to
follow their example.

**PUBLIC-KEY CRYPTOGRAPHY**  
*A. Salomaa*  
(Former volume 23)  
Cryptography, secret writing, is enjoying a scientific renaissance following the seminal discovery in
1977 of public-key cryptography and applications in computers and communications. This book gives a
broad overview of public-key cryptography - its essence and advantages, various public-key
cryptosystems, and protocols - as well as a comprehensive introduction to classical cryptography and
cryptoanalysis. The second edition has been revised and enlarged especially in its treatment of
cryptographic protocols. From a review of the first edition: "This is a comprehensive review ... there
can be no doubt that this will be accepted
as a standard text. At the same time, it is clearly and entertainingly written ... and can certainly stand
alone."

**VERIFICATION OF REACTIVE SYSTEMS**  
*Formal Methods and Algorithms*  
*K. Schneider*  
2003 Approx. 500 pages, ISBN 3-540-00296-0  
Reactive systems are becoming more and more important for essentially all areas of technical and
professional activities as well as for many areas of everyday life. The design of these systems is a great
challenge and requires sound compromises between safety and time-to-market. To meet these needs,
early design phases nowadays include verification of given specifications against system descriptions to find potential design errors as early as possible.
This book is devoted to the foundation of the most popular formal methods for the specification and verification of reactive systems. In particular, the μ-calculus, omega-automata, and temporal logics are covered in full detail; their relationship and state-of-the-art verification procedures based on these formal approaches are presented. Furthermore, the advantages and disadvantages of the formalisms from particular points of view are analyzed. Most results are given with detailed proofs, so that the presentation is almost self-contained.

PARSING SCHEMATAS
A Framework for Specification and Analysis of Parsing Algorithms
K. Sikkel
Parsing, the syntactic analysis of language, has been studied extensively in computer science and computational linguistics. Computer programs and natural languages share an underlying theory of formal languages and require efficient parsing algorithms. This introduction reviews the theory of parsing from a novel perspective. It provides a formalism to capture the essential traits of a parser that abstracts from the fine detail and allows a uniform description and comparison of a variety of parsers, including Earley, Tomita, LR, Left-Corner, and Head-Corner parsers. The emphasis is on context-free phrase structure grammar and how these parsers can be extended to unification formalisms. The book combines mathematical rigor with high readability and is suitable as a graduate course text.

INTRODUCTION TO CIRCUIT COMPLEXITY
A Uniform Approach
H. Vollmer
This advanced textbook presents a broad and up-to-date view of the computational complexity theory of boolean circuits. It combines the algorithmic and the automata-theoretic approaches, and includes extensive discussion of the literature to facilitate further study. It begins with efficient boolean circuits for problems with high practical relevance, then compares the computational model of boolean circuits with other models such as Turing machines and parallel machines. Examination of the complexity of specific problems leads to the definition of complexity classes. The theory of circuit complexity classes is then thoroughly developed, including the theory of lower bounds and advanced topics such as algebraic complexity and connections to finite model theory.

INTRODUCTION TO COMPUTABLE ANALYSIS
K. Weihrauch
Is the exponential function computable? Are union and intersection of closed sets in the real plain computable? Are differentiation and integration computable operators? Is zero finding for complex polynomials computable? Is the Mandelbrot set decidable? And in case of computability, what is the computational complexity? Computable analysis supplies exact definitions for these and many other similar questions and tries to solve them. - Merging fundamental concepts of analysis and recursion theory to a new exciting theory, this book provides a solid fundament for studying various aspects of computability and complexity in analysis. It is the result of an introductory course given for many years and is written in a style suitable for graduate-level and senior students in computer science or mathematics. Many examples illustrate the new concepts while numerous exercises of varying difficulty extend the material and stimulate readers to work actively on the text.
## Contents of SIGACT News

**Volume 34, Number 1, March 2003 (whole number 126)**

<table>
<thead>
<tr>
<th>Section</th>
<th>Pages</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Information</td>
<td>1</td>
<td>Editor's Letter, Quarterly Quote, Mission Statement, For Contributing Authors</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Transitions, Chair's Letter - Gabow</td>
</tr>
<tr>
<td>Regular Features</td>
<td>3</td>
<td>Book Reviews - Gasarch</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td><em>EATCS Bulletin</em>, October 2002 Table of Contents - Rozenberg</td>
</tr>
<tr>
<td></td>
<td>29</td>
<td>Technical Report Column - Kelley</td>
</tr>
<tr>
<td>Technical Columns</td>
<td>31</td>
<td>Complexity Theory - Hemaspaandra</td>
</tr>
<tr>
<td></td>
<td>46</td>
<td>Distributed Computing - Rajsbaum</td>
</tr>
<tr>
<td>Educational Forum</td>
<td>67</td>
<td>Organizer's Column - Ross</td>
</tr>
<tr>
<td>Professional Events</td>
<td>76</td>
<td>Calls for Papers and Participation</td>
</tr>
<tr>
<td></td>
<td>93</td>
<td>Theory Calendar - Pemmaraju</td>
</tr>
<tr>
<td>Articles</td>
<td>99</td>
<td>&quot;A Polynomial-time Algorithm for Solving NP-Hard Problems in Practice - Huang</td>
</tr>
<tr>
<td>Announcements</td>
<td>107</td>
<td>Submission Guidelines</td>
</tr>
</tbody>
</table>
Yet another *Computing: the Australasian Theory Symposium* was held last February. CATS 2003 was the ninth in the series, and was held in Adelaide as part of the *Australasian Computer Science Week* federation of conferences. ACSW was popular once again, with around 350 attendees, approximately 30 of whom attended the CATS sessions. Competition for the attention of attendees was fierce—this year there were eight overlapping ACSW conferences held in a four day period.

The invited speakers for CATS 2003 were George Havas, University of Queensland, who spoke *On the Complexity of the Extended Euclidean Algorithm*, and Arun Sharma from the University of New South Wales. As Conference Chair James Harland said, although both speakers are resident in Australia, their standing in the field is at an international level.

From 31 submissions, the following 17 papers were accepted for presentation at the conference:

- *Information Leakage Detection in Boundary Ambients* by Chiara Braghin, Agostino Cortesi and Riccardo Focardi
- *A New Machine-Checked Proof of Strong Normalisation for Display Logic* by Jeremy Dawson and Rajeev Goré
- *M³: Mobility Types for Mobile Processes in Mobile Ambients* by Maria Coppo, Mariangiola Dezani-Ciancaglini, Elio Giovannetti and Ivano Salvo
- *An Optimal Family of Directed, Bounded-Degree Broadcast Networks* by Michael Dineen and Nian Zhou
- *Large 2-Independent Sets of Regular Graphs* by William Duckworth and Michele Zito
• **On the Relative Complexity of Labelled Modal Tableaux** by Guido Governatori

• **Minimum Augmentation of Edge-connectivity between Vertices and Sets of Vertices in Undirected Graphs** by Toshimasa Ishii, Yoko Akiyama and Hiroshi Nagamochi

• **Flow Analytic Type System for Array Bound Checks** by Yutaka Matsuno and Hiroyuki Sato

• **Space and Time Adaptive Non-blocking Algorithms** by Maurice Herlihy, Victor Luchangco and Mark Moir

• **The Reverse Problem of Range Query** by Tadao Takaoka

• **Hierarchical Automata and P-systems** by Nicoletta Sabadini and Robert Walters

• **Linearity and Passivity** by David Wright

• **Three Approaches to Partiality in the Sketch Data Model** by Michael Johnson and Robert Rosebrugh

• **Formal Semantics for Program Paths** by Karl Lermer, Colin Fidge and Ian Hayes

• **Cutting Up is Hard to Do: the Parameterised Complexity of k-Cut and Related Problems** by Rodney Downey, Vladimir Estivill-Castro, Michael Fellows, Elena Prieto and Frances Rosamond

• **Approximation and Computation of Arbitrage in Frictional Foreign Exchange Market** by Mao-cheng Cai and Xiaotie Deng

• **Characterizing Polynomial Time Computable Functions Using Theories With Weak Set Existence Principles** by Aleksandar Ignjatovic and Phuong Nguyen


In other CATS news, two special Australasian issues of the journal *Theoretical Computer Science* were published recently. Volume 298, Number 2 contains articles derived from papers in the CATS 2000 conference, and Volume 293, Number 3 contains articles originating in CATS 2001.
0. The Fourth International Conference on "Discrete Mathematics and Theoretical Computer Science" organized by the Université de Bourgogne and the CDMTCS will be held in Dijon, France on 7 - 12 July 2003. There will be five invited lectures by Gregory J. Chaitin (IBM, New York), Cunsheng Ding (UST, Hong Kong), Sorin Istrail (Celera Genomics, Rockville), Maurice Margenstern (LITA, Metz), Timothy Walsh (UQAM, Montreal), 18 regular papers and 3 posters. The proceedings will appear as the volume 2731 in Springer-Verlag "Lecture Notes in Computer Science" Series. The webpage of the workshop is https://www.cs.auckland.ac.nz/CDMTCS/conferences/dmcts03/.

1. A workshop on "Logic and Computation" will be held on 11 - 16 January 2004 at the Tahuna Beach Conference Centre in Nelson, at the top of the South Island of NZ. It will be organised by Rod Downey and Rob Goldblatt with funding provided by the New Zealand Institute of Mathematics and its Applications (NZIMA).


   The week before, 3 - 10 January, the NZMRI workshop on "Computational Algebra, Number Theory and Geometry", will be organised by Marston Conder and Eamonn O'Brien; it will be held in the same location in Nelson. The speakers will be: John Cannon (Sydney), John Conway (Princeton), Hendrik Lenstra (Berkeley/Leiden), Peter Neumann (Oxford), Karl Rubin (Stanford) and Charles Sims (Rutgers). The webpage of the workshop is http://www.math.auckland.ac.nz/Conferences/2004/NZMRI/.

2. The latest CDMTCS research reports are (see http://www.cs.auckland.ac.nz/staff-cgi-bin/mjd/secondcgi.pl):


207. B. Khoussainov and T. Kowalski. Games on Graphs: Automata, Structure, and Complexity. 01/2003


210. L. Staiger. Constructive Dimension equals Kolmogorov Complexity. 01/2003

211. M. Margenstern. The Tiling of the Hyperbolic 4D Space by the 120-Cell Is Combinatoric. 02/2003

212. S.A. Terwijn. Complexity and Randomness. 03/2003


C.S. Calude, Auckland
Email: cristian@cs.auckland.ac.nz
In this issue I will present the Workshop on Cryptographic Algorithms, the call for papers of LATIN 2004, the Sixth International Conference on Operations Research to be held in Cuba, the Workshop on Combinatorics, Algorithms, and Applications, the Sixth Brazilian Workshop on Formal Methods, the Argentinian Workshop on Theoretical Computer Science and the Center for Web Research. At the end I will present a list of the main events in Theoretical Computer Science to be held in Latin America in the following months.

Workshop on Cryptographic Algorithms

A workshop on cryptographic algorithms and protocols will be held at the Institute of Computing of the University of Campinas in Brazil, during the week of July 20-27. The main objective of this meeting is to bring together researchers and students in cryptography in South America, in order to foster the development of cooperative research projects. This is the second such workshop held at IC-UNICAMP. This year the workshop will focus on two topics: (i) Elliptic Curve Cryptography, protocols and implementation issues; and (ii) Voting protocols: design and verification. For the first topic, profs. Alfred Menezes, from CACR/Univ. of Waterloo and Darrel Hankerson, from Auburn Univ. have confirmed their participation; they will each lecture a short course on ECC. Detailed programs and registration forms will be available in the next few days at http://www.ic.unicamp.br/~rdahab.

LATIN 2004: call for papers

Latin American Theoretical Informatics (LATIN) was launched in 1992 to foster interaction between the Latin-American computer scientists and computer scientists around the world. This is the sixth in the series, after Sao Paulo, Brazil (1992), Valparaiso, Chile (1995), Campinas, Brazil (1998), Punta del Este, Uruguay (2000) and Cancún, México (2002). It will be held on April 5 to 9, 2004 in Buenos Aires, Argentina, and it is chaired by Martin Farach-Colton.

Papers presenting original research in all areas of theoretical computer science are sought. Topics of interest include (but are not limited to): algorithms and data structures; automata and formal languages; coding; combinatorics and graph theory; complexity theory; computability; computational biology; computational geometry; cryptography and security; databases and information retrieval; the Internet and the web; logic, rewriting, programming theory and semantics; machine learning; parallel and distributed computing; quantum
computing; and verification and theorem proving. The deadline for submission is September 15, 2003 before 11:55 PM GMT. You may visit http://latin04.rutgers.edu to have more information.

Sixth International Conference on Operations Research in Cuba

The last International Conference on Operations Research was held in Havana on March 5 to 8, 2002. This a biannual conference held in Cuba and alternates with the International Workshop on Operations Research, which is devoted to a specific application area. In this occasion both scientific events will coincide, since in autumn 2004 Cuba will organize the Congress of the ALIO (Latin-Iberian American Association of Operations Research). The participants in the 2003 Operations Research's event may assist to the sessions of the Conference and to those of the Workshop. The objectives of the Conference are to foster new results in Optimization, Statistic, Biometry, Probability, Mathematical Economy, Numerical Analysis, Dynamic Systems and Algorithms. Contributions on both theoretical results and applications are welcomed. Papers reporting applications of Operations Research methods for the solution of economic problems will be discussed during the Workshop's sessions. The event will be held in Havana, Cuba on September 15 to 19, 2003, and for more information you may contact sira@matcom.uh.cu or bouza@matcom.uh.cu.

Workshop on Combinatorics, Algorithms, and Applications

The Workshop on Combinatorics, Algorithms, and Applications will take place on the beaches of Sao Paulo state (Hotel Sol e Vida, Ubatuba) from 1st to 5th September 2003. This meeting is being organized by the combinatorics community of Brazil, around the Project "Complexity of Discrete Structures", and it is also expected to be part of the activities of the Mathematics Millennium Institute of IMPA (IM-AGIMB). Dwight Duffus, Alan Frieze, Louis H. Kauffman, Marcos Kiwi, Prabhakar Raghavan, Bruce Reed, Bruce Richmond, Marie-France Sagot, Jeremy Spinrad, Esko Ukkonen, Alfredo Viola, and Kristina Vuskovic have agreed to give invited talks. The organizing committee is in contact with other eminent researchers, and hopes to bring them to this meeting. Information about registration and more details can be found at http://www.ime.usp.br/~yoshi/pronex/Workshop. The event is organized by Yoshiharu Kohayakawa (yoshi@ime.usp.br) and Carlos E. Ferreira (cef@ime.usp.br).

Sixth Brazilian Workshop on Formal Methods (WMF 2003)

The Sixth Brazilian Workshop on Formal Methods will be held in Campina Grande, Brazil on October 13 and 14, 2003. WMF 2003 is the sixth of a series of workshops devoted to the dissemination of the development and use of formal methods for the design and verification of computational systems. It is now a well-established event, with a very good reputation among the academics and research agencies in Brazil. The recent creation of the Special Commission on Formal Methods, within the Brazilian Computer Society, has been an added incentive for our community. The aim of this workshop is to provide an opportunity for researchers with a broad range of interests in formal methods to discuss the recent developments in this field. The themes include, but are not limited to the following: well-founded specification and design languages; formal design methods; model checking; theorem-proving; formal aspects of popular languages and methodologies; formal testing; tools supporting the formal development of computational systems; systems application and experience reports of formal methods.
Papers with a strong emphasis on Formal Methods, whether practical or theoretical, are invited for submission. The deadline for submission of papers July 18, 2003, and the web page is http://www.dsc.ufcg.edu.br/~wmf-2003.

WAIT 2003: Argentinian Workshop on Theoretical Computer Science

The Argentinian Workshop on Theoretical Computer Science (WAIT) has become an important Latin American forum for the exchange of ideas and the presentation of research in theoretical computer science and its applications. The workshop aims are to build a bridge between academic and applied research and to stimulate the exchange of ideas and experience between theory and practise in computer science. It will be held in Buenos Aires, Argentina on September 1 to 5, 2003. The meeting includes contributed and invited talks, and tutorials. Further, we are very pleased to announce that there will be a special issue of ENTCS (http://www.elsevier.nl/locate/entcs) dedicated to WAIT 2003 publishing a selection of outstanding contributions. WAIT 2003 (http://wait2003.famaf.unc.edu.ar), the 7-th workshop in the series, will be held in Buenos Aires during September 1-5 (2003) as part of the 32-nd Argentinian Conference on Informatics and Operations Research (32 JAIIO, http://www.jaiio2003.uade.edu.ar).

Topics of interest include (but are not limited to): logical and algebraic foundations of computer science (logics for computation, category theory, relation algebras, type theory); formal methods (formal specification of sequential and concurrent programs, analysis, verification and transformation of programs, model checking); algorithms and data structures (sequential, parallel, distributed and on-line computing, probabilistic algorithms); automata theory and computational complexity; symbolic and algebraic computation; quantum computing; and bioinformatics.

Center for Web Research (CWR)

The main goal of the Center for Web Research (CWR) is to perform basic research on problems related to the Web, focusing on five specific aspects: management and search of non traditional information (multimedia, structured, etc.), data mining, mathematical modeling of the Web, data extraction from the web, and distributed systems and parallelism. Specifically, this center plans to research on combinatorial pattern matching on images, audio and text; similarity searching of atomic and structured objects; handling semistructured information; mathematical simulation of the process that occur in the Web; and distributed agents and platforms in Internet. The CWR becomes a unique opportunity to join in a common project the efforts of geographically distributed top level researchers. This center is based at the Department of Computer Sciences of the Universidad de Chile, and its director is Dr. Ricardo Baeza-Yates. For more information you can visit http://www.ciw.cl.

Regional Events


- September 15 - 19, 2003, Havana, Cuba: Sixth International Conference on Operations Research and Fifth Workshop on Operations Research: Applications to the Economy (E-mail:sira@matcom.uh.cu,bouza@matcom.uh.cu).


- November 10 - 12, 2003, Santiago de Chile, Chile: First Latin American Web Congress (http://1a-web.org/).

- November 17 - 21, 2003, Sancti Spiritus, Cuba: II Conferencia Internacional de Matemática y Computación (E-mail:matemat@ispss.rimed.cu).

NEWS FROM INDIA

In this edition of News from India, we report on two events held during January, 2003.

TECS Week '03, January 3–9, 2003

The TCS Excellence in Computer Science Week (TECS Week '03), was organized by the Tata Research Development and Design Centre (TRDDC), Pune, a unit of Tata Consultancy Services (TCS), in early January.

The idea of having an annual TECS Week meeting came about after the success of the school on Formal Software Engineering hosted by TRDDC in January, 2002. That school had been jointly organized by WG2.3 of IFIP, UNU/IIST, Macau and the Indian Association for Research in Computing Science (IARCS). The enthusiastic response for last year's school led TRDDC to decide to organize a similar event each year, aimed at a cross-section of computer scientists from the software industry and academic/research institutions.

An International Advisory Board (IAB) has been constituted to supervise this activity, whose members include He Jifeng (UNU/IIST), C A R Hoare (Microsoft, Cambridge), Gerard Huet (INRIA, Rocquencourt), Mathai Joseph (TRDDC), Jayadev Misra (U Texas, Austin), Amir Pnueli (Weizmann) and Natarajan Shankar (SRI).

The topic for TECS Week '03 was Program Analysis, a field rich in theoretical content and practical applications. Three main courses of six hours each were conducted on the topics of abstract interpretation, data flow analysis and set constraint based analysis. In addition, there were two invited lectures, one on type systems and the other on the use of abstraction in program verification. The first half-day of TECS Week '03 was devoted to providing the participants with a background on program analysis and a demonstration on TRDDC's program analysis framework, Darpan.

The faculty at TECS Week '03 were Chris Hankin (Imperial College), Nevin Heintze (Agere Systems Research) and Ganesan Ramalingam (IBM Research). The invited lectures were delivered by Natarajan Shankar (SRI) and Amir Pnueli (Weizmann).

Participation at TECS Week '03 was open to participants from India and neighbouring countries. Out of roughly 70 applications, around 50 participants were selected based on their background, interests and research inclinations. The final count of participants was about 65 (including 15 from TRDDC), of which around 20 (including TRDDC) were from the industry and the rest were from academia or research. Of the 45 participants with a research background, roughly 25 were graduate students, 15 were faculty members and the rest were graduates or research scholars. There were four participants from outside India—two from Nepal and two from Vietnam.

The general view of the participants was that TECS Week was an excellent initiative that will help in increasing awareness of Computer Science in India and near by countries. It was also felt that the standard of the faculty and lectures was outstanding, though there were some questions about the applicability of such techniques in their daily work. There was a suggestion that an increased number of examples, exercises and discussion sessions might address this issue satisfactorily. The faculty at TECS Week was also happy with the level of receptiveness of the audience and their perceptive abilities.

The members of the IAB who were present at TECS Week '03 also held discussions on possible topics for TECS Week '04. "Specification techniques" has emerged as a strong candidate.
ACL 03, IMSc, Chennai, January 24–29, 2003

The 2nd Preparatory School and Workshop on Automata, Concurrency and Logic (ACL 03) was held during 24–29 January at the Institute of Mathematical Sciences (IMSc), Chennai. The first such effort had been in January, 2001.

The first three days (24–26 January) were a Preparatory School, aimed at college and university computer science teachers. The school featured three 5-hour mini-courses: on Automata and Logic by Kamal Lodaya and R Ramanujam (IMSc), Concurrency by Madhavan Mukund and K Narayan Kumar (Chennai Mathematical Institute (CMI)), and Timed Systems by Paritosh K Pandya (Tata Institute of Fundamental Research, Mumbai) and Deepak D'Souza (CMI/Indian Institute of Science, Bangalore).

The three-day Workshop (27–29 January) followed, and had invited talks by 15 researchers from India, the Indian Ocean and Europe. The Workshop had about 50 participants in all.

The talks covered message-passing and timed systems, temporal logics and their expressiveness over traces, monadic second-order logics over classes of graphs, bisimulation, Petri nets... a reasonable sampling of ongoing work in the area. See http://vwv.imsc.res.in/tcsweb/acl/acl03.html for details.

The workshop was timed to coincide with the visit by some French scientists to Chennai under the Indo-French project on “Automata and concurrency: syntactic methods for verification” (principal investigators: Kamal Lodaya from IMSc and Pascal Weil from LaBRI, Bordeaux).

Some of the graduate students attending said this was the first time they had seen such a wide canvas covered over a short period. They were very appreciative of the School in “bringing them up to current research.” Several participants said they look forward to ACL 05!

Acknowledgments

- Ashok Sreenivas <ashoks@pune.tcs.co.in> provided the report on TECS Week '03.
- Kamal Lodaya <kamal@imsc.res.in> provided the report on ACL 03.

Madhavan Mukund <madhavan@cmi.ac.in>
Chennai Mathematical Institute
The Algorithmics Column

by

Josep DÍAZ

Department of Languages and Computer Systems, Polytechnical University of Catalunya
c/ Jordi Girona, 1-3, 08034 Barcelona, Spain
diaz@lsi.upc.es

Analysis of Algorithms (AofA):

Michael Drmota
Department of Geometry
Vienna University of Technology
Vienna, A-1040
Austria

Wojciech Szpankowski*
Department of Computer Science
Purdue University
W. Lafayette, IN 47907
U.S.A.

Abstract

This article is a continuation of our previous Algorithmic Column [54] (EATCS, 77, 2002) dedicated to activities of the Analysis of Algorithms group during the "Dagstuhl-Period" (1993–1997). The first three meetings took place in Schloss Dagstuhl, Germany. The next three meetings of AofA were in Princeton (1998), Barcelona (1999), and Krynica Morska (near Gdańsk, 2000). We shall present here some research problems that have been the highlights of these three meetings. Three special issues [42, 31, 43] were also published after these meetings and we briefly summarize them.

1 Introduction

The area of analysis of algorithms was born on July 27, 1963, when D. E. Knuth wrote his "Notes on Open Addressing" about hashing tables with linear probing. Since then the area has been undergoing substantial changes: we now use various methods from different branches of mathematics: combinatorics, probability theory, graph theory, real and complex analysis, number theory and occasionally algebra, geometry, operations research, and so forth.

In 1993 the first meeting entirely devoted to the analysis of algorithms was organized by P. Flajolet, R. Kemp and H. Prodinger at Schloss Dagstuhl (Germany). After that there have been two further meetings in Dagstuhl (1995, 1997). Some of the research activities of that time have been described in the first Algorithmic Column [54].

The emergence of AofA as an organized field of research, which began with the Dagstuhl seminars and continues till nowadays, started a transformation from a collection of results on individual problems to a study of methods of general applicability, to an understanding of relationships to classical methods of analysis, combinatorics, and discrete probability, to a web of knowledge that applies in a broad context.


*This research was supported in part by the NSF Grant CCR-0208709.
2 The Contraction Method for Recursive Algorithms

Recursive algorithms are popular tools in computer science. Quicksort is one of the most prominent one. Recursive structures are often subject of precise mathematical analysis since usually a parameter of interest can be translated into recurrences (e.g. the number of comparisons in Quicksort). Assuming that a properly normalized version of such a parameter has a limiting distribution (under a probabilistic model), the above recurrence may further translate into a fixed point equation for the distribution. The main thrust of the contraction method introduced by Rösler and Rüschendorf [51] is to solve such a fixed point equation using Banach's fixed point equation.

In what follows we describe the contraction method when applied to the number of comparisons \( L_n \) of Quicksort sorting \( n \) items. The recursive description of Quicksort translates to

\[
L(L_n) = L\left( L_{Z_{n-1}} + L_{n-Z_n} + n-1 \right), \quad n \geq 2, \tag{1}
\]

where \( L_0 = L_1 = 0, L_2 = 1, Z_n \) is uniformly distributed on \( \{1, 2, \ldots, n\} \), \( L(L_j) = L(\overline{L}_j) \), and \( Z_n, L_j, \overline{L}_j (1 \leq j \leq n) \) are independent. For example, it is an easy exercise to obtain explicit representations for the expected value \( EL_n \). From (1) we find the recurrence

\[
EL_n = n - 1 + \frac{1}{n} \sum_{j=1}^{n} (EL_{j-1} + EL_{n-j})
\]

that can be explicitly solved yielding

\[
EL_n = 2n \log n + n \gamma - 4n + 2 \log n + 2\gamma + 1 + O(\log n/n)
\]

with \( \gamma = 0.57721 \ldots \) being Euler's constant.

Let us now consider the random variable \( Y_n = (L_n - EL_n)/n \) that satisfies the following equation

\[
L(Y_n) = L\left( Y_{Z_n-1} \frac{Z_n-1}{n} + Y_{n-Z_n-1} \frac{n-Z_n}{n} + c_n(Z_n) \right), \quad n \geq 2,
\]

where \( Y_0 = Y_1 = 0, Z_n \) is uniformly distributed on \( \{1, 2, \ldots, n\} \), and \( L(Y_j) = L(\overline{Y}_j) \), and \( Z_n, Y_j, \overline{Y}_j (1 \leq j \leq n) \) are independent. Furthermore,

\[
c_n(j) = \frac{n-1}{n} + \frac{1}{n} (EL_{j-1} + EL_{n-j} - EL_n).
\]

Thus if \( Y_n \) has a limiting distribution \( Y \), then it has to satisfy

\[
L(Y) = L\left( UY + (1-U)\overline{Y} + c(U) \right), \tag{2}
\]

where \( U \) is uniformly distributed on \([0, 1]\), \( L(\overline{Y}) = L(Y) \), \( U, \overline{Y}, Y \) are independent, and

\[
c(x) = 2x \log x + 2(1-x) \log(1-x) + 1.
\]

\(^1\) We denote by \( L(X) \) the distribution function of \( X \).
The first step is to show that (2) has actually a unique solution with $EY = 0$.

Let $D$ denote the space of distribution functions with finite second moment and zero first moment. Then the Wasserstein metric $d_2$ is defined as

$$d_2(F,G) = \inf \| X - Y \|_2,$$

where $\| \cdot \|_2$ denotes the $L_2$-norm and the infimum is taken over all random variables $X$ with distributions function $F$ and all $Y$ with distribution function $G$. It is well known that $(D, d_2)$ constitutes a Polish space.\(^2\) Let $S : D \to D$ be a map defined by

$$S(F) = \mathcal{L}(UX + (1 - U)\overline{X} + c(U)),$$

where $X, \overline{X}, U$ are independent, $\mathcal{L}(X) = \mathcal{L}(\overline{X}) = F$, and $U$ is uniformly distributed on $[0, 1]$. Then one can show that $S$ is a contraction with respect to the Wasserstein metric $d_2$ and, thus, there is a unique fixed point $F \in D$ with $S(F) = F$.

Indeed, let $F, G \in D$ and suppose that $\mathcal{L}(X) = \mathcal{L}(\overline{X}) = \mathcal{L}(Y) = G$, and $U$ is uniformly distributed on $[0, 1]$ such that $U, X, X$ and $U, Y, Y$ are independent. Then $S(F) = \mathcal{L}(UX + (1 - U)\overline{X} + c(U))$ and $S(G) = \mathcal{L}(UY + (1 - U)\overline{Y} + c(U))$ and consequently

$$d_2^2(S(F), S(G)) \leq \| UX + (1 - U)\overline{X} - UY - (1 - U)\overline{Y} \|_2^2 \leq \| U(X - Y) + (1 - U)(\overline{X} - \overline{Y}) \|_2^2 = E(X - Y)^2 \cdot EU^2 + E(\overline{X} - \overline{Y})^2 \cdot E(1 - U)^2 = \frac{2}{3}E(X - Y)^2.$$

Taking the infimum over all possible $X, Y$ we obtain

$$d_2(S(F), S(G)) \leq \sqrt{\frac{2}{3}} d_2(F, G),$$

which completes the proof that $S$ is a contraction.

The final step is to show that $Y_n$ actually converges to $Y$. We refer to [49] for details, but it is sufficient to show that $d_2(\mathcal{L}(Y_n), \mathcal{L}(Y)) \to 0$. In fact, Rösler [49] showed that

$$d_2^2(\mathcal{L}(Y_n), \mathcal{L}(Y)) \leq \frac{2}{n} \sum_{j=1}^{n} \left( \frac{j - 1}{n} \right)^2 d_2^2(\mathcal{L}(Y_{j-1}), \mathcal{L}(Y)) + O \left( \frac{\log^2 n}{n} \right).$$

which implies $d_2(\mathcal{L}(Y_n), \mathcal{L}(Y)) \to 0$. This completes the proof that the normalized number of comparisons $(L_n - E L_n)/n$ has a limiting distribution.

From the fixed point equation (2) it also possible to calculate all moments; e.g. the variance of $Y$ is given by

$$\text{Var} Y = 7 - \frac{2}{3}\pi^2.$$
a fixed point equation is due to Rösler [49]. It is now also known that there exists a density ([55]), which is a bounded $C^\infty$ function, tail estimates are available, and orders of convergence are estimated (compare with [21, 22, 23, 32]). However, no explicit representations for the limiting distribution are known.

In passing, we should add that the contraction method has developed into a very powerful tool in the analysis of (recursive algorithms), see [11, 15, 29, 37, 38, 36, 39, 40, 50].

Finally, we also mention that Dobrow and Fill [l7] used a similar approach to analyze the path length of the so-called recursive trees (this unfortunate term is due to Meir and Moon). These are labelled non-plane trees whose labels increase away from the root. The number of such trees is plainly $(n - 1)!$ as can be seen from the fact that their exponential generating function $Y(x)$ satisfies

$$Y(x) = \int_0^x e^{Y(t)} \, dt.$$  

Symbolically one can read this as: "A tree is a root of minimal label (the $f$) to which is attached a set (the $e^Y$) of similar trees." Taking inspiration of Hennesquin's and Rösler's methods, Dobrow and Fill were able to show the existence of a limit distribution that has interesting features not unlike the quicksort distribution. The structure of recursive trees is also of interest as one of the early examples of a priority queue (i.e., a data structure based on unbalanced heap-like trees).

3 The Height of Binary Search Trees

A binary search tree is a binary tree in which each node contains a key, where the keys are drawn from some totally ordered set, say $\{1, 2, \ldots, n\}$. The first key is stored in the root. The next key is placed either in the left child of the root if its value smaller than the key stored in the root or otherwise in the right child. We repeat this procedure recursively until all $n$ keys are inserted into the tree. Observe that Quicksort can be viewed as building a binary search tree. In fact parameter $L_n$ discussed in the previous section is also equal to the total path length in the associated binary search tree.

There are many interesting parameters of a binary search tree built over randomly selected permutation of $\{1, 2, \ldots, n\}$. We mention here the depth of a key, the height (maximum depth), the total path length, and others. The distribution of the height $H_n$ of such a binary search tree turns out to be an interesting (and difficult) problem. We briefly describe such an analysis, but we start with some history.

In 1986 Devroye [12] proved that the expected value $\mathbb{E}H_n$ satisfies the asymptotic relation $\mathbb{E}H_n \sim c \log n$ (as $n \to \infty$), where $c = 4.31107 \ldots$ is the (largest real) solution of the equation $c \log \left(\frac{2e}{c}\right) = 1$. (Earlier Pittel [41] had shown that $H_n / \log n \to \gamma$ almost surely as $n \to \infty$, where $\gamma \leq c$, compare also with Robson [46]. Later Devroye [13] provided a first bound for the error term, he proved $H_n - c \log n = O(\sqrt{\log n \log \log n})$ in probability.) Based on numerical data Robson conjectured that the variance $\text{Var}H_n$ is bounded. In fact, he could prove (see [47]) that there is an infinite subsequence for which

$$\mathbb{E}|H_n - \mathbb{E}H_n| = O(1),$$
and that his conjecture is equivalent to the assertion that the expected value of the number of
dodes at level \( k = H_n \) is bounded (see [46]). The best bounds were given using two completely
different methods by Devroye and Reed [16] and later by Drmota [18]. They (both) proved

\[
\mathbb{E}H_n = c \log n + O(\log \log n)
\]  

(3)

and

\[
\text{Var}H_n = \mathbb{E}(H_n - \mathbb{E}H_n)^2 = O((\log \log n)^2).
\]

Eventually, Reed [44]\(^3\) settled Robson's conjecture by showing that

\[
\text{Var}H_n = O(1) \quad (n \to \infty).
\]

His approach is related to that of [16], but he also showed that

\[
\mathbb{E}H_n = c \log n - \frac{3c}{2(c-1)} \log \log n + O(1).
\]  

(4)

Reed's approach is purely probabilistic. An analytic proof of Robson's conjecture was given
(independently) by Drmota [19].

The analytic proof of Drmota pays off since some time later he was able to extend his
analysis and obtain the limiting distribution for the height. In [20] he uses a sequence of
functions \( y_k(x) \) defined as

\[
y_k(x) = \sum_{n \geq 0} \Pr[H_n \leq k] \cdot x^n.
\]

Then \( y_0(x) \equiv 1 \) and

\[
y_{k+1}(x) = 1 + \int_0^x y_k(t)^2 dt.
\]  

(5)

Obviously, \( y_k(x) \) are polynomials of degree \( 2^k - 1 \) and have a limit \( y(x) = 1/(1 - x) \) (for
\( 0 \leq x < 1 \)). The main result of [20] states

\[
\Pr[H_n \leq k] = \Psi(n/y_k(1)) + o(1) \quad (n \to \infty),
\]  

(6)

where the \( o(1) \)-error term is uniform for all \( k \geq 0 \) and \( \Psi(y) \), \( y \geq 0 \), is a monotonically
decreasing function with \( \Psi(0) = 1 \) and \( \lim_{y \to \infty} \Psi(y) = 0 \) that satisfies the integral equation

\[
y \Psi(y/e^{1/c}) = \int_0^y \Psi(z) \Psi(y - z) dz.
\]  

(7)

Furthermore, there exist constants \( C, \eta > 0 \) such that

\[
\Pr[|H_n - \mathbb{E}H_n| \geq y] \leq Ce^{-\eta y}, \quad (y > 0).
\]  

(8)

Drmota's method is based on a careful analysis of (5). In particular, if one sets

\[
\bar{y}_k(x) := \int_0^\infty \Psi(y/e^{k/c})e^{-\Psi(1-x)} dy,
\]  

(9)

\(^3\)Reed has also presented his result in Barcelona, 1999.

then $\tilde{y}_k(0) = 1 - o(1)$ and (7) translates to

$$\tilde{y}_{k+1}(x) = \tilde{y}_{k+1}(0) + \int_0^x \tilde{y}_k(t)^2 \, dt.$$ 

Thus, the functions $\tilde{y}_k(x)$ emulate the original functions $y_k(x)$. The idea is to approximate $y_k(x)$ by $\tilde{y}_k(x)$ Observe that

$$\tilde{y}_k(x) = \sum_{n \geq 0} \left( \frac{1}{n!} \int_0^\infty y^n e^{-y} \psi(ye^{-k/c}) \, dy \right) x^n.$$ 

and then the resulting relation (6) is not unexpected any more.

4 Random LC Tries

The primary purpose of a trie [28, 33, 34, 52, 53]) is to store a set $C$ of strings (words, sequences), say $C = \{X^1, \ldots, X^n\}$. Each string is a finite or infinite sequence of symbols taken from a finite alphabet $A = \{\omega_1, \ldots, \omega_V\}$ of size $V = |A|$. Strings are stored in leaves of the trie. The trie over $C$ is built recursively as follows: For $|C| = 0$, the trie is, of course, empty. For $|C| = 1$, trie$(C)$ is a single node. If $|C| > 1$, $C$ is split into $V$ subsets $C_1, C_2, \ldots, C_V$ so that a string is in $C_j$ if its first symbol is $\omega_j$. The tries trie$(C_1), \text{trie}(C_2), \ldots, \text{trie}(C_V)$ are constructed in the same way except that at the $k$th step, the splitting of sets is based on the $k$th symbol. These subtrees are then connected from their respective roots to a single node to create trie$(C)$. When a new string is inserted, the search starts at the root and proceeds down the tree as directed by the input symbols.

There are many possible variations of the trie. One such variation is the $b$-trie, in which a leaf is allowed to hold as many as $b$ strings. The $b$-trie is particularly useful in algorithms for extendible hashing in which the capacity of a page or other storage unit is $b$. A second variation of the trie, the PATRICIA trie (Practical Algorithm To Retrieve Information Coded In Alphanumeric) eliminates the waste of space caused by nodes having only one branch. This is done by collapsing one-way branches into a single node.

Level Compression (LC) tries were introduced by Andersson and Nilsson [5]. They are further compacted versions of tries or PATRICIA tries. The following operation is repeated recursively: at the root of the trie (or PATRICIA trie) $T$, find the highest complete subtree $C$ (of height $h$). Let $T_i$ ($1 \leq i \leq 2^h$) denote the subtrees rooted at level $h$. Replace $T$ by the root of $T$ and the $2^h$ subtrees $T_i$. Repeat the above path compression procedure recursively for every $T_i$. The resulting trie is called LC trie (or LC PATRICIA trie). Note that the number of children of each node is a power of 2.

To analyze LC tries we assume throughout that data $X^1, \ldots, X^n$ are drawn independently and uniformly from $[0,1]$ (and the keys are just the binary expansions of $X^i$). The quantities of interest in a trie (or LC trie) are $D_n$, the depth of the $n$-th string, $A_n$, the typical depth defined as $A_n = \frac{1}{n} \sum_{i=1}^n D_i$, and $H_n$, the height of the trie. Andersson and Nilsson [5] showed that for such probabilistic model (i.e., unbiased memoryless source) the typical depth in LC
tries is $A_n = \Theta(\log^* n)$, where $\log^* n$ is the log-star function, defined as the minimum positive integer $i$ such that $i$-th iterate $\log_2 \log_2 \cdots \log_2 n \leq 1$.

Devroye [14] substantially improved results of Andersson and Nilsson. He showed that for LC tries and LC PATRICIA tries we have (under the uniform model), as $n \to \infty$,

$\mathbb{E} A_n \sim \mathbb{E} D_n \sim \log^* n.$

Furthermore, he showed that

$$\frac{D_n}{\log^* n} \to 1$$

in probability,

$$\frac{H_n}{\log_2 n} \to 1$$

in probability for the height of LC tries, and

$$\frac{H_n}{\sqrt{2 \log_2 n}} \to 1$$

in probability for the height of LC PATRICIA tries.

The proof is based on the property that in a random (PATRICIA) trie the fill-up-level (the number of consecutive full levels starting at the root) is about $\log_2 n - \log_2 \log_2 n$ in probability (cf. [53]). Thus, all these levels are compressed into one node in the corresponding LC trie. The remaining subtrees are now of size about $\log_2 n$. Hence, the fill-up-level of these subtrees is about $\log_2 \log_2 n - \log_2 \log_2 \log_2 n$ and so on. This heuristics shows that the number of levels in the LC trie is approximately $\log^* n$.

These considerations also show that random tries in the uniform model constitute very well balanced binary trees, even if we look at the typical structure after the fill-up-level. The only puzzling thing is that the height of LC-tries is relatively large. For random tries the height is about $2 \log_2 n$ and for random LC tries about $\log_2 n$. For random PATRICIA tries the height is about $\log_2 n + \sqrt{2 \log_2 n}$ and for random LC PATRICIA tries about $\sqrt{2 \log_2 n}$. This means that the LC-construction for tries only compresses the first $\log_2 n$ levels to $\log^* n$ new levels whereas the remaining levels are not really effected by this procedure. There are relatively few nodes at these higher levels because the average depth is not affected but the height is.

5 Lopsided Trees

In this section, we briefly describe a remarkable contribution of Choi and Golin [9] on lopsided trees. Lopsided trees are ordered rooted r-ary trees in which the length of the edge from a parent to its i-th child is $c_i$ (where $c_1 \leq c_2 \leq \cdots \leq c_r$). These kinds of trees model prefix codes, where different letters may have different costs. The total cost of such prefix codes corresponds to the external path length of the corresponding lopsided tree. Especially, one is interested in Varn codes. Varn codes for n symbols are the minimal prefix codes. Equivalently, a Varn code of n words corresponds to a lopsided tree with n external nodes and minimal external path length.
The main contribution of this paper is the classification of the optimal structure and analysis of such trees $T_n$ with $n$ external nodes. We first describe the optimal construction:

One starts by labeling the nodes of an infinite lopsided tree in order of increasing depths. Now, for any set $V$ of nodes we denote by $\text{LEAF}(V)$ the set of nodes that are not in $V$ but their immediate ancestor is in $V$. Furthermore, for $n \leq \lvert \text{LEAF}(V) \rvert$ let $\text{LEAF}_n(V)$ be the $n$ smallest labeled nodes in $\text{LEAF}(V)$ and set

$$
T_n^m = \{1, 2, \ldots, m\} \cup \text{LEAF}_n(\{1, 2, \ldots, m\}),
$$

where $\lceil (n-1)/(r-1) \rceil \leq m \leq n - 1$. Next, let $l_0, l_1, l_2, \ldots$ denote the consecutive levels upon which nodes appear, i.e. $l_0 = 0$ and $l_i = \min\{\text{depth}(v) : \text{depth}(v) > l_{i-1}\}$, and let $m_j$ be the number of nodes $v$ with $\text{depth}(v) \leq l_j$. Finally, set $x_m = (\sum_{i=1}^{m} c_i) / (m-1)$ (for $r = 2, \ldots, r$) and let $k \geq 2$ be defined by

$$
x_2 \geq x_3 \geq \cdots \geq x_{k-1} \geq x_k < x_{k+1} < \cdots < x_r.
$$

With help of this notation we set

$$
A_j = \{v \in \text{LEAF}(V_{m_j}) : \text{depth}(v) \leq l_j + x_k\},
$$

$$
a_j = \lvert A_j \rvert,
$$

$$
B_j = A_j \cup \{v \in \text{LEAF}(V_{m_j}) : l_j + x_k < \text{depth}(v) \leq l_j+1 + x_k\}
$$

and $b_j = \lvert B_j \rvert$.

The classification of optimal lopsided trees of size $n$ is as follows:

1. If $n = a_j$ for some $j \geq 0$ then $T_{a_j}^{m_j} = V_{m_j} \cup A_j$ is an optimal lopsided tree.
2. If $a_j < n \leq b_j$ for some $j \geq 0$ then $T_{a_j}^{m_j}$ and $T_{b_j}^{m_j} = V_{m_j} \cup B_j$ are optimal lopsided trees.
3. If $b_j < n \leq a_{j+1}$ for some $j \geq 0$ then $T_{b_j}^{m_j+p}$ is optimal if $n = b_j + p(k-1)$ and $T_{a_j}^{m_j+p}$ or $T_{b_j}^{m_j+p+1}$ is optimal if $n = b_j + p(k-1) + q$ for $q < k-1$.

This characterization can be also applied to formulate an algorithm to construct an optimal tree $T_n$ in $O(n \log r)$ time which is better than previous algorithms.

After building the optimal trees, the authors of [9] analyze lopsided trees. In particular, they present asymptotic analysis of $F(x)$ (the number of nodes in $A_x = \{v : \text{depth}(v) \leq x\}$), $L(x)$ (the number of leaves in $A_x = \{v : \text{depth}(v) \leq x\}$), and the minimum height of a tree with $n$ leaves and the cost $C(T_n)$ (resp. the cost of Varn codes of $n$ words).

Let us describe the analysis of $F(x)$, that is, the number of nodes of depth no bigger than $x$. It is easy to see that $F(x)$ satisfies the following equation

$$
F(x) = \begin{cases} 
1 + F(x - c_1) + \cdots + F(x - c_r) & \text{if } x \geq c_1 \\
1 & \text{if } 0 \leq x \leq c_1 \\
0 & \text{if } x < 0.
\end{cases}
$$

This functional equation can be solved either by using Laplace's transform or the Mellin transform. The authors of [9] set $x = \ln t$ and $d_i = \ln c_i$ to reduce the above equation to the
one on \( f(t) = F(\ln t) \) for \( t > 1 \) that is accessible by the Mellin transform approach. Indeed, the Mellin transform \( f^*(s) = \int_1^\infty f(t)t^{s-1}dt \) becomes

\[
f^*(s) = \frac{1}{s(1 - d_1^s - \cdots - d_r^s)}
\]

for \( \Re(s) < -1 \). Using the inverse Mellin transform, one can extract the asymptotics of \( F(x) \) as \( x \to \infty \). In particular, it is proved in [9] that

- if \( (c_1, \ldots, c_r) \) are rationally related (i.e., for all \( 1 \leq i, j \leq r \) the ratio \( c_i/c_j \) is rational), then
  \[
  F(x) = D(x)\varphi^x + O(\rho^x), \quad x \to \infty
  \]
  where \( 1/\varphi \) is the smallest positive solution of \( 1 - x^c_1 - \cdots - x^c_r = 0 \) and \( \rho < \varphi \), and
  \[
  D(x) = \frac{d}{c}(1 - \varphi^{-d})\varphi^{-d(x/d)}
  \]
  with \( d = \gcd(c_1, \ldots, c_r) \), \( c = \sum_{i=1}^r c_i \varphi^{-a} \) and \( \{a\} = a - [a] \) is the fractional part of \( a \).

- if \( (c_1, \ldots, c_r) \) are irrationally related (i.e., for some \( 1 \leq i, j \leq r \) the ratio \( c_i/c_j \) is irrational), then
  \[
  F(x) = \frac{1}{c\ln \varphi} \varphi^x + o(\varphi^x)
  \]
  as \( x \to \infty \).

6 Dynamical Sources and Algorithms

It is quite natural to consider an algorithm together with its possible inputs as a dynamical system. The (discrete) time is related to the number of iterations. In what follows we shortly review on the realization of this idea by B. Vallée and her collaborators [1, 7, 8, 10, 56, 58, 57].

One considers a dynamical systems (or sources \( S \)) on a finite or countable alphabet \( \mathcal{M} \). Let \( T : (0,1) \to (0,1) \) be a mapping of the kind that there is a partition \( (I_m : m \in \mathcal{M}) \) of \( (0,1) \) such that the restriction of \( T : I_m \to (0,1) \) is a bijection (satisfying certain analytic properties). Then each \( x \in (0,1) \) is associated with an infinite sequence (word)

\[
M(x) = (M_1(x), M_2(x), \ldots),
\]

where \( M_j(x) = m \in \mathcal{M} \) if \( T^{j-1}(x) \in I_m \). Furthermore there is a probability distribution on \( (0,1) \) so that one can consider statistical properties of such dynamical systems.

The key element of the whole analysis is a function \( \lambda(s) \) (where \( s \) in a suitable complex neighborhood of the real interval \( I = [0,1] \)) which is the largest eigenvalue of an appropriate bounded compact operator such that an analog of the Perron-Frobenius theory can be applied. These operators are called classical \( \mathcal{G}_s \) (resp. generalized) Ruelle operators. For \( s = 1 \) the classical Ruelle operator \( \mathcal{G}_1 \) is just the density transform operator on \( (0,1) \) with respect to the mapping \( T : (0,1) \to (0,1) \) (see [57]).
Two parameters are of particular interest, namely the entropy $h(S)$ and the coincidence probability (related to the second order Rényi entropy) $c(S)$. They are related to $\lambda(s)$ via $h(S) = -\lambda'(1)$ and $c(S) = \lambda(2)$. For example, one asserts that the number $B(x)$ of finite prefixes of $M(x)$ with probability $\geq x$ is asymptotically given by

$$B(x) \sim \frac{-1}{\lambda'(1)} \frac{1}{x} = \frac{1}{h(S)x}$$

as $x \to \infty$ (see [57]).

One can apply dynamic sources and this new methodology to the analysis of tries. In such a case, it is assumed that $M(x)$ determines the infinite strings of the data keys (see [10]). One obtains that the height $H_n$ of these random tries satisfies

$$\mathbb{E} H_n \sim \frac{2}{|\log c(S)|} \log n$$

and

$$\Pr[H_n \leq k] = \exp \left( -\rho c(S)^k n^2 \right) + o(1)$$

uniformly for all integers $k \geq 0$ as $n \to \infty$ (where $\rho > 0$ is a constant depending on the source and the initial density $f$). Furthermore, the average size of such a trie is approximately $n/h(S)$ and the average path length (the sum of all depth of leaves) is approximately $n \log n / h(S)$.

Another application of this concept is the analysis of generalized pattern matchings ("hidden patterns", see [7, 24]) where the words are generated according to a dynamical source. The authors of [7] determine the mean and the variance of the number of occurrences in this generalized pattern matching problem, and establish a property of concentration of distributions. The motivation to study this problem comes from an attempt at finding a reliable threshold for intrusion detections, from textual data processing applications, and from molecular biology.

Finally, Vallée and her collaborators applied dynamic sources to various versions of the Euclidean algorithm (e.g. the binary Euclidean algorithm [56], the Lehmer-Euclid algorithm, the $\alpha$-Euclidean algorithm [8]). Again the entropy $h(S)$ governs the analysis of these algorithms. For example, one obtains that the average number of iterations $P_n$ in the Euclidean algorithm is given by

$$P_n \sim \frac{2}{h(S)} \log n$$

and the average bit complexity $C_n$ becomes

$$C_n \sim \frac{\rho}{h(S)} \log^2 n$$

as $n \to \infty$, where the constant $\rho$ is related to the mean value of the digits.

7 The Random Assignment Problem

In this section, we report on the solution of a long standing conjecture concerning the average value of the random assignment problem due to David Aldous.\(^5\) In the linear assignment

\(^5\)Aldous outlined his proof in his talk in Krynica Morska, 2002, on “Zeta(2) and the random assignment problem".
problem (LAP) a matrix \( \{a_{ij}\}_{i,j=1}^{n} \) is given and one asks for the best permutation \( \sigma \) such that

\[
A_n = \min_{\sigma} \sum_{i=1}^{n} a_{i,\sigma(i)}.
\]

In the random assignment problem the elements \( a_{ij} \) are uniformly distributed in \([0,1]\). The long standing open problem was to evaluate the average value \( \mathbb{E} A_n \).

There is another model of the LAP problem. In this representation, a complete bipartite graph \( K_{n,n} \) is given with random weights on edges that obey an exponential law of parameter 1. Let \( A_n \) be the cost of a random assignment which is the same as the cost of LAP. It has long been conjectured that

\[
\mathbb{E} A_n \sim \zeta(2) = \sum_{k=1}^{\infty} \frac{1}{k^2}.
\]

There is indeed a finite version of the conjecture, namely,

\[
\mathbb{E} A_n = \sum_{k=1}^{n} \frac{1}{k^2}.
\]

In fact, this problem has been open for some 20 years: Karp [30] proved in 1983 that \( \mathbb{E} A_n < 2 \); Aldous [2] (1992) proved the existence of the limit \( \alpha = \lim \mathbb{E} A_n \) and Goemans and Kodialam [27] (1993) established that \( \mathbb{E} A_n \) is a little over 1 + e^{-1}. Mezard and Parisi [35] have a non rigorous argument based on ideas from statistical mechanics that \( \mathbb{E} A_n \to \pi^2/6 \). Aldous developed the ideas of an approach to proving the infinite \( n \) conjecture—this by viewing it as an infinite matching problem. This gives already the improved upper bound \( \mathbb{E} A_n \leq \zeta(2) \) and there was good hope that the infinite \( n \) conjecture will succumb. Indeed, it did. After our seminar Aldous submitted a complete proof and it was recently published in [4].

There are several interesting points in Aldous’ lecture commented by Philippe Flajolet in his post-conference Research Notes. First, the general approach of the probabilistic methods consists in designing an infinite (continuous) model in which the finite scale models are immersed; see Aldous’ continuum random tree [3]. This is dual to analytic-combinatorial methods that aim at an exact modeling by generating function complemented by subsequent asymptotic analysis: “First approximate, then analyze!” versus “First analyze then approximate!” Second, Aldous spent quite some time during his talk advocating “pure thought” proofs: this is the way he envisions the probabilistic approach. This made Flajolet wonders, however, as to the amount of technology that is needed. Flajolet’s impression was that everything is in the eye of the beholder, and perhaps what is “pure thought” for some is hard work for others? Conversely, perhaps, analysts should devote more time structuring proofs by taking the “pure thought” motto as an inspiration?

A last fact regarding this motivating lecture. One may consider the analogous problem of the cost of a minimal spanning tree of \( K_n \) with edge weights that are uniform \((0,1)\). Frieze [26] showed in 1985 that the expected cost tends to \( \zeta(3) \) as \( n \to \infty \). Is there a finite \( n \) version of Frieze’s result?

---

6 They were published in August 2000 on the AofA web page.

8 Coalescing Saddle Points

We finally comment on an analytic method that has appeared in several applications, namely on coalescing saddle points and the Airy function. For many years, there had been good reason to suspect that Airy functions play a role in quantifying certain transition regions of random combinatorics. The Airy function can be defined either as a solution of the differential equation \( y'' - z y = 0 \) or by the integral representation

\[
\text{Ai}(z) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} e^{i\left(t + t^3/3\right)/3} dt = \frac{1}{\pi^{3/2}} \sum_{n=0}^{\infty} \frac{\Gamma((n + 1)/3)}{n!} \sin \left( \frac{2(n + 1)\pi}{3} \right) \left(3^{1/3}z\right)^n.
\]

(12)

It is thus the prototype of integrals involving the exponential of a cubic.

Many limit distributions of analytic combinatorics are known to be attainable through perturbation of a singularity analysis or a saddle point analysis. The approximations are of an exponential quadratic form, \( e^{-x^2} \), which usually leads to Gaussian laws. However, when there is some confluence of singularities or some “coalescence” of saddle points, approximations of a more complicated form should be sought. Precisely, coalescence of two saddle points is known in applied mathematics to lead to expressions involving the Airy function.

We first observe that some complications may arise with straightforward saddle point method. For example, imagine that the integral \( I(n, \alpha) \) defined

\[
I(n, \alpha) = \int f(z) e^{-nh(z, \alpha)} dz.
\]

depends on the parameter \( \alpha \) such that for \( \alpha \neq \alpha_0 \) there are two distinct saddle points \( z_+ \) and \( z_- \) of multiplicity one. For \( \alpha = \alpha_0 \) these two points coincide to a single saddle point \( z_0 \) of multiplicity two. Therefore, (under appropriate assumptions) for \( \alpha \neq \alpha_0 \)

\[
I(n, \alpha) \sim f(z_+) e^{-nh(z_+)} \left[ \frac{2\pi}{nh''(z_+)} \right]^{1/2} + f(z_-) e^{-nh(z_-)} \left[ \frac{2\pi}{nh''(z_-)} \right]^{1/2}
\]

For \( \alpha = \alpha_0 \) the asymptotic behavior of \( I(n, \alpha_0) \) differs radically since \( h''(z_0) = 0 \). Then one arrives at

\[
I(n, \alpha_0) \sim Af(z_0) e^{-nh(z_0)} \Gamma \left( \frac{4}{3} \right) \left[ \frac{3!}{nh''(z_+)} \right]^{1/3},
\]

where \( A \) is a constant that depend on the contour of the integration. Thus the order of \( n \) changes discontinuously from \( \frac{1}{2} \) to \( \frac{1}{3} \). The interested reader is refereed to Wong [59] and [6] for more in depth discussion.

Flajolet’s talk in Krynica Morska focuses on the case of random maps. Recall that a map is a connected planar graph given together with a rigid embedding on the plane or the Riemann sphere. Consider now the core which is the largest 2-connected component of a map (this is in essence the largest submap obtained by breaking the original map at its articulation points). Then core size admits a limiting distribution that has several surprising features: the
tails are highly dissymmetric, decaying like $x^{-5/2}$ on the left and like $e^{-x^3}$ on the right. The authors of [6] propose to call this distribution the *map-Airy distribution*: it arises precisely from a confluence of two saddle points (as seen via Lagrange inversion) or, equivalently, from a certain type of confluence of singularities (in the realm of the original generating functions) and it involves the Airy function—whence the name given to the distribution. Indeed, the map-Airy distribution is found to have density

$$A(x) = 2 \exp \left(-\frac{2}{3}x^3\right) \left(x \text{Ai}(x^3) - \text{Ai}'(x^3)\right),$$

and is, in disguise, a stable law of index $\frac{3}{2}$.

The next talk in Krynica Morska by Soria put these results into the more general framework of composition of singularity schemes. The final talk in this series by Schaeffer made explicit the generality of the approach. In fact a dozen or so types of maps exhibit the distribution (13)) and this has implication in the fast random generation of maps with higher connectivity indices. Finally, readers of these pages have already heard about the Airy function, e.g., in the context of linear probing hashing [54]. As a matter of fact, there is good hope to attack the evolution of the random graph $G_{n,m}$ ($n$ vertices and $m$ edges) and of linear probing hashed tables by means of coalescing saddle points [25].

References


The Formal Specification Column

by

Hartmut EHRIG

Technical University of Berlin, Department of Computer Science
Franklinstrasse 28/29, D-10587 Berlin, Germany
ehrig@cs.tu-berlin.de

Net Transformations for Petri Net Technology

Milan Urbášek
Technische Universität Berlin
urbasek@cs.tu-berlin.de

The application-oriented presentation of Petri nets has been developed during several years in the Petri net research group at the Technical University and Humbold University Berlin. Definition of different Petri net techniques has been one of the main research areas in this project (see papers in proceedings [28] and [29]). The main concept focuses on Petri nets as a specification technique, hence, it means more than mere Petri net based model. It comprises e.g. structuring and refinement of nets, tool support, exchange formats, or process model and exemplary methodology, so that adequate and scalable use of nets is provided for specific application domains. Net transformations are very important for manipulation with Petri nets in specific Petri net techniques for their formal background and applicability. We distinguish between net model and net class transformations.

The idea of net model transformations follows the development in the area of graph transformations and its application to Petri nets as described in [8]. This idea was formally extended by the so-called Q-theory in order to handle the net model transformations preserving certain system properties in [20] and in [21]. Preservation of certain system properties like safety properties, deadlock freedom, liveness, etc. is of great importance for net system designers as these transformations make the verification of properties of the final model easier.

Net class transformations describe a change of the underlying Petri net class. They add or remove certain aspects of a net like marking, time features, etc., while preserving relations between nets represented by morphisms. The are based on the approach presented in [12,13].

Next, we describe net transformations from an informal point of view. We will give a motivation for research on net transformations and show the role of net transformations in the above mentioned Petri net technology, called >Petri Net Baukasten<.

1 The Notion of Petri Net Technology

Within the last four decades of research on Petri nets numerous Petri net notions and methods as well as tools and tool environments have evolved. These have been successfully employed in various application areas, such as automatic production, control
systems, workflow management, etc. In such large scale of applications different Petri net variants, called Petri net classes, can be employed. A Petri net class represents a Petri net variant including a set of techniques based on that variant like structuring, analysis, and verification techniques. To identify adequate Petri net classes and methods for use of Petri nets within the system development process in a specific application domain is still a difficult task. Hence, there is a strong need for a structured approach to various Petri net techniques comprising methodological procedures, tool support, and formal techniques. A structured presentation of various methods and techniques is called a technology. The strong motivation for such a technology derives from the rich and diverse Petri net theory and its various applications [24,15,25].

In this column we refer to a technology, called >Petri Net Baukasten<, which has been previously presented in [28,6]. The >Petri Net Baukasten< has been developed within the joint research project “DFG-Forscherguppe PETRINETZ-TECHNOLOGIE” between H. Weber (Coordinator), H. Ehrig (Technische Universität Berlin) and W. Reisig (Humboldt Universität zu Berlin), supported by the German Research Council (DFG).

The >Petri Net Baukasten< provides a classification of Petri nets and corresponding notions, which is independent of their use in applications, their formalizations, and tool support. The semi-formal classification is given by class diagrams that describe Petri net class and its notions. It is represented using UML and constitutes the base of the >Petri Net Baukasten<.

The >Petri Net Baukasten< enables more straightforward understanding of Petri net classes and the development of Petri net tools, to aid in the application of Petri net classes, and to provide a unified representation of the formal definitions of Petri net classes. The >Petri Net Baukasten< includes a classification concept for Petri nets that serves these purposes.

These representations of Petri net notions are given in specific views. They concern the use of the Petri net notions within a certain application domain, their formalization in the theory of Petri nets, and their implementation as Petri net tools. These views are called Application Developer View, Expert View, and Tool Developer View, respectively. These techniques allow visualization, formal description, early evaluation, and verification of models based on nets.

Generally, we consider a Petri net technology to be a Petri net based manner of accomplishing the task of system development using methods for employing Petri net techniques. A method deals with principles of employing Petri net techniques in applications.

2 Motivation and Requirements for Net Transformations

For the development of concurrent and distributed systems Petri nets [23] are a well-known specification technique. They are successfully employed in many different application areas like workflow management, traffic control, automated production etc. (see e.g. [5,14,15]). Their intuitive graphic representation on the one hand and good mathematical foundation on the other hand are main reasons for employing them in modeling, verification, and simulation of systems. Therefore, numerous variants have
been proposed for specific purposes and application areas, each supplied with its own theoretical foundation. These variants differ in their complexity and expressivity, as well as in available operations like structuring, analysis, and verification techniques.

2.1 Motivation for Net Transformations

A design of Petri net based models employs a methodology how and in which order the Petri nets are to be used.

The common approach of the Petri net based process models is to start with a simple model which is gradually enhanced. The final model usually comprises causalities among the system’s actions as well as the functionality of the entire system. This includes data, time aspects, exception handling, business rules, etc. Thus, although significantly differing in paradigms, enhancement is of central importance. We make the enhancement explicit leading to the concept of transformation.

Petri net based models in literature, see e.g. in [19,16,4,17,18,2,3], propose enhancing a simple start model in order to proceed from analysis to design phase. The enhancement of models is given mainly on an informal level in these papers. Enhancement comprises addition of new aspects like data and time as well as addition of details like exception handling and business rules. The enhancement leads to different Petri nets for different phases in the development process. With respect to the used Petri net formalisms there are two opposed paradigms: [19,16,18,2,3] basically use one rich high-level Petri net formalism throughout the whole development process. This has the advantage of an integrated presentation of all aspects and details of a system. However, due to the complexity of the employed Petri net formalism there are few, if any, structuring, analysis, or verification techniques. Moreover, the constructed Petri nets are complex and therefore difficult to understand. In contrast, [4,17] propose varying Petri net formalism according to the development phase. The used Petri net formalism comprises only those aspects which are relevant in a given development phase. The advantage is that Petri net models are simpler and generally there are more analysis, verification, and structuring techniques available, see [6,10].

According to these paradigms, enhancement of Petri net models differs: it means some modification of Petri nets (first paradigm) or it additionally comprises a change of the Petri net formalism (second paradigm).

In order to cover enhancement of Petri nets within both basic paradigms we now introduce our concept of transformations. We distinguish transformations of Petri nets on two different levels: on the level of the Petri net formalism and on the level of the Petri net. These transformations can be combined in order to enhance both the formalism and the Petri net model.

Net class transformation The Petri net formalism determines the aspects that are modeled in a corresponding Petri net. In order to enhance a model by additional aspects like data, time, or roles, the Petri net formalism has to cover new aspect. In other words, the new aspect has to be integrated into the Petri net formalism. The addition of new aspects is realized by net class transformation, which changes the underlying Petri net formalism. This change comprises enrichment of or abstraction from certain aspects of the system.
Net model transformation In contrast, enhancing a model by additional details operates on one level of abstraction. Addition of details like exception handling, refining the description of the model, etc. means specifying or changing a Petri net model, not the underlying formalism. This kind of transformation is called net model transformation. An important special case is a change of a single aspect, e.g., data.

Combined transformation In order to enhance a model by an aspect, e.g., data, we have to perform both transformations. By net class transformation in we change the net formalism resulting in formalism, which comprises data. Subsequently, we specify the actual data by net model transformation. The net model transformation yields then data of a specific system. For an example see Section 5.

Summarizing, net transformation is given as a net class transformation or a net model transformation or their combination. Apparently, this concept of transformation covers enhancement of Petri net based models described in literature, as presented above. In the first paradigm only net model transformation is used, whereas the second paradigm employs both net class and net model transformation.

2.2 Formal Foundation of Net Transformations

The formal foundation of net model and net class transformations is different for each kind of transformation because they operate on different abstraction levels. The underlying theories are category theory, see e.g., [1], for net class transformation and the theory of high-level replacement systems, see e.g., [9, 8], for net model transformation.

Net class transformation Technically, net class transformation is achieved by a functor between the appropriate categories of Petri nets. Functors are known from category theory as some kind of mapping between categories. Intuitively, a functor is a mapping, which assigns a Petri net of the new formalism to each Petri net of the old formalism and which is compatible with relations between Petri nets given by morphisms.

Net model transformation To realize model transformation we employ the concept of rules and applications of rules. This approach has been originally introduced for graphs in [7] and generalized to all kind of structures — including Petri nets — in high-level replacement systems [9, 8]. Conceptually, rules are given by \( r : L \rightarrow R \) with two nets \( L \) and \( R \), called left-hand side and right-hand side of the rule, respectively. Application of a rule to a Petri net informally means replacing the left-hand side of the rule by the right-hand side. This notion comprises any kind of change of the model including replacement of the whole system.

Combined Transformation A combined transformation is given by an arbitrary sequence of net class and net model transformation. The results of one (net class or net model) transformation is transformed further by a subsequent transformation.

3 The Role of Net Transformations in the Petri Net Technology

The Petri net technology mentioned in Section 1 is considered as a unified presentation on theory, applications, and tools of Petri nets. In this section we will describe the role
of net transformations within the Petri net technology as used in the Petri Net Baukasten.

Petri net transformations (or succinctly net transformations) are used to perform modifications of a net. They are formalized on a rigorous mathematical foundation, see [20] and [12,13]. For a systematic study two levels of transformations, called net model transformations and net class transformations are distinguished, as discussed in the foregoing section. The purpose of the formal transformations is twofold:

- On the one hand net class transformations extend the theory of a given net class in the following sense: Petri net operations in the target net class are made available also for the source net class by transforming the net class, performing the operation, and subsequently interpreting back the result of the operation in the source net class.

- On the other hand both kinds of transformations together allow arbitrary modifications of a (start) net. Therefore, they are suitable to support stepwise enhancement of nets in the context of system development. In this sense, they yield a formal support of the models based on nets.

The net transformations are essential for realizing stepwise system development within the given process models. These process models present a methodology in the way Petri nets are to be used in the different steps during development of models based on nets. They typically start with an abstract model of the system which is refined in further development steps. Refinement here means integration of system aspects like time, reactivity, roles, data, etc. as well as modification of these aspects for incorporating exception handling, etc.

The main idea of stepwise development of systems is to offer to an application developer a number of net transformation techniques. These techniques describe how to change a net model in order to obtain another more elaborated, refined and expressive net model.

The sequence of net transformations provides a transformation from the initial net $G$ to the final net $H$:

$$G = G_0 \xrightarrow{P_1} G_1 \xrightarrow{P_2} \ldots \xrightarrow{P_n} G_n = H.$$  

Each step $G_i \xrightarrow{P_{i+1}} G_{i+1}$ denotes a single net class or net model transformation. The transformation process is either rule-based (net model transformation) or based on functors (net class transformation).

The preservation of suitable system or structural properties during the transformation process is of interest in most applications, since the final model may become very large and hence difficult to check for certain properties. It is of great importance for the application developer to check only the initial – usually quite small system – directly and to apply property preserving net transformations. For this reason structure and property preserving transformations are supported by net transformations and other net techniques, especially by rule-based refinement, horizontal structuring techniques (union and fusion), property preserving net model transformations, and formal transformations of net classes.
4 Informal Description of Net Transformations

In this section we describe the ideas used in the theory of net transformations informally. The explanatory example is presented in Section 5 afterwards. Formal definitions of net transformations are given in [20] and [12,13] in detail.

The net model transformations are defined as rule-based. Each net model transformation represents a specific application of a given rule on a transformed net. Net class transformations are defined as functors.

Generally, rules and net model transformations of Petri nets are given by instantiation of high-level replacement systems. These can be considered as general description of replacement systems, where the left-hand side of the rule is replaced by the right-hand side. Rules and transformations fully capture the replacement and thus can define any kind of system development or modification. High-level replacement systems have been introduced in [9] as a categorical generalization of graph transformations. The application of high-level replacement systems to different domains as place/transition nets, algebraic specification [8], etc. requires a suitable category. High-level replacement systems are formulated for an arbitrary category NET with a distinguished class \( M \) of morphisms called \( M \)-morphisms. Here we give the corresponding notions of replacement systems in terms of Petri nets and not on the abstract level as in [21].

We now explain the used notions rather informally to show that the complex category theory behind them can be omitted in applications. The most important notion from category theory used almost throughout whole theory of high-level replacement systems is the notion of a pushout. In the next paragraph we describe the pushout construction. For illustration see also Figure 1. Next we briefly address other necessary notions as rules, net model and net class transformations, property preserving rules, structuring techniques and proof rules.

**Pushouts of Petri nets.** A pushout is a categorical construction that requires a commutative square and has some universal properties. Informally, a pushout can be characterized as the "largest" object, that yields a commutative square for two given morphisms without "new" nodes. A pushout of two Petri nets can be considered as a union of nets with respect to a common interface. That is, for objects and injective morphisms as on the top and left of Figure 1 we glue the nets together as illustrated in the same figure and we achieve right and bottom morphisms with the common object. For other net classes the construction of pushouts has to take additional components like e.g. the data, arc inscription, transition guards, etc. into account. Basically the construction of these components is analogous to the presented construction of the net structure.

![Figure 1. An example of a pushout](image-url)
Rules. A rule \( r = (L \xleftarrow{L'} K \xrightarrow{r} R) \) consists of the Petri nets \( L, K \) and \( R \), called left-hand side, interface, right-hand side, respectively, and two \( M \)-morphisms \( K \xrightarrow{l} L \) and \( K \xleftarrow{r} R \).

Net model transformations. Given a rule \( r = (L \xleftarrow{k_1} K \xrightarrow{k_2} R) \) a direct transformation \( N_1 \xrightarrow{r} N_2 \), from \( N_1 \) to \( N_2 \) is given by the following two pushout diagrams (1) and (2).

The morphisms \( L \xrightarrow{l} N_1 \) and \( R \xrightarrow{r} N_2 \) are called occurrences. The net \( C \) is called pushout complement.

Informally, a rule \( r = (L \xleftarrow{L'} K \xrightarrow{r} R) \) is given by three nets \( L, K, \) and \( R \). Moreover, \( K \) is a subnet of both \( L \) and \( R \) expressed by the morphisms \( l \) and \( r \). Application of a rule to the net \( N_1 \) is a net model transformation of \( N_1 \). The net model transformation means replacing a subnet specified by the left-hand side of the rule with the net specified by the right-hand side. More precisely, we first identify the subnet \( L \) in \( N_1 \). Then we delete those parts of the subnet \( L \) which are not subnets of the interface net \( K \). This results in an intermediate net \( C \), where in a further step we add the difference of \( R \) and \( K \) to the preserved subnet \( C \) to obtain the transformed net \( N_2 \). In case the left-hand side is empty, we simply add the right-hand side to the first net.

(Horizontal) Structuring. There are two abstract structuring constructions in the theory of high-level replacement systems, namely union and fusion. Generally, they combine two subnets or two different nets into one. The union of two Petri nets is given with respect to a defined subnet. Union is defined as the pushout of two nets and is given by a span of morphisms. The resulting net preserves the common subnet, i.e. the source of both morphisms and keeps the rest of the two nets distinct, e.g. see Figure 1. The fusion is the gluing of two subnets within one Petri net.

Refinement. Based on the notion of rules and transformations, the general theory of high-level replacement systems has been enriched by the \( Q \)-theory in order to formulate abstraction/refinement morphisms of structures. These morphisms are more suitable for the stepwise development of systems. The main idea is to add an abstraction/refinement morphism to a rule going from left-hand side of a rule to the right-hand side or vice versa (see the drawing in the paragraph Property preserving net model transformations). The main advantage of this approach is the fact that the additional abstraction/refinement morphisms can be defined as preserving or reflecting certain properties. This means that certain important system properties may be preserved by transformations as defined below. The general theory of rules and transformations with additional refinement morphisms has been introduced in [20] in the general framework of high-level replacement systems.

System properties. Petri nets are an adequate specification technique for behavioral aspects of systems. So, the desired properties of the system to be specified usually concern the behavior of the model. These properties can be expressed in various ways, e.g in terms of Petri nets (as liveness, boundedness etc.), in terms of logic (e.g. temporal
logic, logic of actions etc.) in terms of relation to other models (e.g. bisimulation, correctness etc.) and so on. Up to now we have focused on liveness of Petri nets and on safety properties in the sense of temporal logic. Liveness of nets means that no deadlock and even livelock of a net can occur, i.e. there always exists a firing sequence which enables any chosen transition from any reachable marking. A safety property is expressed by a logic formula stating facts about markings of a net. A formula is given in terms of numbers of tokens on places. For a place/transition net the static formula $2d \land 3a$ is true for a marking $m$ where at least 2 tokens are present on the place $d$ and at least 3 tokens on the place $a$. The always operator $\Box$ in a safety property $\Box(2d \land 3a)$ requires that the static formula $(2d \land 3a)$ is true for all reachable markings from $m$.

Q-Rules. A pair $(r, q)$ is a property preserving rule, if $r = (L \leftarrow K \rightarrow R)$ is a rule with morphisms $l, r \in \mathcal{M}$ and with

- either a property preserving morphism $q : L \rightarrow R$ s.t. $q \circ l = r$,
- or a property respecting morphism $q : R \rightarrow L$ s.t. $q \circ r = l$.

According to the notion of property preserving morphisms and rules, we can now define property preserving transformations. The general idea is that the application of a property preserving rule leads to a net transformation that also preserves this property.

Property preserving net model transformations. Given a property preserving rule $(r = (L \leftarrow K \rightarrow R), q)$ with $q : L \rightarrow R$ being a property preserving morphism ($q : R \rightarrow L$ being property respecting, respectively). Then the direct net model transformation

\[ N_1 \xrightarrow{(r,q)} N_2 \]

is a property preserving transformation\(^1\) with a property preserving (respecting, respectively) morphism $\overline{q} : N_1 \rightarrow N_2$ ($\overline{q} : N_2 \rightarrow N_1$, respectively). The graphical representation of such net model transformations is depicted on the right.

Preservation of certain system properties during transformation saves the time necessary to verify the modified net.

Net class transformations. Net class transformations are based on the notion of a functor. Informally, the net class transformations assign to each net of a specified source net class an appropriate net of the target net class. The net class transformations preserve the relations between nets, which are represented by morphisms. Net class transformation can be applied to all nets of the source class. So, it is possible to say that the source net class is transformed to the target net class. The net class transformations are depicted in a usual functorial notation, i.e. $F : \mathcal{NC}_1 \rightarrow \mathcal{NC}_2$ stands for the transformation from the net class $\mathcal{NC}_1$ to the net class $\mathcal{NC}_2$. The net class transformations are developed to be compatible with other techniques as horizontal structuring and net model

\(^1\) provided the morphisms satisfy certain assumptions
transformations. This is important for transferring the results (as stepwise development) achieved within one net class to another net class.

5 Explanatory Example

In this section the explanatory example is shown, which presents the application of net transformations on a simple system. We will show the use of net model transformations first, following by the discussion of property preservation. We will conclude the example by showing the use of net class transformations.

To illustrate the net transformations, a simple communication based system is constructed as interconnection of three components: a buffer with two tasks, a printer, and a communication unit (network) between buffer and printer. The communication unit consists of a secure and non-secure channel. All components are depicted in the Figure 2. Buffer and printer are modeled using elementary nets, communication unit by place/transition net (arc weights are used).

![Figure 2. Components of the system](image)

The behavior of the printer and the buffer are obvious from the figure. The communication unit serves a secure or non-secure communication from the buffer to the printer.

This unit receives a task which has to be sent over the transition (GT). It can send the message forward through a secure channel (SSC) or via a non-secure one (NSC0, NSC1, NSC2). On the non-secure channel the message may become corrupted. Therefore, when a non-secure channel is used, two copies (C1, C2) of the mes-
sage are to be sent and a transmission subunit waits (W) for an acknowledgment. A receiving subunit on the other end receives both copies and compares them (CP). (We assume that a message cannot be lost during the transfer.) If both copies are the same, the OK acknowledgment is sent back to transmission subunit and the communication will end. If both copies differ, then NOK acknowledgment is sent back to transmission subunit and the transmission subunit resends (RS) the message again (in two copies). The two possible results of the comparison are modeled by a nondeterministic choice (conflict) in place D. The communication ends when a successful transfer is performed (RT).

5.1 Net Model Transformations

The net model transformations are used in the development process to build up the whole system from its components.

Thus, the three components of the system shown in Figure 2 are interconnected by the application of the rules shown in Figure 3. In a first step we apply the buffer-printer rule in Figure 3(a) to the buffer and printer components in Figure 2. The corresponding net model transformation is shown in Figure 3(a).

The rule buffer-printer is shown in the top row of Figure 3(a). It consists of a left-hand side net $L$, a right-hand side net $R$, and an interface net $I$. The transformation from net $N_1$ to net $N_2$ via this rule is constructed in two steps. In the first step we apply $L$ to $N_1$ and remove all items of $L$ in $N_1$ which do not belong to the interface $I$. It leads to the intermediate net $N_0$. In the second step we glue together nets $N_0$ and $R$ via the interface $I$ leading to the net $N_2$. It is important to note that also $N_1$ can be considered as a result of a gluing construction, namely the gluing of $N_0$ and $L$ along $I$ such that the transformation in Figure 3(a) consists of two gluing constructions shown in diagrams (1) and (2) respectively according to the general construction of net model transformations (see [20]).

In a similar way we can apply the two other rules in Figures 3(b),(c) to net $N_2$ combined with the communication unit net in Figure 2 leading to the net in Figure 4.

Finally, we can apply the rule in Figure 5 modeling mutual exclusion between the secure and the non-secure channel leading to the final model in Figure 6.

One wants to show suitable safety properties for the final model in Figure 6 from corresponding properties of the basic components in Figure 2 and the fact that the net model transformations from the basic components to the final model in Figure 6 considered above are safety properties preserving.

In our example the system has the safety property

$$\square((\text{NOK} \lor \text{OK}) \rightarrow W)$$

for the communication unit in Figure 2(c). This property expresses a fact that, independently of the result NOK or OK of the comparison of the two copies C1 and C2 modeled by nondeterministic choice in place C, the transmission subunit waits for acknowledgment in place W.

For the printer net in Figure 2 we have the safety property that at most one job is processed at a time, i.e.

$$\square((P \ xor PP \ xor PC) \land \neg(P \land PP \land PC)),$$
Figure 3. Interconnection of components
Figure 4. Simple tasks' model

Figure 5. Modeling exclusivity

Figure 6. Final model
where P, PP and PC stand for Printer, Printing prepared and Printing completed, respectively and xor for exclusive or operator.

Surely, we would like to keep these safety properties valid during net model transformation. There are two classes of rules available for the application developer which preserve safety properties for P/T Petri nets, namely transition gluing and place preserving rules, see [22].

The main result concerning safety property preserving net model transformations depicted in [22] states that for each net model transformation sequence \( N_1 \rightarrow^* N_2 \) via safety property preserving rules, where \( N_1 \) satisfies a safety property \( \square \varphi \), we can conclude that net \( N_2 \) satisfies a corresponding translated safety property \( T(\square \varphi) \). In our small case study the rules shown in Figure 3 are transition gluing rules and the rule in Figure 5 is place preserving. This implies that the two safety properties for the communication unit and the printer considered above are also true in the final model in Figure 6.

Let us recall that a net is called live if for each reachable marking \( m \) and each transition \( t \) there is some other marking \( m' \) reachable from \( m \) such that \( t \) is enabled under \( m' \). We want to show via liveness preserving net model transformations that the final model in Figure 6 is live (see main result concerning liveness preserving transformations in [11,27]). For this purpose we consider the simple model of printing tasks in Figure 7, where it is easy to check directly that this net is live. The printing-refinement and transmission-refinement rules in Figure 8 are liveness preserving rules and can be applied to the net in Figure 7 leading to the final model in Figure 6. Since the net in Figure 7 is live and the rules liveness preserving, the final model is live as well.

5.2 Net Class Transformations

Net class transformations can be used to begin the modeling in an abstract class with simple nets and to switch to a more expressive class in later development steps. Moreover, it is important for the development process, that net class transformations are compatible with net model transformations and with the horizontal structuring techniques union and fusion.

In the following we present a development process to derive an algebraic high-level net \( AHL_4 \) in Figure 11, via the place/transition net \( PT_3 \) in Figure 6 (without marking) from the simple elementary net \( EN_1 \) in Figure 9 using net model transformations from previous section and net class transformations Weight: \( EN \rightarrow PT \) and Data: \( PT \rightarrow AHL \) as shown in Figure 12.

In this section we only consider nets without initial marking. The initial marking is dealt similarly. We can start modeling of our example with the elementary net \( EN_1 \) like in Figure 9. Then we can use an elementary net version \( printref_{EN} \) of the printing-
(a) Printing-refinement

(b) Transmission-refinement

Figure 8. Liveness preserving transition refinement
refinement used in the previous section to elaborate the modeling of the printer. This results in the elementary net $EN_2$ in Figure 10.

Elementary nets are similar to place/transition nets but do not involve arc weights in the description of the net. To use the transmission-refinement $transrefPT$ from Figure 8(b) as well, we need to transform the model to the class of place/transition nets, because the refinement adds an arc with weight 2. The net class transformation $Weight$ assigns the weight 1 to each arc leading to place/transition nets $PT_1 = Weight(EN_1)$ and $PT_2 = Weight(EN_2)$ and a net model transformation $printrefPT$ between them. After the application of the transmission-refinement $transrefPT$ we get the same model $PT_3$ as in Figure 6 without marking. In order to be able to distinguish between different tasks we first transform our model to the class of algebraic high level nets. We obtain algebraic high level nets $AHL_1 = Data(PT_1)$, $AHL_2 = Data(PT_2)$ and $AHL_3 = Data(PT_3)$, where the net class transformation $Data: PT \to AHL$ adds a trivial data type specification and algebra (see [12, 26]). The model transformations $printrefPT$ and $transrefPT$ can be transformed as well yielding transformations $printrefAHL$ and $transrefAHL$ in Figure 12.

Then we can further refine the net $AHL_3$ by an AHL net model transformation $specrefAHL$, which changes the arc inscriptions of $AHL_3$ to the ones of $AHL_4$ given in Figure 11 and adds the following specification with a suitable algebra.

\[
BOOL^+ \\
\text{sorts: Printer, Task, Comm, Result} \\
\text{ops: } printer \to Printer \\
\text{com: } \to Comm \\
\text{task}_1 : \to Task \\
\text{task}_2 : \to Task \\
\text{equ: Task Task } \to Bool \\
\text{ok: Task } \to Result \\
\text{nok: } \to Result \\
\text{get: Result } \to Task \\
\text{vars: } t: Task, r: Result \\
\text{eqns: } \text{get(ok(t))} = t \\
\text{equ(t,t)=true}
\]

In the net $AHL_4$ we are able to model the comparison of the two tasks sent through the insecure channel by inscribing the transition $CP$ with the equation

\[
r = \text{if } \text{equ}(t', t'') \text{ then } ok(t') \text{ else } nok.
\]

For simplicity we assume that our non-secure channel either transmits the duplicated task in a correct way ($ok$-case, $t' = t'' = t$) or $t'$ and $t''$ are modifications of $t$. 
with \( t' \neq t'' \) (nok-case). In the second case the transmission is repeated with the old value \( t \) of the task.

The whole development process is depicted in Figure 12, where horizontal steps are net model transformations and vertical steps are net class transformations. The used net class transformations are elaborated formally in [12] or [26].

\[
\begin{align*}
&EN \\
\downarrow & Weight \\
&PT \\
\downarrow & Data \\
&AHL \\
\end{align*}
\]

\[
\begin{align*}
&EN_1 \xrightarrow{\text{printref}_{EN}} EN_2 \\
\downarrow & \\
&PT_1 \xrightarrow{\text{printref}_{PT}} PT_2 \xrightarrow{\text{transref}_{PT}} PT_3 \\
\downarrow & \\
&AHL_1 \xrightarrow{\text{printref}_{AHL}} AHL_2 \xrightarrow{\text{transref}_{AHL}} AHL_3 \xrightarrow{\text{specref}_{AHL}} AHL_4 \\
\end{align*}
\]

**Figure 12. Development of the printing system**

6 Conclusion

We conclude with several remarks to the area of net transformations.

We have described net transformations informally in this contribution and demonstrated the applicability of this approach on a simple example. Nevertheless, the theory of net transformations is a wide theory with several important results.

The formal theory of net transformations is formally based on category theory. It presents net transformations either as rule-based modification of nets (net model transformations) or as functors between categories of nets (net class transformations). This theory offers many important results for Petri net experts, e.g. several compatibility results, preservation of structural and system's properties.

The complexity of the theory can be kept hidden in applications. The concept of transformations can be turned to algorithms easily. With tool support the user-driven application of net transformations would supply a methodology for modeling Petri net based systems by net transformations. Thus, net transformations are of interest for system designers as well.

There are several open questions to the future left. The most important task is development of a Petri net tool supporting the given transformations fully.

References


Every third year the Conference on Computational Complexity is held in Europe and this summer the University of Aarhus (Denmark) will host the meeting July 7-10. More details at the conference web page

http://www.computationalcomplexity.org

This month we present a historical view of computational complexity written by Steve Homer and myself. This is a preliminary version of a chapter to be included in an upcoming North-Holland Handbook of the History of Mathematical Logic edited by Dirk van Dalen, John Dawson and Aki Kanamori.

A Short History of Computational Complexity

Lance Fortnow1
NEC Research Institute
4 Independence Way
Princeton, NJ 08540

Steve Homer2
Computer Science Department
Boston University
111 Cummington Street
Boston, MA 02215

1 Introduction

It all started with a machine. In 1936, Turing developed his theoretical computational model. He based his model on how he perceived mathematicians think. As digital computers were developed in the 40’s and 50’s, the Turing machine proved itself as the right theoretical model for computation.
Quickly though we discovered that the basic Turing machine model fails to account for the amount of time or memory needed by a computer, a critical issue today but even more so in those early days of computing. The key idea to measure time and space as a function of the length of the input came in the early 1960's by Hartmanis and Stearns. And thus computational complexity was born.

In the early days of complexity, researchers just tried understanding these new measures and how they related to each other. We saw the first notion of efficient computation by using time polynomial in the input size. This led to complexity’s most important concept, NP-completeness, and its most fundamental question, whether $P = NP$.

The work of Cook and Karp in the early 70’s showed a large number of combinatorial and logical problems were NP-complete, i.e., as hard as any problem computable in nondeterministic polynomial time. The $P = NP$ question is equivalent to an efficient solution of any of these problems. In the thirty years hence this problem has become one of the outstanding open questions in computer science and indeed all of mathematics.

In the 70’s we saw the growth of complexity classes as researchers tried to encompass different models of computations. One of those models, probabilistic computation, started with a probabilistic test for primality, led to probabilistic complexity classes and a new kind of interactive proof system that itself led to hardness results for approximating certain NP-complete problems. We have also seen strong evidence that we can remove the randomness from computations and most recently a deterministic algorithm for the original primality problem.

In the 80’s we saw the rise of finite models like circuits that capture computation in an inherently different way. A new approach to problems like $P = NP$ arose from these circuits and though they have had limited success in separating complexity classes, this approach brought combinatorial techniques into the area and led to a much better understanding of the limits of these devices.

In the 90’s we have seen the study of new models of computation like quantum computers and propositional proof systems. Tools from the past have greatly helped our understanding of these new areas.

One cannot in the short space of this article mention all of the amazing research in computational complexity theory. We survey various areas in complexity choosing papers more for their historical value than necessarily the importance of the results. We hope that this gives an insight into the richness and depth of this still quite young field.
2 Early History

While we can trace the idea of "efficient algorithms" to the ancient Greeks, our story starts with the seminal 1965 paper of Hartmanis and Stearns, "On the Computational Complexity of Algorithms" [HS65]. This paper laid out the definitions of quantified time and space complexity on multitape Turing machines and showed the first results of the form given more time (or space) one can compute more things.

A multitape Turing machine consists of some fixed number of "tapes" each of which contains an infinite number of tape cells. The contents of a tape cell comes from a finite set of symbols called the tape alphabet of the Turing machine. All the tapes initially contain only a special "blank" character except for the finite input written at the beginning of the first tape. Each tape has a tape head sitting on the first character on each tape. The Turing machine also has a finite state memory to control its operations. In each step, it can move each tape independently one character left or right, read and possibly change the characters under each head, change its current state and decide whether to halt and accept or reject the input. Time is measured by the number of steps before halting as a function of the length of the input. Space is measured as the number of different character locations touched by the various heads.

The Hartmanis-Stearns paper did not develop in a vacuum. Turing [Tur36], of course, developed his notion of a computational device back in 1936. This machine model did and still does form the basis for most of computational complexity. Slightly earlier, Yamada [Yam62] studied "real-time computable functions", Myhill [Myh60] looked at linear bounded automata and Smullyan [Smu61] considered rudimentary sets. These models looked at specific time and space-bounded machines but did not give a general approach to measuring complexity.

After Hartmanis and Stearns developed the general method for measuring computational resources, one can ask how the different variations of Turing machines affect the complexity of problems. Rabin [Rab63] shows problems solvable faster by two-tape machine than by one-tape machines. Hennie and Stearns [HS66] show that a 2-tape Turing machine can simulate any constant tape machine taking only a logarithmic factor more time.

Hartmanis and Stearns show that given space functions $s_1$ and $s_2$ with a "constructibility" condition and $s_1(n) = o(s_2(n))$, i.e., $s_1(n)/s_2(n)$ goes to zero, then there are problems computable in space $s_2(n)$ but not space $s_1(n)$. The Hennie-Stearns result gives the best known time-hierarchy, getting a separation if $t_1(n) \log t_1(n) = o(t_2(n))$. These proofs use straightforward
diagonalization arguments that go back to Cantor [Can74].

Nondeterministic computation allows a Turing machine to make a choice of several possible transitions. We say the machine accepts if any collection of choices leads to an accepting state. Nondeterministic time and space hierarchies are much trickier to prove because one cannot do straightforward diagonalization on nondeterministic computations.

Savitch [Sav70] showed that problems computable in nondeterministic space \( s(n) \) are computable in deterministic space \( s^2(n) \). In 1972, Ibarra [Iba72] using translational techniques of Ruby and Fischer [RF65] used Savitch's theorem to show that there exist problems computable in nondeterministic space \( n^a \) but not space \( n^b \) for \( a > b \geq 1 \). Sixteen years later, Immerman [Imm88] and Szelepcsényi [Sze88] independently showed that nondeterministic space is closed under complement. The Immerman-Szelepcsényi result immediately gives a nondeterministic space hierarchy as tight as the deterministic hierarchy.

For nondeterministic time, Cook [Coo73] uses a more careful translation argument to show problems computable in nondeterministic time \( n^a \) but not time \( n^b \) for \( a > b \geq 1 \). Seiferas, Fischer and Meyer [SFM78] give the current best known nondeterministic time hierarchy, getting a separation if \( t_1(n + 1) = o(t_2(n)) \).

In 1967, Blum [Blu67] had his speed-up theorem: For any computable unbounded function \( r(n) \) there exists a computable language \( L \) such that for any Turing machine accepting \( L \) in time \( t(n) \) there is another Turing machine accepting \( L \) in time \( r(t(n)) \). This seems to violate the time hierarchy mentioned earlier but one must realize \( t(n) \) will not necessarily be time-constructible.

Blum's speed-up theorem holds not only for time but also for space and any other measure fulfilling a small list of axioms, which we now call Blum complexity measures.

Soon after we saw two other major results that we will state for time but also hold for all Blum complexity measures. Independently Borodin [Bor72] and Trakhtenbrot [Tra64] proved the gap theorem: For any computable unbounded \( r(n) \) there exist a computable time bound \( t(n) \) such that any language computable in time \( t(n) \) is also computable in time \( r(t(n)) \). McCreight and Meyer [MM69] showed the union theorem: Given any computably presentable list of computable time bounds \( t_1, t_2, \ldots \) such that \( t_{i+1} > t_i \) for all \( i \) then there exist a time bound \( t \) such that a problem is computable in time \( t \) if and only if it is computable in time \( t_i \) for some \( i \).

In 1964, Cobham [Cob64] noted that the set of problems computable in polynomial time remains independent of the particular deterministic ma-
chine model. He also showed that many common mathematical functions can be computed in polynomial time.

In 1965, Edmonds [Edm65b] in his paper showing that the matching problem has a polynomial-time algorithm, argues that polynomial-time gives a good formalization of efficient computation. He noted the wide range of problems computable in polynomial time and as well the fact that this class of problems remains the same under many different reasonable models of computation. In another paper, Edmonds [Edm65a] gave an informal description of nondeterministic polynomial-time. This set the stage for the $P = NP$ question, the most famous problem in theoretical computer science that we discuss in Section 3.

Several Russians, notably Barzdin and Trakhtenbrot, independently developed several of these notions of complexity during the sixties though their work was not known to the West until the seventies.

3 NP-completeness

It was in the early 1970's that complexity theory first flowered, and came to play a central role in computer science. It did so by focusing on one fundamental concept and on the results and ideas stemming from it. This concept was NP-completeness and it has proved to be one of the most insightful and fundamental theories in the mathematics of the last half century. NP-completeness captures the combinatorial difficulty of a number of central problems which resisted efficient solution and provides a method for proving that a combinatorial problem is as intractable as any NP problem.

By the late 1960's, a sizable class of very applicable and significant problems which resisted polynomial time solution was widely recognized. These problems are largely optimization problems such as the traveling salesman problem, certain scheduling problems, or linear programming problems. They all have a very large number of possible solution where there is no obvious way to find an optimal solution other than a brute force search. As time passed and much effort was expended on attempts at efficiently solving these problems, it began to be suspected that there was no such solution. However, there was no hard evidence that this was the case nor was there any reason to suspect that these problems were in any sense difficult for the same reasons or in the same ways. The theory of NP-completeness provided precisely this evidence.

Proving a problem in NP to be NP-complete tells us that it is as hard to solve as any other NP problem. Said another way, if there is any NP-
complete problem that admits an efficient solution then every NP problem does so. The question of whether every NP problem has an efficient solution has resisted the efforts of computer scientists since 1970. It is known as the P versus NP problem and is among the most central open problems of mathematics. The fact that a very large number of fundamental problems have been shown to be NP-complete and that the problem of proving that P is not NP has proved to be so difficult has made this problem and the connected theory one of the most celebrated in contemporary mathematics. The P = NP problem is one of the seven Millennium Prize Problems and solving it brings a $1,000,000 prize from the Clay Mathematics Institute [Cla00].

Quite surprisingly, one of the earliest discussions of a particular NP-complete problem and the implications of finding an efficient solution came from Kurt Gödel. In a 1956 letter to von Neumann [Har86, Sip83] Gödel asks von Neumann about the complexity of what is now known to be an NP-complete problem concerning proofs in first-order logic and asks if the problem can be solved in linear or quadratic time. In fact, Gödel seemed quite optimistic about finding an efficient solution. He fully realized that doing so would have significant consequences.

It is worth noting that in about the same period there was considerable effort by Russian mathematicians working on similar combinatorial problems trying to prove that brute force was needed to solve them. Several of these problems eventually turned out to be NP-complete as well [Tra64].

The existence of NP-complete problems was proved independently by Stephen Cook in the United States and Leonid Levin in the Soviet Union. Cook, then a graduate student at Harvard, proved that the satisfiability problem is NP-complete [Coo71]. Levin, a student of Kolmogorov at Moscow State University, proved that a variant of the tiling problem is NP-complete [Lev73].

Researchers strove to show other interesting, natural problems NP-complete. Richard Karp, in a tremendously influential paper [Kar72], proved that eight central combinatorial problems are all NP-complete. These problems included the the clique problem, the independent set problem, the set cover problem, and the traveling salesman problem, among others.

Karp's paper presented several key methods to prove NP-completeness using reductions from problems previously shown to be NP-complete. It set up a general framework for proving NP-completeness results and established several useful techniques for such proofs. In the following years, and continuing until today, literally thousands of problems have been shown to be NP-complete. A proof of NP-completeness has come to signify the
case) intractability of a problem. Once proved NP-complete, researchers turn to other ways of trying to solve the problem, usually using approximation algorithms to give an approximate solution or probabilistic methods to solve the problem in "most" cases.

Another fundamental step was taken around 1970 by Meyer and Stockmeyer [MS72], [Sto76]. They defined the polynomial hierarchy in analogy with the arithmetic hierarchy of Kleene. This hierarchy is defined by iterating the notion of polynomial jump, in analogy with the Turing jump operator. This hierarchy has proven useful in classifying many hard combinatorial problems which do not lie in NP. It is explored in more detail in Section 4.2.

Of course, all problems in the polynomial hierarchy are recursive and in fact very simple problems within the vast expanse of all recursive sets. So are there natural problems which are recursive and are not captured by the hierarchy? The answers is yes and results in the exploration of several important larger complexity classes which contain the polynomial hierarchy. One such class is PSPACE, those problems which can be solved using work space which is of polynomial length relative to the length of the problem's input. Just as with P and NP, the full extent of PSPACE is not known. PSPACE contains P and NP. It is not known if either of these conclusions are proper. Settling these questions would again be significant steps forward in this theory.

The notion of PSPACE-completeness is defined very similarly to NP-completeness, and has been studies alongside the the NP-completeness notion. Namely, a problem C is PSPACE-complete if it is in PSPACE and if any other PSPACE problem can be reduced to it in polynomial time. As is the case with NP-complete problems, PSPACE-complete problems are quite common and often arise quite naturally. Typical PSPACE-complete problems are or arise from generalized games such as hex or checkers played on boards of unbounded finite size (see [GJ79]). Beyond PSPACE lie the exponential time (EXPTIME) and exponential space complexity classes. A small number of natural problems have been shown complete for these classes (see [GJ79]), and as well EXPTIME is the smallest deterministic class which has been proved to contain NP.

4 Structural Complexity

By the early 1970's, the definitions of time and space-bounded complexity classes were precisely established and the import of the class NP and of NP-
complete problems realized. At this point effort turned to understanding
the relationships between complexity classes and the properties of problems
within the principal classes. In particular, attention was focused on NP-
complete problems and their properties and on the structure of complexity
classes between \textsc{Logspace} and \textsc{Pspace}. We briefly survey some of
these studies here.

4.1 The Isomorphism Conjecture

In the mid-70's, building on earlier work on Gödel numberings [HB75, Har82]
and in analogy with the well-known result of Myhill from computability
theory [Myh55], Berman and Hartmanis [BH77, HB78] formulated their iso-
morphism conjecture. The conjecture stated that all NP-complete sets are
P-isomorphic (that is, isomorphic via polynomial time computable and in-
vertible isomorphisms). This conjecture served as a springboard for the
further study of the structure of NP-complete sets. As evidence for their
conjecture, Berman and Hartmanis and others [MY85, KMR87] were able to
give simple, easily checkable properties of NP-complete sets which implied
they were isomorphic. Using these, they proved that all of the known NP-
complete sets were in fact P-isomorphic. This conjecture remains an open
question today. A positive resolution of the conjecture would imply that P
is not equal to NP. Much effort was focused on proving the converse, that
assuming P is not NP then the isomorphism conjecture holds. This remains
an open question today.

As the number of known NP-complete problems grew during the 1970's,
the structure and properties of these problems began to be examined. While
very disparate, they are all rather dense sets. Density of a set is measured here
simply in the sense of how many string of a given length are in the set. So
(assuming a binary encoding of a set) there are $2^n$ different strings of length
n. We say that set S is sparse if there is a polynomial p(n) which bounds
the number of strings in S of length n, for every n. It is dense otherwise.
All known NP-complete sets are dense.

One consequence of the isomorphism conjecture is that no NP-complete
set can be sparse. As with the isomorphism conjecture, this consequence im-
plies that P is not NP and so it is unlikely that a proof of this consequence
will soon be forthcoming. Berman and Hartmanis also conjectured that if
P in not equal to NP there are no sparse NP-complete sets. This con-
jecture was settled affirmatively by the famous result of Mahaney [Mah82].
Mahaney's elegant proof used several new counting techniques and had a
lasting impact on work in structural complexity theory.

4.2 The Polynomial Hierarchy

While numerous hard decision problems have been proved NP-complete, a small number are outside NP and have escaped this classification. An extended classification, the polynomial time hierarchy (PH), was provided by Meyer and Stockmeyer [Sto76]. They defined the hierarchy, a collection of classes between P and PSPACE, in analogy with Kleene's arithmetic hierarchy.

The polynomial time hierarchy (PH) consists of an infinite sequence of classes within PSPACE. The bottom (0th) level of the hierarchy is just the class P. The first level is the class NP. The second level are all problems in NP relative to an NP oracle, etc. Iterating this idea to all finite levels yields the full hierarchy.

If P=PSPACE then the whole PH collapses to the class P. However, quite the opposite is believed to be the case, namely that the PH is strict in the sense that each level of the hierarchy is a proper subset of the next level. While every class in the PH is contained in PSPACE, the converse is not true if the hierarchy is strict. In this case, PSPACE contains many problems not in the PH and in fact has a very complex structure (see, for example, [AS89]).

4.3 Alternation

Another unifying and important thread of results which also originated during the 1970's was the work on alternation initiated out by Kozen, Chandra and Stockmeyer [CKS81]. The idea behind alternation is to classify combinatorial problems using an alternating Turing machine, a generalization of a nondeterministic Turing machine. Intuitively a nondeterministic Turing machine can be thought of as having an existential acceptance criterion. That is, an input to the TM is accepted if there exists a computation path of the machine which results in acceptance. Similarly, we could consider a universal acceptance criterion whereby a nondeterministic machine accepts if all computation paths lead to acceptance. Restricting ourselves to polynomial length alternation, we see that NP can be characterized as those problems accepted by nondeterministic TM running in polynomial time using the existential acceptance criterion. Similarly, the universal acceptance criterion with the same type of machines defines the class co-NP consisting of problems whose complements are in NP. Furthermore, we can iterate
these two acceptance methods, for example asking that there exist an path of a TM such that for all paths extending that path there exists an extension of that path which accepts. This idea gives a machine implementation of the notion of alternations of universal and existential quantifiers. It is not hard to see that finitely many alternations results in the finite levels of the polynomial time hierarchy and that alternating polynomial time is the same thing as \textit{PSPACE}. Other relationship between time and space classes defined using alternation can be found in [CKS81], for example, alternating log space = \textit{P} and alternating \textit{PSPACE} = \textit{EXPTIME}.

4.4 Logspace

To this point all the complexity classes we have considered contain the class \textit{P} of polynomial time computable problems. For some interesting problems it is useful to consider classes within \textit{P} and particularly the seemingly smaller space classes of deterministic log space, denoted \textit{L}, and nondeterministic log space, denoted \textit{NL}. These classes provide a measure with which to distinguish between some interesting problems within \textit{P}, and present interesting issues in their own right.

At first glance logarithmic space is a problematic notion at best. An input of length \( n \) takes \( n \) squares by itself, so how can a computation on such an input take only \( \log n \) space? The answer lies in changing our computation model slightly to only count the space taken by the computation and not the space of the input. Formally, this is done by considering an “off-line Turing machine.” This is a (deterministic or nondeterministic) Turing machine whose input is written on a special read-only input tape. Computation is carried out on read-write work tapes which are initially blank. The space complexity of the computation is then taken to be the amount of space used on the work tapes. So in particular this space can be less than \( n \), the length of the input to the computation. We define logspace, \textit{L}, to be the class of languages decided by deterministic Turing machines which use at most \( O(\log n) \) tape squares. Similarly, \textit{NL} is defined using nondeterministic Turing machines with the same space bound.

It is straightforward to check that \textit{L} \( \subseteq \textit{NL} \subseteq \textit{P} \), and these three classes are thought to be distinct. There are a number of nontrivial problems solvable in \textit{L} (for example see [LZ77]) as well as problems known to be in \textit{NL} which are not believed to be in \textit{L} (for example see [Sav73, Jon75]). Numerous problems in \textit{P} are thought to lie outside of \textit{L} or \textit{NL}. For example, one such problem is the circuit value problem, the problem of determining the value of a Boolean circuit, given inputs to the circuit. The circuit value
problem is one of many problems in \( \mathbb{P} \) which is known to be \( \mathbb{P} \) complete. These are problems in \( \mathbb{P} \) which are proved to be complete with respect to log-space bounded reductions, reductions defined analogously to polynomial time bounded reduction in the previous section. Proving a \( \mathbb{P} \)-complete problem is in \( \mathbb{L} \) would imply that \( \mathbb{L} = \mathbb{P} \).

4.5 Oracles

Oracle results play a unique role in complexity theory. They are metamathematical results delineating the limitations of proof techniques and indicating what results might be possible to achieve and which are likely beyond our current reach. Oracle results concern relativized computations. We say that a computation is carried out "relative to an oracle set \( O \)" if the computation has access to the answers to membership queries of \( O \). That is, the computation can query the oracle \( O \) about whether or not a string \( x \) is in \( O \). The computation obtains the answer (in one step) and proceeds with the computation, which may depend on the answer to the oracle query.

The first, and still most fundamental oracle results in complexity were carried out by Baker, Gill and Solovay [BGS75]. They proved that there is an oracle reactive to which \( \mathbb{P} = \mathbb{NP} \) and another oracle relative to which \( \mathbb{P} \) and \( \mathbb{NP} \) differ.

What do these results say about the \( \mathbb{P} \) vs \( \mathbb{NP} \) question? They say little about the actual answer to this question. The existence of an oracle making a statement \( S \) true is simply a kind of consistency result about \( S \). It says that the statement is true in one particular model or "world" (that is, the oracle set itself). As such, we can conclude that a proof of the negation of \( S \) will not itself relativize to any oracle. Thus, as many proof methods do relativize to every oracle, an oracle result provides a limitation to the possible methods used to prove \( S \) and hence are evidence that the result is, in this sense, hard. Oracle results have been most useful in delineating theorems which are difficult to prove (i.e., those which do not relativize), from those which might more likely be settled by well-understood, relativizing proof techniques. In particular, the Baker, Gill and Solovay results concerning \( \mathbb{P} \) and \( \mathbb{NP} \) question indicate that a proof will be difficult to come by, as has indeed been the case.

Since 1978 numerous other oracle results have been proved. Techniques used to achieve these results have become quite sophisticated and strong. For instance, Fenner, Fortnow and Kurtz [FFK94] gave a relativized world where the isomorphism conjecture holds where Kurtz, Mahaney and Royer [KMR89] had showed that it fails relative to most oracles. They were the culmination
of a long series of partial results addressing this question.

There are a few results in complexity that do not relativize, mostly relating to interactive proof systems (see Section 6.1) but these tend to be the exception and not the rule.

5 Counting Classes

Another way to study \( \text{NP} \) computations is to ask how many computations paths in the computation lead to acceptance. For example, consider the satisfiability problem. Given an instance of this problem, that is a propositional formula, instead of asking if the formula has a solution (a truth assignment making the formula true), ask how many such assignments there are. In 1979, Valiant [Val79] defined the complexity class \( \#\text{P} \) as the class of functions computing the number of accepting paths of a nondeterministic Turing machine. Every \( \#\text{P} \) function is computable in polynomial space. Valiant used this class to capture the complexity of the counting version of satisfiability as well as other interesting problems such as computing the permanent function.

Counting complexity has since played an important role in computational complexity theory and theoretical computer science. The techniques used in counting complexity have significant applications in circuit complexity and in the series of recent results on interactive proof systems. (See the next section.) The two most important counting function classes are \( \#\text{P} \), described above, and \( \text{GapP} \). \( \text{GapP} \) consists of the class of functions which compute the difference between the number of accepting paths and the number of rejecting paths of a nondeterministic Turing machine. For example, the function which tells, for any propositional formula, computes the difference of the number of accepting and rejecting truth assignments is in the class \( \text{GapP} \).

Perhaps the two most important recent results in counting complexity are Toda’s theorem [Tod91] and the closure theorem of Beigel, Reingold and Spielman’s [BRS95]. Toda’s theorem asserts that one can reduce any language in the polynomial-time hierarchy to a polynomial time computation which uses a \( \#\text{P} \) function as an oracle. Hence, that in terms of complexity, hard functions in \( \#\text{P} \) lie above any problem in the polynomial time hierarchy. In 1994, Beigel, Reingold and Spielman [BRS95] proved that \( \text{PP} \) is closed under union. This result solved a longstanding open problem in this area, first posed by Gill in 1977 [Gil77] in the initial paper on probabilistic classes. It implies that \( \text{PP} \) is also closed under intersection Those interested
in further exploring counting classes and the power of counting in complexity
theory should consult the papers of Schöning [Sch90] and Fortnow [For97].

6 Probabilistic Complexity

In 1977, Solovay and Strassen [SS77] gave a new kind of algorithm for testing
whether a number is prime. Their algorithm flipped coins to help search for
a counterexample to primality. They argued that if the number was not
prime then with very high confidence a counterexample could be found.

This algorithm suggested that we should revise our notion of “efficient
computation”. Perhaps we should now equate the efficiently computable
problems with the class of problems solve in probabilistic polynomial time.
A whole new area of complexity theory was developed to help understand
the power of probabilistic computation.

Gill [Gil77] defined the class \textbf{BPP} to capture this new notion. Adleman
and Manders [AM77] defined the class \textbf{R} that represented the set of prob-
lems with one-sided randomness—the machine only accepts if the instance
is guaranteed to be in the language. The Solovay-Strassen algorithm puts
compositeness in \textbf{R}.

Babai introduced the concept of a “Las Vegas” probabilistic algorithm
that always gives the correct answer and runs in expected polynomial time.
This class \textbf{ZPP} is equivalent to those problems with both positive and
negative instances in \textbf{R}. Adleman and Huang [AH87] building on work of
Goldwasser and Kilian [GK99] show that primality is in \textbf{R} and thus \textbf{ZPP}.

Very recently, Agrawal, Kayal and Saxena [AKS02] gave a deterministic
polynomial-time algorithm for primality. If this result was known in the 70’s,
perhaps the study of probabilistic algorithms would not have progressed as
quickly.

In 1983, Sipser [Sip83] showed that \textbf{BPP} is contained in the polynomial-
time hierarchy. Gács (see [Sip83]) improves this result to show \textbf{BPP} is in
the second level of the hierarchy and Lautemann [Lau83] gives a simple proof
of this fact.

One can also consider probabilistic space classes. Aleliunas, Karp, Lipton,
Lovász and Rackoff [AKL+79] show that undirected graph connectivity
can be computed in one-sided randomized logarithmic space, a class called
\textbf{RL}. Similarly one can define the classes \textbf{BPL} and \textbf{ZPL}. Borodin, Cook,
Dymond, Ruzzo and Tompa [BCD+89] showed that undirected graph noncon-
nectivity also sits in \textbf{RL} and thus \textbf{ZPL}. Nisan and Ta-Shma [NT95]
showed that the connectivity question reduced directly to the nonconnectiv-
ity question.

### 6.1 Interactive Proof Systems

One can think of the class NP as a *proof system*: An arbitrarily powerful prover gives a proof that say a formula is satisfiable. One can generalize this notion of proof system by allowing probabilistic verification of the proof. This yields the complexity class MA. One can also consider interaction where the verifier sends messages based on her random coins. The bounded round version of this class is AM and the unbounded round version is IP. The incredible power of these interactive proof systems has led to several of the most surprising and important recent results in computational complexity theory.

Babai [Bab85] defined interactive proof systems to help classify some group questions. An alternative interactive proof system was defined by Goldwasser, Micali and Rackoff [GMR89] as a basis for the cryptographic class zero-knowledge. Zero-knowledge proof systems have themselves played a major role in cryptography.

The two models differed on whether the prover could see the verifier’s random coins, but Goldwasser and Sipser [GS89] showed the two models equivalent. Babai and Moran [BM88] showed that any bounded-round protocol needs only one question from the verifier followed by a response from the prover. Fürer, Goldreich, Mansour, Sipser and Zachos [FGM+89] showed that one can assume that for positive instances the prover can succeed with no error.

Goldreich, Micali and Wigderson [GMW91] show that the set of pairs of nonisomorphic graphs has a bounded-round interactive proof system. Boppana, Håstad and Zachos [BHZ87] show that if the complement of any NP-complete language has bounded-round interactive proofs than the polynomial-time hierarchy collapses. This remains the best evidence that the graph isomorphism problem is probably not NP-complete.

In 1990, Lund, Fortnow, Karloff and Nisan [LFKN92] showed that the complements of NP-complete languages have unbounded round interactive proof systems. Shamir [Sha92] quickly extended their techniques to show that every language in PSPACE has interactive proof system. Feldman [Fel86] had earlier shown that every language with interactive proofs lies in PSPACE.

Interactive proofs are notable in that in general proofs concerning them do not relativize, that is they are not true relative to every oracle. The classification of interactive proofs turned out not to be the end of the story
but only the beginning of a revolution connecting complexity theory with approximation algorithms. For the continuation of this story we turn to probabilistically checkable proofs.

6.2 Probabilistically Checkable Proofs

In 1988, Ben-Or, Goldwasser, Kilian and Wigderson [BGKW88] developed the multiprover interactive proof system. This model has multiple provers who cannot communicate with each other or see the conversations each has with the verifier. This model allows the verifier to play one prover off another.

Fortnow, Rompel and Sipser [FRS94] show this model is equivalent to probabilistically checkable proofs, where the prover writes down a possibly exponentially long proof that the verifier spot checks in probabilistic polynomial time. They also show that every language accepted by these proof systems lie in NEXP, nondeterministic exponential time.

In 1990, Babai, Fortnow and Lund [BFL91] show the surprising converse—that every language in NEXP has probabilistically checkable proofs. Babai, Fortnow, Levin and Szegedy [BFLS91] scale this proof down to develop “holographic” proofs for NP where, with a properly encoded input, the verifier can check the correctness of the proof in very short amount of time.

Feige, Goldwasser, Lovász, Safra and Szegedy [FGL+96] made an amazing connection between probabilistically checkable proofs and the clique problem. By viewing possible proofs as nodes of a graph, they showed that one cannot approximate the size of a clique well without unexpected collapses in complexity classes.

In 1992, Arora, Lund, Motwani, Sudan and Szegedy [ALM+98] building on work of Arora and Safra [AS98] showed that every language in NP has a probabilistically checkable proof where the verifier uses only a logarithmic number of random coins and a constant number of queries to the proof.

The Arora et. al. result has tremendous implications for the class MAXSNP of approximation problems. This class developed by Papadimitriou and Yannakakis [PY91] has many interesting complete problems such as max-cut, vertex cover, independent set, traveling salesman on an arbitrary metric space and maximizing the number of satisfiable clauses of a formula.

Arora et. al. show that, unless P = NP, every MAXSNP-complete set does not have a polynomial-time approximation scheme. For each of these problems there is some constant $\delta > 1$ such that they cannot be approximated within a factor of $\delta$ unless P = NP.

Since these initial works on probabilistically checkable proofs, we have
seen a large number of outstanding papers improving the proof systems and getting stronger hardness of approximation results. Hästad [Hä97] gets tight results for some approximation problems. Arora [Aro98] after failing to achieve lower bounds for traveling salesman in the plane, has developed a polynomial-time approximation algorithm for this and related problems.

A series of results due to Cai, Condon, Lipton, Lapidot, Shamir, Feige and Lovász [CCL92, CCL90, CCL91, Fei91, LS91, FL92] have modified the protocol of Babai, Fortnow and Lund [BFL91] to show that every language in NEXP has a two-prover, one-round proof systems with an exponentially small error. This problem remained so elusive because running these proof systems in parallel does not have the expected error reduction [FRS94]. In 1995, Raz [Raz98] showed that the error does go done exponentially when these proofs systems are run in parallel.

6.3 Derandomization

If you generate a random number on a computer, you do not get a truly random value, but a pseudorandom number computed by some complicated function on some small, hopefully random seed. In practice this usually works well so perhaps in theory the same might be true. Many of the exciting results in complexity theory in the 1980’s and 90’s consider this question of derandomization—how to reduce or eliminate the number of truly random bits to simulate probabilistic algorithms.

The first approach to this problem came from cryptography. Blum and Micali [BM84] first to show how to create randomness from cryptographically hard functions. Yao [Yao90] showed how to reduce the number of random bits of any algorithm based on any cryptographically secure one-way permutation. Hästad, Impagliazzo, Levin and Luby [HILL99] building on techniques of Goldreich and Levin [GL89] and Goldreich, Krawczyk and Luby [GKL93] show that one can get pseudorandomness from any one-way function.

Nisan and Wigderson [NW94] take a different approach. They show how to get pseudorandomness based on a language hard against nonuniform computation. Impagliazzo and Wigderson [IW97] building on this result and Babai, Fortnow, Nisan and Wigderson [BFNW93] show that BPP equals P if there exists a language in exponential time that cannot be computed by any subexponential circuit.

For derandomization of space we have several unconditional results. Nisan [Nis92] gives general tools for derandomizing space-bounded computation. Among the applications, he gets a $O(\log^2 n)$ space construction for
universal traversal sequences for undirected graphs.

Saks and Zhou [SZ99] show that every probabilistic logarithmic space algorithm can be simulated in $O(\log^{3/2} n)$ deterministic space. Armoni, Ta-Shma, Wigderson and Zhou [ATWZ97] building on work of Nisan, Szemerédi, and Wigderson [NSW92] show that one can solve undirected graph connectivity in $O(\log^{4/3} n)$ space.

7 Descriptive Complexity

Many of the fundamental concepts and methods of complexity theory have their genesis in mathematical logic, and in computability theory in particular. This includes the ideas of reductions, complete problems, hierarchies and logical definability. It is a well-understood principle of mathematical logic that the more complex a problem's logical definition (for example, in terms of quantifier alternation) the more difficult its solvability. Descriptive complexity aims to measure the computational complexity of a problem in terms of the complexity of the logical language needed to define it. As is often the case in complexity theory, the issues here become more subtle and the measure of the logical complexity of a problem more intricate than in computability theory. Descriptive complexity has its beginnings in the research of Jones, Selman, Fagin [JS74, Fag73, Fag74] and others in the early 1970's. More recently descriptive complexity has had significant applications to database theory and to computer-aided verification.

The ground breaking theorem of this area is due to Fagin [Fag73]. It provided the first major impetus for the study of descriptive complexity. Fagin's Theorem gives a logical characterization of the class NP. It states that NP is exactly the class of problems definable by existential second order Boolean formulas. This result, and others that follow, show that natural complexity classes have an intrinsic logical complexity.

To get a feel for this important idea, consider the NP-complete problem of 3 colorability of a graph. Fagin's theorem says there is a second order existential formula which holds for exactly those graphs which are 3-colorable. This formula can be written as $(\exists A, B, C)(\forall v)[(A(v) \lor B(v) \lor C(v)) \land (\forall w)(E(v, w) \rightarrow \neg(A(v) \land A(w)) \land \neg(B(v) \land B(w)) \land \neg(C(v) \land C(w)))].$

Intuitively this formula states that every vertex is colored by one of three colors A, B, or C and no two adjacent vertices have the same color. A graph, considered as a finite model, satisfies this formula if and only if it is 3-colorable.

Fagin's theorem was the first in a long line of results which prove that
complexity classes can be given logical characterizations, often very simply and elegantly. Notable among these is the theorem of Immerman and Vardi [Imm82, Var82] which captures the complexity of polynomial time. Their theorem states that the class of problems definable in first order logic with the addition of the least fixed point operator is exactly the complexity class P. Logspace can be characterized along these same lines, but using the transitive closure (TC) operator rather than least fixed point. That is, nondeterministic logspace is the class of problems definable in first order logic with the addition of TC (see Immerman [Imm88]). And if one replaces first order logic with TC with second order logic with TC the result is \textbf{PSPACE} (see Immerman [Imm83]). Other, analogous results in this field go on to characterize various circuit and parallel complexity classes, the polynomial time hierarchy, and other space classes, and even yield results concerning counting classes.

The intuition provided by looking at complexity theory in this way has proved insightful and powerful. In fact, one proof of the famous Immerman-Szelepcsenyi Theorem [Imm88, Sze88] (that by Immerman) came from these logical considerations. This theorem says that any nondeterministic space class which contains logspace is closed under complement. An immediate consequence is that the context sensitive languages are closed under complement, answering a question which had been open for about 25 years.

To this point we have considered several of the most fully developed and fundamental areas of complexity theory. We now survey a few of the more central topics in the field dealing with other models of computation and their complexity theory. These include circuit complexity, communication complexity and proof complexity.

8 Finite Models

8.1 Circuit Complexity

The properties and construction of efficient Boolean circuits are of practical importance as they are the building block of computers. Circuit complexity studies bounds on the size and depth of circuits which compute a given Boolean functions. Aside from their practical value, such bounds are closely tied to important questions about Turing machine computations.

Boolean circuits are directed acyclic graphs whose internal nodes (or "gates") are Boolean functions, most often the "standard" Boolean functions, \textbf{and}, \textbf{or} and \textbf{not}. In a circuit, the nodes of in-degree 0 are called input nodes and labeled with input variables. The nodes with out-degree 0 are
called output nodes. The value of the circuit is computed in the natural way by giving values to the input variables, applying the gates to these values, and computing the output values.

The size, $s(C)$, of a circuit $C$ is the number of gates it contains. The depth, $d(C)$, of a circuit $C$ is the length of the longest path from an input to an output node.

A circuit with $n$ inputs can be thought of as a recognizer of a set of strings of length $n$, namely those which result in the circuit evaluating to 1. In order to consider circuits as recognizing an infinite set of strings, we consider circuit families which are infinite collections of circuits, $C_n$, one for each input length. In this way a circuit family can recognize a language just as a Turing machine can.

A circuit family is a nonuniform model, the function taking $n$ to $C_n$ may not be computable. A nonuniform circuit family can recognize noncomputable sets. We can measure the size and depth of circuit families using asymptotic notation. So, for example, we say that a circuit family has polynomial size if $s(C_n)$ is $O(p(n))$, for some polynomial $p(n)$. Any language in P has polynomial size circuits. That is, it is recognized by a circuit family which has polynomial size. And so proving that some NP problem does not have polynomial size circuits would imply that $P \neq NP$. Largely because of many such implications for complexity classes, considerable effort has been devoted to proving circuit lower bounds. However, to this point this effort has met with limited success.

In an early paper, Shannon [Sha49] showed that most Boolean functions require exponential size circuits. This proof was nonconstructive and proving bounds on particular functions is more difficult. In fact, no non-linear lower bound is known for the circuit size of a concrete function.

To get more positive results one needs to restrict the circuit families being considered. This can be done by requiring some uniformity in the function mapping $n$ to $C_n$, or it can be done by restricting the size or depth of the circuits themselves. For example, the class $AC^0$ consists of those languages recognized by uniform, constant depth, polynomial size circuits with and, or and not gates which allow unbounded fan-in. One early and fundamental results, due to Furst, Saxe and Sipser [FFS88] and Ajtai [Ajt83] is that the parity function is not in $AC^0$, and in fact requires exponential size $AC^0$-type circuits [Yao90]. This immediately implies that $AC^0$ differs from the class $ACC$ of languages which have circuit families made from $AC^0$ circuits with the addition of $Mod_m$ gates, with $m$ fixed for the circuit family. It also can be shown to imply the existence of an oracle separating the polynomial hierarchy from $PSPACE$. 
It is also known that the classes $\text{ACC}(p)$ are all distinct, where only $\text{Mod}_p$ gates are allowed, for $p$ a prime. This was shown by Smolensky [Smo87] and Razborov [Raz98]. $\text{ACC}$ itself has resisted all lower bound techniques and in fact it is not even know to be properly contained in $\text{NP}$.

Razborov [Raz85b] showed that clique does not have small monotone circuits, i.e., just AND and OR gates without negations. However, this result says more about the limitations of monotone circuits as Razborov [Raz85a] showed that the matching problem, known to be in $\text{P}$, also does not have small monotone circuit.

8.2 Communication Complexity

Much of modern computer science deals with the speed and efficiency at which digital communication can take place. Communication complexity is an attempt to model the efficiency and intrinsic complexity of communication between computers. It studies problems which model typical communication needs of computations and attempts to determine the bounds on the amount of communication between processors that these problems require.

The basic question of communication complexity is, how much information do two parties need to exchange in order to carry out a computation? We assume both parties have unlimited computational power.

For example, consider the case where both parties have $n$ input bits and they want to determine if there is a position $i < n$ where the two bits in position $i$ match. It is not hard to see that the communication complexity of this problem is $n$, as the $n$ bits are independent and in the worst case, all $n$ bits of one party have to be transmitted to the other.

Now consider the problem of computing the parity of a string of bits where $1/2$ of the bits are given to party 1 and the other half to party 2. In this case, party 1 need only compute the parity of her bits and send this parity to party 2 who can then compute the parity of the whole bit string. So in this case the communication complexity is a single bit.

Communication complexity has provided upper and lower bounds for the complexity of many fundamental communication problems. It has clarified the role which communication plays in distributed and parallel computation as well as in the performance of VLSI circuits. It also applies and has had an impact on the study of interactive protocols. For a good survey of the major results in this field, consult Nisan and Kushelevitz [KN96].
8.3 Proof Complexity

The class NP can be characterized as those problems which have short, easily verified membership proofs. Dual to NP-complete problems, like SAT, are co-NP-complete problems, such as TAUT (the collection of propositional tautologies). TAUT is not known to have short, easily verified membership proofs, and in fact if it did then NP = co-NP (see Cook and Reckhow [CR73]). Proof complexity studies the lengths of proofs in propositional logic and the connections between propositional proofs and computational complexity theory, circuit complexity and automated theorem proving. In the last decade there have been significant advances in lower bounds for propositional proof complexity as well as in the study of new and interesting proof systems.

Cook and Reckhow [CR73] were the first to make the notion of a propositional proof system precise. They realized that to do this they needed to specify exactly what a proof is and to give a general format for presenting and efficiently verifying a proof p. They defined a propositional proof system S to be a polynomial-time computable predicate, R, such that for all propositional formulas, F, F ∈ TAUT ↔ ∃p S(F, p). The complexity of S is then defined to be the smallest function f : N → N which bounds the lengths of the proofs of S as a function of the lengths of the tautologies being proved. Efficient proof systems, those with complexity bounded by some polynomial, are called polynomial-bounded proof systems.

Several natural proof systems have been defined and their complexity and relationship explored. Among the most studied are Frege and extended-Frege Proof systems [Urq87] and [KP89], refutation systems, most notably resolution [Rob65] and circuit based proof systems [Ajt83] and [Bus87]. We briefly discuss the complexity of resolution systems here, but see Beame and Pitassi [BP98] for a nice overview of results concerning these other proof systems.

Resolution proof systems are the most well-studied model. Resolution is a very restricted proof system and so has provided the setting for the first lower bound proofs. Resolution proof systems are refutation systems where a statement D is proved by assuming its negation and deriving a contradiction from this negation. In a resolution proof system there is a single rule of inference, resolution, which is a form of cut. In its propositional form it say that is F ∨ x and G ∨ ¬x are true then F ∨ G follows. A restricted form of resolution, called regular resolution, was proved to have a superpolynomial lower bound by Tseitin [Tse68] on certain tautologies representing graph properties. The first superpolynomial lower bound for general resolution was
achieved by Haken [Häs89] who in 1985 proved an exponential lower bound for the pigeonhole principle. Since then several other classes of tautologies have been shown to require superpolynomial long resolution proofs.

9 Quantum Computing

The mark of a good scientific field is its ability to adapt to new ideas and new technologies. Computational complexity reaches this ideal. As we have developed new ideas of probabilistic and parallel computation, the complexity community has not thrown out the previous research, rather they have modified the existing models to fit these new ideas and have shown how to connect the power of probabilistic and parallel computation to our already rich theory. Most recently complexity theorists have begun to analyze the computational power of machines based on quantum mechanics.

In 1982, Richard Feynman [Fey82], the physicist, noted that current computer technology could not efficiently simulate quantum systems. He suggested the possibility that computers built on quantum mechanics might be able to perform this task. David Deutsch [Deu85] in 1985 developed a theoretical computation model based on quantum mechanics and suggested that such a model could efficiently compute problems not computable by a traditional computer.

Two quantum algorithms have received quite a bit of notice: Shor's [Sho97] procedure for factoring integers in polynomial time on a quantum computer and Grover's [Gro96] technique for searching a database of n elements in $O(\sqrt{n})$ time.

We know surprisingly little about the computational complexity of quantum computing. Bernstein and Vazirani [BV97] give a formal definition of the class BQP of language efficiently computable by quantum computers. They show the surprising robustness of BQP which remains unscathed under variations of the model such as restricting to a small set of rational amplitudes, allowing quantum subroutines and a single measurement at the end of the computation.

Bernstein and Vazirani show that BQP is contained in PSPACE. Adleman, DeMarrais and Huang [ADH97] show that BQP is contained in the counting class PP. Bennett, Bernstein, Brassard and Vazirani [BBBV97] give a relativized world where NP is not contained in BQP. We do not know any nonrelativized consequences of NP in BQP or if BQP lies in the polynomial-time hierarchy.

What about quantum variations of NP and interactive proof systems?
Fenner, Green, Homer and Pruim [FGHP99] consider the class consisting of the languages \( L \) such that for some polynomial-time quantum Turing machine, \( x \) is in \( L \) when \( M(x) \) accepts with positive probability. They show the equivalence of this class to the counting class \( \text{co-} \mathsf{C}_\mathsf{P} \).

Watrous [Wat99] shows that every language in \( \mathsf{PSPACE} \) has a bounded-round quantum interactive proof system. Kitaev and Watrous [KW00] show that every quantum interactive proof system has an equivalent bounded-round proof system and every such language sits in deterministic exponential time.

We have seen quite a bit of progress on quantum decision tree complexity. In this model we count the number of queries made to a black-box database of size \( n \). Quantum queries can be made in superposition.

Deutsch and Jozsa [DJ92] gave an early example of a simple function that can be solved with one query quantumly but requires \( \Omega(n) \) queries deterministically or probabilistically with no error. Bernstein and Vazirani [BV97] give the first example of a problem that can be solved with polynomial number of queries quantumly but requires a superpolynomial number of queries probabilistically with bounded error. Simon [Sim97] gives another example with an exponential gap. Brassard and Hoyer [BH97] gave a zero-error quantum algorithms for Simon's problem. Shor's factoring algorithm [Sho97] can be viewed as an extension of Simon's problem that finds the period in a periodic black-box function.

All of these examples require a promise, i.e., restricting the allowable inputs to be tested. Fortnow and Rogers [FR99] and Beals, Buhrman, Cleve, Mosca and de Wolf [BBC+98] show that a promise is necessary to get a superpolynomial separation.

10 Future Directions

Despite the plethora of exciting results in computational complexity over the past forty years, true complexity class separations have remained beyond our grasp. Tackling these problems, especially showing a separation of \( \mathsf{P} \) and \( \mathsf{NP} \), is our greatest challenge for the future.

How will someone prove that \( \mathsf{P} \) and \( \mathsf{NP} \) differ? As of this writing, we have no serious techniques that could help separate these classes. What kind of future ideas could lead us to answer this difficult question? Some possibilities:

- A unexpected connection to other areas of mathematics such as algebraic geometry or higher cohomology. Perhaps even an area of math-
ematics not yet developed. Perhaps someone will develop a whole new direction for mathematics in order to handle the \( P \) versus \( NP \) question.

- New techniques to prove lower bounds for circuits, branching programs and/or proof systems in models strong enough to give complexity class separations.

- A new characterization of \( P \) or \( NP \) that makes separation more tractable.

- A clever twist on old-fashioned diagonalization, still the only techniques that has given any lower bounds on complexity classes.

Complexity theory will progress in areas beyond class separation. Still, quite a few interesting questions remain in many areas, even basic questions in quantum computational complexity remain. Complexity theorists will continue to forge new ground and find new and exciting results in these directions.

As with probabilistic, parallel and quantum complexity, new models of computation will be developed. Computational complexity theorists will be right on top of these developments leading the way to understand the inherent efficient computational power of these models.

We have seen many books and popular news stories about the other “complexity”, complex systems that occur in many aspects of society and nature such as financial markets, the internet, biological systems, the weather and dehatably even physical systems. This theory suggests that such systems have a very simple set of rules that when combined produce quite a complex behavior. Computer programs exhibit a very similar behavior. We will see computational complexity techniques used to help understand the efficiency of the complex behavior of these systems.

Finally, computational complexity will continue to have the Big Surprise. No one can predict the next big surprise but it will happen as it always does.

Let us end this survey with a quote from Juris Hartmanis’ notebook (see [Har81]) in his entry dated December 31, 1962

This was a good year.

This was a good forty years and complexity theory is only getting started.
11 Further Reading

There have been several articles on various aspects of the history of complexity theory, many of which we have used as source material for this article. We give a small sampling of pointers here:

- [Har81] Juris Hartmanis reminisces on the beginnings of complexity theory.
- [Tra84] Boris Trakhtenbrot describes the development of NP-completeness from the Russian perspective.
- [Sip92] Michael Sipser gives a historical account of the P versus NP question including a copy and translation of Gödel's historic letter to von Neumann.
- [GJ79] Michael Garey and David Johnson give a “terminological history” of NP-completeness and a very readable account of the basic theory of NP-completeness.
- The collection of papers edited by Hochbaum [Hoc95] is a good overview of progress made in approximating solutions to NP-hard problems.
- Consult the book by Greenlaw, Hoover and Ruzzo [RGR95] to learn more of complexity theory within P and for many more P-complete problems.
- The Turing award lectures of Cook [Coo83], Karp [Kar86], Hartmanis [Har94] and Stearns [Ste94] give interesting insights into the early days of computational complexity.
- The textbook of Homer and Selman [HS00] contains a careful development of the definitions and basic concepts of complexity theory, and proofs of many central facts in this field.
- The complexity columns of SIGACT news and the Bulletin of the EATCS have had a number of excellent surveys on many of the areas described in this article.
- The two collections Complexity Theory Retrospective [Sel88] and Complexity Theory Retrospective II [HS97] contain some excellent recent surveys of several of the topics mentioned here.
Acknowledgments

The authors would like to thank their colleagues, far too numerous to mention, whom we have had many wonderful discussions about complexity over the past few decades. Many of these discussions have affected how we have produced various aspects of this article.

We would like to thank the editors of this book, particularly Aki Kanamori, for inviting us to write this article. The first author would like to thank the program committee of the 17th Annual Conference on Computational Complexity for inviting him to present some of this material at that conference.

References


The Natural Computing Column

by

Grzegorz ROZENBERG
Leiden University, Leiden Center for Natural Computing
Niels Bohrweg 1, 2333 CA Leiden, The Netherlands
rozenberg@ilacs.nl

Quantum Computation explained to my Mother

Pablo Arrighi,1,*

1Computer Laboratory, University of Cambridge,
15 JJ Thomson Avenue, Cambridge CB3 0FD, U.K.

There are many falsely intuitive introductions to quantum theory and quantum computation in a handwave. There are also numerous documents which teach those subjects in a mathematically sound manner. To my knowledge this paper is the shortest of the latter category. The aim is to deliver a short yet rigorous and self-contained introduction to Quantum Computation, whilst shining the reader with no prior knowledge of anything but the fundamental operations on real numbers. Successively I introduce complex matrices; the postulates of quantum theory and the simplest quantum algorithm. The document originates from a fifty minutes talk addressed to a non-specialist audience, in which I sought to take the shortest mathematical path that proves a quantum algorithm right.

Keywords: introduction, tutorial

1. SOME MATHEMATICS

I will begin this introduction with less than three pages of mathematics, mainly definitions. These notions constitute the vocabulary, the very language of quantum theory, and every single one of them will find its use in the second part, when I introduce the postulates of quantum theory.

A. Complex Numbers

A real number is a number just like you are used to. E.g. 1, 0, -4.3 are all real numbers. A complex number, on the other hand, is just a pair of real numbers. I.e. suppose z is a complex number (z is just a name we give to the number, we could call it zorro), then z must be of the form (a, b) where a and b are real numbers.

Now I must teach you how to add or multiply complex numbers. Suppose we have two complex numbers \( z_1 = (a_1, b_1) \) and \( z_2 = (a_2, b_2) \). Addition first: \( z_1 + z_2 \) is defined to be the pair of real numbers \( (a_1 + a_2, b_1 + b_2) \). And now multiplication (when I put two number next to one another, with no sign in between that means they are multiplied): \( z_1 z_2 \) is defined to be the pair of real numbers given by \( (a_1 a_2 - b_1 b_2, a_1 b_2 + a_2 b_1) \).

Sometimes we want to change the sign of the second (real) component of the complex number \( z \). This operation is called conjugation, and is denoted by a upper index "\(^*\)" i.e. \( z^* \) is defined to be the pair of real numbers \( (a, -b) \).

Another useful operation we do on a complex number is to take its norm. The norm of \( z = (a, b) \) is defined to be the real number \( \sqrt{a^2 + b^2} \). This operation is denoted by two vertical bars surrounding the complex number, in other words \( |z| \) is simply a notation for \( \sqrt{a^2 + b^2} \).

B. Matrices

A matrix of things is a table containing those things, for instance: \( \begin{pmatrix} \heartsuit & \spadesuit \\ \diamondsuit & \clubsuit \end{pmatrix} \) is a matrix of card suits. We shall call this matrix \( M \) for use in later examples.

A matrix does not have to be square. We say that a matrix is \( m \times n \) if it has \( m \) horizontal lines and \( n \) vertical lines. For instance a column is a \( 1 \times n \) matrix e.g. \( \begin{pmatrix} \heartsuit \\ \diamondsuit \end{pmatrix} \). Similarly a row is a \( m \times 1 \) matrix, e.g. \( \begin{pmatrix} \heartsuit & \spadesuit \end{pmatrix} \) is a row.

The \( ij \)-component of a matrix designates the 'thing' which is sitting at vertical position \( i \) and horizontal position \( j \) in the table, starting from the upper left corner. For instance the 2 1-component of \( M \) is \( \heartsuit \). If \( A \) is a matrix then the

*Electronic address: pja35@cam.ac.uk
The ij-component of $A$ is denoted $A_{ij}$, e.g. here you have that $M_{11} = \diamond$, $M_{21} = \diamond$ etc. Given a matrix we often need to make its vertical lines into its horizontal lines and vice-versa. This operation is called transposition and is written $^t$. In other words if $A$ the $m \times n$ matrix with ij-component $A_{ij}$, then $A^t$ is defined to be the $n \times m$ matrix which has ij-component $A_{ji}$. Thus we have $A_{ij}^t = A_{ji}$. Here are two examples:

$$M^t = \begin{pmatrix}
\diamond & \diamond \\
\diamond & \diamond
\end{pmatrix} ; \quad \left( \begin{pmatrix}
\diamond \\
\diamond
\end{pmatrix} \right)^t = \begin{pmatrix}
\diamond \\
\diamond
\end{pmatrix}$$

C. Matrices of Numbers

Let us now consider matrices of numbers. The good thing about numbers (real or complex, it does not matter at this point) is that you know how to add and multiply them. This particularity will now enable us to define addition and multiplication of matrices of these numbers.

In order to add two matrices $A$ and $B$ they must both be $m \times n$ matrices (they have the same size). Suppose $A$ has $ij$-components. Then $A + B$ is defined to be the $m \times n$ matrix with $ij$-components $A_{ij} + B_{ij}$.

If we now want to multiply the matrix $A$ by the matrix $B$ it has to be the case that the number of vertical lines of $A$ equals that of the number of horizontal lines of $B$. Now suppose $A$ is an $m \times n$ matrix with $ij$-components $A_{ij}$, whilst $B$ is an $n \times r$ matrix with $pq$-components $B_{pq}$. Then $AB$ is defined to be the $m \times r$ matrix with $iq$-components $A_{ij}B_{jk} + A_{ij}B_{jk} + \ldots + A_{ij}B_{jk}$. To make things clear let us work this out explicitly for general $2 \times 2$ matrices of numbers:

Let $A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$ and $B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$

Then $A + B = \begin{pmatrix} A_{11} + B_{11} & A_{12} + B_{12} \\ A_{21} + B_{21} & A_{22} + B_{22} \end{pmatrix}$ and $AB = \begin{pmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{pmatrix}$

D. Matrices of Complex Numbers

Matrix addition and multiplication work on numbers, whether they are real or complex. But from now we look at matrices of complex numbers only, upon which we define one last operation called dagger. To do a dagger operation upon a matrix is to transpose the matrix and then to conjugate all the complex numbers it contains. This operation is denoted $^\dagger$. We thus have $A_{ij}^\dagger = A_{ji}^*$. In other words if $A$ is the $m \times n$ matrix with $ij$-component $A_{ij}$, then $A^\dagger$ is defined to be the $n \times m$ matrix which has $ij$-component $A_{ji}^*$.

Quite a remarkable $n \times n$ matrix of complex numbers is the one we call 'the identity matrix'. It is defined such that its $ij$-component is the complex number $(0, 0)$ when $i \neq j$, and the complex number $(1, 0)$ when $i = j$. The $n \times n$ identity matrix is denoted $I_n$, as in:

$$I_1 = \begin{pmatrix} (1, 0) \end{pmatrix} \quad \text{and} \quad I_2 = \begin{pmatrix}
(1, 0) & (0, 0) \\
(0, 0) & (1, 0)
\end{pmatrix}$$

Having defined the identity matrices we are now able to explain what it means to be a unit matrix of complex numbers. Consider $M$ an $m \times n$ matrix of complex numbers. $M$ is said to be a unit matrix if (and only if) it is true that $M^\dagger M = I_n$.

E. Some properties

You may skip the following three properties if you wish, but they will be needed in order to fully understand the comments which follow postulates 2 and 3. Moreover by going through the proofs you will exercise your understanding of the many definitions you have just swallowed.

Property 1 Let $A$ be an $n \times m$ matrix of complex numbers and $I_m$ the $m \times m$ identity matrix. We then have that $AI_m = A$. In other words multiplying a matrix by the identity matrix leaves the matrix unchanged.

Proof. First note that a complex number $(a, b)$ multiplied by the complex number $(1, 0)$ is, by definition of complex number multiplication, given by $(1a - 0b, 0a + 1b)$, which is just $(a, b)$ again. Likewise note that a complex number
\((a, b)\) multiplied by the complex number \((0, 0)\) is given by \((0a - 0b, 0a + 0b)\), which is just \((0, 0)\). Now by definition of matrix multiplication the \(i\)-component of \(A_{1n}\) is given by: (we denote \(I_m\) by just \(I\) until the end of the proof)

\[
(AI)_{1q} = A_{11}I_{1q} + A_{12}I_{2q} + \ldots + A_{1n}I_{nq}
= A_{11}(0, 0) + A_{12}(0, 0) + \ldots + A_{1q}(1, 0) + \ldots + A_{1n}(0, 0)
\]

The second line was obtained by replacing the \(I_{pq}\) with their value, which we know from the definition of the identity matrix. Now using the two remarks at the beginning of the proof we can further simplify this equation:

\[
(AI)_{1q} = (0, 0) + (0, 0) + \ldots + A_{1q} + \ldots + (0, 0)
= A_{1q}
\]

by complex number addition.

Thus the components of \(AI\) are precisely those of \(A\). □

Property 2 Let \(A\) be an \(m \times n\) matrix of complex numbers and \(B\) be an \(n \times r\) matrix of complex numbers. Then the following equality is true:

\[
(AB)^\dagger = B^\dagger A^\dagger
\]

Proof. First note that

\[
((a_1, b_1) + (a_2, b_2))^\ast = (a_1, b_1)^\ast + (a_2, b_2)^\ast
\]

(1)

This is obvious since

\[
((a_1, b_1) + (a_2, b_2))^\ast = (a_1 + a_2, b_1 + b_2)^\ast
= (a_1 + a_2, -b_1 - b_2)
\]

and

\[
(a_1, b_1)^\ast + (a_2, b_2)^\ast = (a_1, -b_1) + (a_2, -b_2)
= (a_1 + a_2, -b_1 - b_2)
\]

Likewise note that

\[
((a_1, b_1)(a_2, b_2))^\ast = (a_1, b_1)^\ast(a_2, b_2)^\ast
\]

(2)

and also

\[
(a_1, b_1)(a_2, b_2) = (a_2, b_2)(a_1, b_1)
\]

(3)

again this is easily verified by computing the left-hand-side and the right-hand-side of those equalities. You may want to check this as an exercise.

Now by definition of matrix multiplication we have that

\[
(AB)_{1q} = A_{11}B_{1q} + A_{12}B_{2q} + \ldots + A_{1n}B_{nq}
\]

Thus the components of \((AB)^\dagger\) are given by

\[
(AB)^\dagger_{1q} = (AB)_{1q}^\ast
= A_{11}^\ast B_{1q}^\ast + A_{12}^\ast B_{2q}^\ast + \ldots + A_{1n}^\ast B_{nq}^\ast
= B_{11}^\ast A_{11}^\ast + B_{12}^\ast A_{12}^\ast + \ldots + B_{1n}^\ast A_{1n}^\ast
\]

where we used equations (1) and (2) to obtain the second line, and equation (3) to obtain the third line. Now consider the components of \(B^\dagger A^\dagger\). By definition of matrix multiplication we have that

\[
(B^\dagger A^\dagger)_{1q} = B_{11}^\dagger A_{1q}^\dagger + B_{12}^\dagger A_{2q}^\dagger + \ldots + B_{1n}^\dagger A_{nq}^\dagger
= B_{11}^\dagger A_{11}^\dagger + B_{12}^\dagger A_{12}^\dagger + \ldots + B_{1n}^\dagger A_{1n}^\dagger
\]

where the last line was obtained using the fact that \(A_{1q}^\dagger = A_{1q}^\ast\). Thus the components of \((AB)^\dagger\) are precisely those of \(B^\dagger A^\dagger\). □
Property 3 Let \( V \) be a \( n \times 1 \) unit matrix of complex numbers (a column). Then it is the case that:

\[
|V_{11}|^2 + |V_{21}|^2 + \ldots + |V_{n1}|^2 = 1
\]

\[\text{Proof.} \] First let \( z = (a, b) \) be a complex number, and note that

\[
z^*z = (a^2 + b^2, 0)
\]

\[
= (|z|^2, 0)
\]

Now consider the 11-component of \( V^\dagger V \). We have:

\[
(V^\dagger V)_{11} = V_{11}^*V_{11} + V_{12}^*V_{21} + \ldots + V_{n1}^*V_{n1}
\]

where we used successively: the definition of matrix multiplication, and \( A^\dagger A \). The last line can be further simplified using our first remark, namely:

\[
V_{11}^*V_{11} = (|V_{11}|^2, 0)
\]

Thus

\[
(V^\dagger V)_{11} = (|V_{11}|^2, 0) + (|V_{21}|^2, 0) + \ldots + (|V_{n1}|^2, 0)
\]

\[
= (|V_{11}|^2 + |V_{21}|^2 + \ldots + |V_{n1}|^2, 0)
\]

Because \( V \) is unit the last line must be equal to \((1, 0)\), and so we have proved the property. \( \square \)

II. QUANTUM THEORY

Quantum theory is one of the pillars of modern physics. The theory is 100 years old and thoroughly checked by experiments; it enables physicists to understand and predict the behaviors of any closed (perfectly isolated from the rest of the world) physical system. Usually these are small systems such as atoms, electrons, photons etc. (only because they are generally less subject to outside interactions).

A. States

Postulate 1 The state of a closed physical system is wholly described by a unit \( n \times 1 \) matrix of complex numbers.

Comments. In other words a state is given by a column of \( n \) complex numbers

\[
V = \begin{pmatrix} V_{11} \\ \vdots \\ V_{n1} \end{pmatrix} \quad \text{such that} \quad V^\dagger V = I_1.
\]

What we mean by closed physical system is just about anything which is totally isolated from the rest of the world. The number of components \( n \) varies depending on how complicated the system is; it is called the degrees of freedom or the dimension of the system. The postulate itself is extremely short and simple. It is nonetheless puzzling as soon as you attempt to apprehend it with your classical intuition.

Example. Consider a coin, which insofar as we have always observed, can either by 'head @' or 'tail @'. Thus we will suppose it has \( n = 2 \) degrees of freedom, and we will further assume that the state:

'head @' corresponds to quantum state \( \begin{pmatrix} 1, 0 \\ 0, 0 \end{pmatrix} \)

whilst 'tail @' corresponds to quantum state \( \begin{pmatrix} 0, 0 \\ 1, 0 \end{pmatrix} \)
Now if the coin was to be shut in a perfectly closed box, it would start behaving like a quantum coin. Thus the state:

\[
\begin{pmatrix}
\frac{1}{\sqrt{2}} \\
\frac{1}{\sqrt{2}}
\end{pmatrix}
\]

would become perfectly allowable. A quantum coin can be in a superposition of head and tail, i.e. it can be both head and tail at the same time, in some proportion. Quantum theory is more general than our classical intuition: it allows for more possible states. It is as if 'head' and 'tail' were two axes, and the quantum coin was allowed to live in the plane described by those axes.

B. Evolution

Postulate 2 A closed physical system in state **V** will evolve, after a certain period of time, into a new state **W** according to

\[ W = UV \]

where **U** is a \( n \times n \) unit matrix of complex numbers.

Comments. In other words, in order to see how the quantum state of a closed physical system evolves, you have to multiply it by the matrix which describes its evolution (which we call **U**). **U** could be any matrix of complex numbers so long as it is \( n \times n \) (remember **V** is an \( n \times 1 \) matrix) and verifies the condition \( U^*U = I_n \). Note that this postulate is coherent with the first one, because evolution under **U** takes an allowed quantum state into an allowed quantum state. Indeed suppose **V** is a valid state, i.e. an \( n \times 1 \) matrix verifying \( V^*V = I_1 \). By definition of the matrix multiplication an \( n \times 1 \) matrix multiplied by an \( n \times n \) matrix is also an \( n \times 1 \) matrix, and thus **W** has the right sizes. Is it a unit matrix? Yes:

\[
W^*W = (UV)^*(UV) \quad \text{by definition of } W
\]
\[
= V^*U^*UV \quad \text{by Property 2}
\]
\[
= V^*I_n V \quad \text{since } U \text{ is unit}
\]
\[
= V^*V \quad \text{by Property 1}
\]
\[
= I_1 \quad \text{since } V \text{ is unit}
\]

Thus **W** is a valid quantum state.

C. Measurement

Postulate 3 When a physical system in state

\[
V = \begin{pmatrix}
V_{11} \\
\vdots \\
V_{n1}
\end{pmatrix}
\]

is measured, it yields outcome \( i \) with probability \( p_i = |V_{i1}|^2 \). Whenever outcome \( i \) occurs, the system is left in the state:

\[
W = \begin{pmatrix}
(0,0) \\
\vdots \\
(1,0) \\
\vdots \\
(0,0)
\end{pmatrix} \quad \text{\( i \)th position}
Example. Suppose you have a quantum coin in state:

\[
\begin{pmatrix}
\frac{1}{\sqrt{2}} \\
\frac{1}{\sqrt{2}}
\end{pmatrix}
\]

which you decide to measure. With a probability \( p_1 = \frac{1}{\sqrt{2}}^2 = \frac{1}{2} \) you will know that outcome '1' has occurred, in which case your quantum system will be left in state

\[
\begin{pmatrix}
1 \\
0
\end{pmatrix}
\]

But with probability \( p_2 = \frac{1}{2} \) outcome '2' may occur instead, in which case your quantum system will be left in state \( \psi^' \).

Comments. Thus a measurement in quantum theory is fundamentally a probabilistic process. For this postulate to work well we need to be sure that the probabilities all sum up to 1 (so that something happens 100% of the time). But you can check that this is the case:

\[
p_1 + \ldots + p_n = |V_{11}|^2 + \ldots + |V_{n1}|^2 \quad \text{by postulate 3}
\]

The other striking feature of this postulate is that the state of the system gets changed under the measurement. In our example everything happens as though the quantum coin in state \( \psi^' + \psi \) is asked to make up its mind between \( \psi \) and \( \psi^' \). The quantum coin decides at random, but once it does it remains coherent with its decision: its new state is either \( \psi \) or \( \psi^' \).

This feature provides the basis for one of the latest high-tech applications of quantum theory: quantum cryptography. Suppose Alice and Bob want to communicate secretly over the phone, but Eve, the Eavesdropper, might be spying upon their conversation. What Alice and Bob can do is to send quantum coins to each other across the (upgraded) phone network. As Eve attempts to measure what the honest parties are saying, she is bound to change the state of the coin. This will enable Alice and Bob to detect her malevolent presence.

III. DEUTSCH-JOZSA ALGORITHM

The measurement postulate will (probably) make you think that quantum theory is just a convoluted machinery whose only purpose is to describe objects which might be in 'state 1' with probability \( p_1 \), in 'state 2' with probability \( p_2 \) etc. until \( n \). After all why bother thinking of the state \( \psi \psi \) as a coin which is both head \( \psi \) and tail \( \psi \) at the same time - when after it gets observed it collapses to either head \( \psi \) or tail \( \psi \) anyway?

No. You have to consider that the coin is both \( \psi \) and \( \psi \) until you measure it, because this is how it behaves experimentally (until you measure it). In other words the only way to account for what happens between the moment you prepare your initial system and the moment you measure it is to think of the complex components of the state \( V \) as amplitudes, proportions and not as probabilities. This has much to do with what Postulate 2 enables us to do.

In this last part we shall illustrate this point by considering the simplest of all known quantum algorithms[2]. An algorithm is just a recipe that is used to systematically solve a mathematical problem. But the mathematical problem we will now introduce cannot be solved by classical means: it can only be solved using quantum theory, that is with a quantum algorithm. The fact that this algorithm does work in practice ought to demonstrate that the amplitudes of quantum theory permit us to do things which mere probabilities would not allow, and would not explain.

A. The problem

A boolean value is something which can either be True or False. For instance the statement 'the sky is blue' has the boolean value True almost anywhere in the world with the exception of England, where it takes the value False. A boolean operator is just a 'box' which takes one or several boolean values and returns one or several boolean values. In order to define our problem we need to become familiar with two boolean operators, which we now describe.

The boolean operator Not takes the boolean value True into False and the boolean value False into True. We denote this as follows:

\[
\text{Not(True)} = \text{False} \\
\text{Not(False)} = \text{True}
\]
The boolean operator \( \text{Xor} \) (exclusive or) takes two boolean values and returns one boolean value. It returns True either if the first boolean value it takes is True and the second one is False or if the second boolean value it takes is True and the first one is False. Otherwise it returns False. We denote this as follows:

\[
\begin{align*}
\text{Xor}(\text{True}, \text{False}) &= \text{True} \\
\text{Xor}(\text{False}, \text{True}) &= \text{True} \\
\text{Xor}(\text{False}, \text{False}) &= \text{False} \\
\text{Xor}(\text{True}, \text{True}) &= \text{False}
\end{align*}
\]

In other words \( \text{Xor} \) compares its two input boolean values: it returns True if they are different and False if they are the same. We are now ready to state the problem.

**Problem 1** Suppose we are given a mysterious boolean operator \( F \) (a black box) which takes one boolean value and returns another boolean value. We want to calculate \( \text{Xor}(F(\text{False}), F(\text{True})) \), i.e. the boolean value returned by \( \text{Xor} \) when applied to the two possible results of \( F \). But we are allowed to use the mysterious boolean operator \( F \) only once.

It is clear that this problem cannot be solved classically. This is because in order to learn anything about \( F \) you will have to use \( F \). But we are allowed to do this only once. Suppose we use \( F \) on input boolean value False. This gives us \( F(\text{False}) \), but tells us nothing about \( F(\text{True}) \) which may still be either True or False. Thus we cannot compute \( \text{Xor}(F(\text{False}), F(\text{True})) \) and we fail to solve the problem. The same reasoning applies if we begin by using \( F \) to obtain \( F(\text{True}) \).

But what would happen if we had the possibility to use \( F \) upon an input boolean value which is both True and False, in some proportions (a superposition)?

**B. The quantum setup**

Now suppose that the mysterious boolean operator \( F \) is given in the form of a 'quantum black box' instead. To make this more precise we need to call

\[
\begin{align*}
\text{"False, False"} \text{ the quantum state } & \begin{pmatrix} 1, 0 \\ 0, 0 \\ 0, 0 \end{pmatrix} ; \text{ "False, True"} \text{ the quantum state } \begin{pmatrix} 0, 0 \\ 0, 0 \\ 1, 0 \end{pmatrix} ; \\
\text{"True, False"} \text{ the quantum state } & \begin{pmatrix} 0, 0 \\ 0, 0 \\ 1, 0 \end{pmatrix} ; \text{ "True, True"} \text{ the quantum state } \begin{pmatrix} 1, 0 \\ 0, 0 \\ 0, 0 \end{pmatrix}
\end{align*}
\]

We assume we have access, for one use only, to a physical device which implements \( F \) as a quantum evolution. This quantum evolution \( U \) must take

\[
\begin{align*}
\text{"True, False"} \text{ into } & \text{"True, } F(\text{True})\text{"} \\
\text{"False, False"} \text{ into } & \text{"False, } F(\text{False})\text{"}
\end{align*}
\]

Notice that if for instance \( F(\text{True}) = \text{True} \) then \( \text{"True, } F(\text{True})\text{"} \) simply denotes the quantum state \("\text{True, True}\"\). Furthermore we assume \( U \) takes

\[
\begin{align*}
\text{"True, } F(\text{True})\text{"} \text{ into } & \text{"True, } \text{Not}(\text{\text{F(True)}})\text{"} \\
\text{"False, True"} \text{ into } & \text{"False, } \text{Not}(\text{\text{F(False)}})\text{"}
\end{align*}
\]

The quantum evolution \( U \) is fully specified in this manner. In matrix form it is given as follows:

\[
\begin{pmatrix}
(1 - F_{\text{False}}, 0) & (F_{\text{False}}, 0) & (0, 0) & (0, 0) \\
(F_{\text{False}}, 0) & (1 - F_{\text{False}}, 0) & (0, 0) & (0, 0) \\
(0, 0) & (0, 0) & (1 - F_{\text{True}}, 0) & (F_{\text{True}}, 0) \\
(0, 0) & (0, 0) & (F_{\text{True}}, 0) & (1 - F_{\text{True}}, 0)
\end{pmatrix}
\]
with:
\( F_{\text{False}} \) equal to 1 if \( F(\text{False}) \) is True, and 0 otherwise.
\( F_{\text{True}} \) equal to 1 if \( F(\text{True}) \) is True, and 0 otherwise.

Whatever the values of \( F_{\text{False}} \) and \( F_{\text{True}} \), the matrix of complex number defined above is unit, i.e. \( U'U = I_4 \). Thus according to postulate 2 this mysterious quantum black box is perfectly allowable physically.

As an exercise you may want to check that the matrix \( U \) does take 'True, False' into 'True, F(True)', etc., and that it is indeed unit.

For our quantum algorithm we will need another quantum evolution:

\[
H = \begin{pmatrix}
\frac{1}{2}, 0 & \frac{1}{2}, 0 & \frac{1}{2}, 0 & \frac{1}{2}, 0 \\
\frac{1}{2}, 0 & -\frac{1}{2}, 0 & \frac{1}{2}, 0 & -\frac{1}{2}, 0 \\
\frac{1}{2}, 0 & \frac{1}{2}, 0 & -\frac{1}{2}, 0 & -\frac{1}{2}, 0 \\
\frac{1}{2}, 0 & -\frac{1}{2}, 0 & -\frac{1}{2}, 0 & \frac{1}{2}, 0
\end{pmatrix}
\]

This \( H \) is also a unit matrix of complex numbers.

C. The solution

Algorithm 1 In order to solve problem 1 one may use the following algorithm:

1. Start with a closed physical system in quantum state 'False, True'.
2. Evolve the system under the quantum evolution \( H \).
3. Evolve the system under the quantum evolution \( U \).
4. Evolve the system under the quantum evolution \( H \).
5. Measure the system.

If \( \text{Xor}(F(\text{False}), F(\text{True})) \) is False the quantum measurement always yields outcome '2'.
On the other hand if \( \text{Xor}(F(\text{False}), F(\text{True})) \) is True the quantum measurement always yields outcome '4'.
Thus the algorithm always manages to determine \( \text{Xor}(F(\text{False}), F(\text{True})) \), and does so with only one use of the quantum evolution \( U \).

Proof. In Step 1 we start with a closed physical system whose quantum state is \( V = \begin{pmatrix} 0, 0 \end{pmatrix} \).

After Step 2 the quantum state of the system has become \( HV \). By working out this matrix multiplication we have

\[
HV = \begin{pmatrix} 1/2, 0 \\ -1/2, 0 \end{pmatrix}
\]

You may want to check this matrix multiplication and the ones to follow, as an exercise.

After Step 3 the quantum state of the system has become \( UHV \). We can still work out the matrix multiplication but obviously the result now depends upon our mysterious boolean operator \( F \). Indeed we have

\[
UHV = \begin{pmatrix} 1/2 - F_{\text{False}}, 0 \\ -1/2 + F_{\text{False}}, 0 \end{pmatrix}
\]

Notice that \( UHV \) depends both upon \( F(\text{False}) \) and \( F(\text{True}) \), in some proportions.

After Step 4 the quantum state of the system has become \( HUHV \) and we have, by working out the multiplication:

\[
HUHV = \begin{pmatrix} 0, 0 \\ (1 - F_{\text{False}} - F_{\text{True}}, 0) \end{pmatrix}
\]

Finally in Step 5 we measure the state \( HUHV \). According to Postulate 3 this yields:
- outcome '1' with probability 0 (never).
- outcome '2' with probability \( p_2 = (1 - (F_{\text{False}} + F_{\text{True}}))^2 \).
- outcome '3' with probability 0 (never).
- outcome '4' with probability \( p_4 = (F_{\text{True}} - F_{\text{False}})^2 \).

Now if \( \text{Xor}(F(\text{False}), F(\text{True})) \) is \text{False} then \( F_{\text{False}} \) and \( F_{\text{True}} \) have to be the same. Thus \( F_{\text{False}} + F_{\text{True}} \) equals either 0 or 2, whereas \( F_{\text{True}} - F_{\text{False}} \) is necessarily worth 0. As a consequence \( p_2 \) must equal 1 whereas \( p_4 \) is worth 0.

Similarly, if \( \text{Xor}(F(\text{False}), F(\text{True})) \) is \text{True} then \( F_{\text{False}} \) and \( F_{\text{True}} \) have to be the different values. Thus \( F_{\text{False}} + F_{\text{True}} \) is necessarily worth 1, whereas \( F_{\text{True}} - F_{\text{False}} \) equals either -1 or 1. As a consequence \( p_2 \) is worth 0 whereas \( p_4 \) must equal 1. \( \Box \)

D. Comments

It is quite a remarkable fact that with only one use of the 'quantum black box' we succeed to determine a quantity which intrinsically depends 'on both possible values which the box may return'. Although this algorithm does not seem extremely useful in every day life, it teaches us an important lesson: the components of a quantum state must be viewed as proportions (amplitudes), not as probabilities. The quantum coin can be both head or tail in some proportions, simultaneously, until you measure it.

Until recently this feature of quantum theory was essentially regarded as an unfortunate oddity which made the theory difficult to grasp. But we are now learning to turn this feature to our own advantage, as a means of 'exploring several possibilities simultaneously' (so to speak).

This is recent research however, and to this day not so many quantum algorithms are known. Yet we do know that Quantum Computers can factorize large integer numbers efficiently, or even find a name within an unordered list of 100 people in only 5 tries. These are quite useful things to be able to do. The best place to learn about them is [3], if you have followed me this far you can go further.

IV. ACKNOWLEGMENTS

The author would like to thank his mother for suggesting this article, Anuj Dawar for his patient listening, EPSRC, Marconi, the Cambridge European and Isaac Newton Trusts for financial support.

The DF0L language equivalence problem

Juha Honkala
Department of Mathematics, University of Turku
FIN-20014 Turku, Finland
juha.honkala@utu.fi

Abstract. We give a new proof of the decidability of the DF0L language equivalence problem. As a byproduct we get a new solution of the D0L language equivalence problem.

1 Introduction

The decidability of the D0L equivalence problem is one of the basic results concerning free monoid morphisms and their iterations. The simplest known solution of the D0L sequence equivalence problem relies upon Hilbert's basis theorem and other simple properties of ideals in polynomial rings (see [2, 3]). The decidability of the D0L language equivalence problem follows from the decidability of the D0L sequence equivalence problem as shown by Nielsen [7].

The language equivalence problem remains decidable for D0L systems with finite axiom sets (see [4, 6]). The purpose of this paper is to give a new, largely self-contained proof of this decidability result. We assume that the reader is familiar with some basic properties of D0L length sequences (see [8, 9, 11, 12]) and word equations (see [2, 5, 9]).

For additional information concerning the DF0L equivalence problem the reader is referred to [4].
2 Definitions

If \( w \in X^* \) is a word and \( x \in X \) is a letter, the number of occurrences of the letter \( x \) in \( w \) is denoted by \( |w|_x \). The length of \( w \) is denoted by \( |w| \). If \( w \in X^* \), the set \( \text{alph}(w) \) is defined by

\[
\text{alph}(w) = \{ x \in X \mid |w|_x \geq 1 \}.
\]

We say that a word \( w \in X^* \) is full if \( \text{alph}(w) = X \). A language \( L \) is full if we have \( \text{alph}(u) = \text{alph}(v) \) for all \( u, v \in L \).

A DOL system is a triple

\[
G = (X, g, w),
\]

where \( X \) is a finite alphabet, \( g : X^* \rightarrow X^* \) is a morphism and \( w \in X^* \) is a word. A DFOL system is obtained from a DOL system by replacing the word \( w \) by a finite set \( F \). Hence, a DFOL system is a triple

\[
G = (X, g, F),
\]

where \( X \) is a finite alphabet, \( g : X^* \rightarrow X^* \) is a morphism and \( F \subseteq X^* \) is a finite set.

The sequence \( S(G) \) and the language \( L(G) \) of the DOL system \( G = (X, g, w) \) are defined by

\[
S(G) = (g^n(w))_{n \geq 0}
\]

and

\[
L(G) = \{ g^n(w) \mid n \geq 0 \}.
\]

The language \( L(G) \) of the DFOL system \( G = (X, g, F) \) is defined by

\[
L(G) = \{ g^n(w) \mid w \in F, \ n \geq 0 \}.
\]

As a tool, we use also HDOL systems. By definition, an HDOL system is a construct

\[
H = (X, Y, g, h, w),
\]

where \( (X, g, w) \) is a DOL system, \( Y \) is a finite alphabet and \( h : X^* \rightarrow Y^* \) is a morphism. The sequence \( S(H) \) and the language \( L(H) \) of \( H \) are defined by

\[
S(H) = (h g^n(w))_{n \geq 0}
\]

and

\[
L(H) = \{ h g^n(w) \mid n \geq 0 \}.
\]

If \( G = (X, g, w) \) is a DOL system the sequence

\[
(\text{alph}(g^n(w)))_{n \geq 0}
\]
is ultimately periodic (see [8]). Using this fact it is not difficult to see that for a DF0L system \( G = (X, g, F) \) and a subset \( X_1 \) of \( X \), the set

\[ \{ u \in L(G) \mid \text{alph}(u) = X_1 \} \]

is a DF0L language. Consequently, to prove the decidability of the DF0L language equivalence problem it suffices to consider full DF0L languages. These two observations are due to [10].

Suppose \( G = (X, g, w) \) is a D0L system. \( G \) is said to be \textit{polynomially bounded} if there is a polynomial \( r(n) \) such that

\[ |g^n(w)| \leq r(n) \]

for all \( n \geq 0 \). \( G \) is said to be \textit{exponential} if there is a real number \( \alpha > 1 \) such that

\[ |g^n(w)| \geq \alpha^n \]

for almost all \( n \geq 0 \). (We say that a property holds for almost all integers \( n \geq 0 \), if there is an integer \( n_0 \) such that the property holds for all \( n \geq n_0 \).) Every D0L system is either polynomially bounded or exponential (see [8]). It is easy to see that two D0L systems \( G_i = (X, g, w_i), i = 1, 2 \), have the same growth type if \( \text{alph}(w_1) = \text{alph}(w_2) \). Consequently, also DF0L systems generating full languages can be divided into polynomially bounded and exponential systems.

We need two lemmas concerning D0L growth.

**Lemma 1** Let \( G = (X, g, w) \) be a D0L system and assume that for every \( x \in X \) there is a nonzero polynomial \( p_x(n) \) such that

\[ |g^n(x)| = p_x(n) \]

for all \( n \geq 0 \). Assume that \( L(G) \) is not finite. Then there is a letter \( x \in X \) such that

\[ p_x(n) = an + b, \]

where \( a \) and \( b \) are integers and \( a \geq 1 \).

**Proof.** See Lemma 4.1 in [11]. \( \Box \)

**Lemma 2** Suppose \( G = (X, g, w) \) is an exponential D0L system. Then there exist positive real numbers \( a, b, \alpha \) with \( \alpha > 1 \), and a nonnegative integer \( t \) such that

\[ an^t \alpha^n \leq |g^n(w)| \leq bn^t \alpha^n \]

for almost all \( n \geq 0 \).

**Proof.** See [8, 12]. \( \Box \)
3 The HDOL covering problem

Let \( H_i = (X_i, Y_i, g_i, h_i, w_i), 1 \leq i \leq s + 1, \) be HDOL systems. Then we say that the sequences \( S(H_i), 2 \leq i \leq s + 1, \) cover \( S(H_1) \) if

\[
\{h_1g_i^n(w_i) | 2 \leq i \leq s + 1\}
\]

for all \( n \geq 0. \) By the **HDOL covering problem** we understand the problem of deciding whether \( S(H_i), 2 \leq i \leq s + 1, \) cover \( S(H_1) \) when the HDOL systems \( H_i, 1 \leq i \leq s + 1, \) are given.

The HDOL covering problem is the key tool in our solution of the DFOL equivalence problem. In this section we show that methods used to solve the DOL sequence equivalence problem can be generalized to solve the HDOL covering problem (see also [5, 6]). First we recall some results concerning word equations.

Let \( X \) be a fixed finite alphabet and let \( Z \) be a fixed finite set of word variables. A (word) equation is a pair \( (u, v) \in Z^* \times Z^* \), also written as \( u = v \). A solution of \( u = v \) is any morphism \( h : Z^* \rightarrow X^* \) such that \( h(u) = h(v) \). If \( E_p, 1 \leq p \leq k, \) are equations, the formula

\[
\bigvee_{p=1}^k E_p
\]

is called an alternative equation. A morphism \( h : Z^* \rightarrow X^* \) is called a solution of (1) if \( h \) is a solution of \( E_p \) for some \( p, 1 \leq p \leq k \). More generally, assume that \( k(n) \) is an integer for all \( n, 1 \leq n \leq m, \) and let \( E_{np} \) be an equation for \( 1 \leq n \leq m, 1 \leq p \leq k(n) \). Then a morphism \( h : Z^* \rightarrow X^* \) is called a solution of the **system of alternative equations**

\[
\bigwedge_{n=1}^m \bigvee_{p=1}^{k(n)} E_{np}
\]

if \( h \) is a solution of

\[
\bigvee_{p=1}^{k(n)} E_{np}
\]

whenever \( 1 \leq n \leq m \). Two systems of alternative equations are called equivalent if they have the same solutions.

Suppose now that \( Z \) has \( t \geq 1 \) letters. Let \( \tilde{Z} \) be a set of \( 4t \) indeterminates and let \( \mathbb{Q}[	ilde{Z}] \) be the polynomial ring. Hence the elements of \( \mathbb{Q}[	ilde{Z}] \) are polynomials having indeterminates in \( \tilde{Z} \) and rational coefficients. If \( A \) is an arbitrary subset of \( \mathbb{Q}[	ilde{Z}] \) define the set \( V(A) \) by

\[
V(A) = \{(c_1, \ldots, c_{4t}) \in \mathbb{Q}^{4t} | p(c_1, \ldots, c_{4t}) = 0 \text{ for all } p \in A\}.
\]

The following lemma is well known (see e.g. [2]).

**Lemma 3** For each equation \( E = (u, v) \in Z^* \times Z^* \) we can effectively compute a finite set \( P(E) \subseteq \mathbb{Q}[	ilde{Z}] \) of polynomials and for each morphism \( h : Z^* \rightarrow X^* \) we
can effectively compute a 4t-tuple $\text{TEST}(h) \in \mathbb{Q}^4$ such that $h$ is a solution of $E$ if and only if $\text{TEST}(h) \in V(P(E))$.

For the proof of the following lemma see [2, 5].

**Lemma 4** Suppose $(P_n)_{n \geq 1}$ is an ascending chain of finite subsets of $\mathbb{Q}[\hat{Z}]$. Then there exists an integer $m$ such that

$$V(P_m) = V(P_{m+1}).$$

Moreover, if the sets $P_n$, $n \geq 1$, are given effectively, such an integer $m$ can be computed effectively.

**Theorem 5** The HD0L covering problem is decidable.

**Proof.** Suppose $H_i = (X_i, Y, g_i, h_i, w_i)$, $1 \leq i \leq s+1$, are HD0L systems. Without restriction we assume that $X_i \cap X_j = \emptyset$ whenever $1 \leq i < j \leq s+1$. Denote

$$Z = \bigcup_{i=1}^{s+1} X_i$$

and consider $Z$ as an alphabet of word variables. If $n \geq 0$ and $2 \leq p \leq s+1$, let $E_{np}$ be the word equation

$$g^n_i(w_i) = g^n_p(w_p).$$

Let $h : Z^* \rightarrow Y^*$ be the common extension of the morphisms $h_i$, $1 \leq i \leq s+1$. Then $h$ is a solution of $E_{np}$ if and only if

$$h_1 g^n_i(w_i) = h_p g^n_p(w_p).$$

Consequently, for $m \geq 0$, $h$ is a solution of

$$\bigwedge_{n=0}^{m} \bigvee_{p=2}^{s+1} E_{np}$$

if and only if

$$h_1 g^n_i(w_i) \in \{ h_p g^n_p(w_p) \mid 2 \leq p \leq s+1 \}$$

for all $n$, $0 \leq n \leq m$.

Next, if $n \geq 0$ and $2 \leq p \leq s+1$, let $P(E_{np}) \subseteq \mathbb{Q}[\hat{Z}]$ be as in Lemma 3. Moreover, if $m \geq 0$, define the set $P_m \subseteq \mathbb{Q}[\hat{Z}]$ by

$$P_m = \bigcup_{n=0}^{m} \prod_{p=2}^{s+1} P(E_{np}).$$
Each $P_m$ is a finite set of polynomials which can be computed effectively. Moreover,

$$V(P_m) = \bigcap_{n=0}^{m} \bigcup_{p=2}^{s+1} V(P(E_{np}))$$

for $m \geq 0$.

Next, by Lemma 4 we can compute an integer $q$ such that

$$V(P_q) = V(P_{q+1}).$$

To complete the proof of Theorem 5 it suffices to show that

$$h_1g_1^n(w_i) \in \{h_1g_1^n(w_i) \mid 2 \leq i \leq s + 1\} \text{ for all } n \geq 0$$

(2)

if and only if

$$h_1g_1^n(w_i) \in \{h_1g_1^n(w_i) \mid 2 \leq i \leq s + 1\} \text{ for all } n \text{ such that } 0 \leq n \leq q.$$  

(3)

Clearly, (2) implies (3). To prove that (3) implies (2) assume on the contrary that (3) holds but (2) does not. Choose the smallest $m$ such that

$$h_1g_1^m(w_i) \notin \{h_1g_1^m(w_i) \mid 2 \leq i \leq s + 1\}.$$

Then necessarily $m > q$.

Let $g : Z^* \rightarrow Z^*$ be the common extension of the morphisms $g_i$, $1 \leq i \leq s + 1$, and consider the morphism $f = h g^{m-q-1}$. Because $f$ is a solution of

$$\bigcap_{n=0}^{q} \bigcup_{p=2}^{s+1} E_{np},$$

we have

$$\text{TEST}(f) \in \bigcap_{n=0}^{q} \bigcup_{p=2}^{s+1} V(P(E_{np})) = V(P_q),$$

where $\text{TEST}(f)$ is as in Lemma 3. Therefore

$$\text{TEST}(f) \in V(P_{q+1}) = \bigcap_{n=0}^{q+1} \bigcup_{p=2}^{s+1} V(P(E_{np})).$$

Hence there is an integer $p$ such that $2 \leq p \leq s + 1$ and

$$\text{TEST}(f) \in V(P_{q+1,p}).$$

In other words

$$f g_1^{s+1}(w_1) = f g_1^{s+1}(w_p)$$

or, equivalently,

$$h_1g_1^m(w_1) = h_1g_1^m(w_p),$$

which contradicts the choice of $m$. This concludes the proof that (3) implies (2). $\Box$
4 The DF0L language equivalence problem

In this section we use the HD0L covering problem to solve the DF0L language equivalence problem.

Theorem 6 The DF0L language equivalence problem is decidable.

As pointed out earlier, to prove Theorem 6 it suffices to consider DF0L systems generating full languages. Such systems can be divided into polynomially bounded and exponential systems. It is not difficult to see that two DF0L systems generating full languages and having different growth types are not language equivalent. In the following two subsections we will give a semialgorithm for the language equivalence of two polynomially bounded (resp. exponential) DF0L systems generating full languages. This proves Theorem 6 because there is a trivial semialgorithm for the nonequivalence of two DF0L languages.

4.1 Polynomially bounded DF0L systems

Assume that $G_i = (X, g_i, F_i), i = 1, 2,$ are polynomially bounded DF0L systems such that the languages $L(G_i), i = 1, 2,$ are full and infinite. Without restriction we assume that $\text{alph}(u) = X$ for all $u \in L(G_1) \cup L(G_2)$.

Because $G_1$ is polynomially bounded, for every $x, y \in X$ there exist an integer $t(x, y) \geq 1$ and a polynomial $p_{1,x,y}(n)$ such that

$$|g_1^{nt(x,y)}(x)|_y = p_{1,x,y}(n)$$

for almost all $n \geq 0$. Let $t$ be the product of the integers $t(x, y), x, y \in X$. By replacing $g_1$ by $g_1^t$ and $F_1$ by $\{g_1^t(w) \mid w \in F_1, 0 \leq j < t\}$ we may assume that for every $w \in F_1$ and $y \in X$ there exists a polynomial $p_{w,y}(n)$ such that

$$|g_1^n(w)|_y = p_{w,y}(n)$$

for almost all $n \geq 0$.

Similarly, we may assume that for every $v \in F_2$ and $y \in X$ there exists a polynomial $q_{v,y}(n)$ such that

$$|g_2^n(v)|_y = q_{v,y}(n)$$

for almost all $n \geq 0$.

Now, fix a word $w \in F_1$. We will give a semialgorithm for the inclusion

$$\{g_1^n(w) \mid n \geq 0\} \subseteq L(G_2).$$

This clearly gives a semialgorithm for the language equivalence of $G_1$ and $G_2$.

First, by Lemma 1 there is a letter $y \in X$ and integers $a, b$ with $a \geq 1$ such that

$$|g_1^n(w)|_y = an + b$$
for almost all \( n \geq 0 \). Consider then the polynomials \( q_{v,y}(n), v \in F_2 \). Because \( \alpha(v_1) = \alpha(v_2) \) whenever \( v_1, v_2 \in F_2 \), the polynomials \( q_{v,y}(n), v \in F_2 \), have the same degree. If this degree is not one, we have \( L(G_1) \neq L(G_2) \). Therefore, assume the common degree is one and denote
\[
q_{v,y}(n) = a_v n + b_v,
\]
where \( a_v, b_v \) are integers and \( a_v \geq 1 \) for \( v \in F_2 \). Further, denote
\[
s = \prod_{v \in F_2} a_v.
\]
Now, suppose\[
\{g_1^m(w) \mid n \geq 0\} \subseteq L(G_2)
\]
and fix an integer \( j, 0 \leq j < s \). Let
\[
F_3 = \{v \in F_2 \mid a_v \text{ divides } a_j + b - b_v\}
\]
and denote
\[
c_v = \frac{a_s}{a_v}, \quad d_v = \frac{a_j + b - b_v}{a_v}
\]
whenever \( v \in F_3 \). Then, for almost all \( n \geq 0 \), we have
\[
g_1^{n+j}(w) \in \{g_2^{n+d_v}(v) \mid v \in F_3\}.
\]
By Theorem 5 we obtain a semialgorithm for the inclusion
\[
\{g_1^{n+j}(w) \mid n \geq 0\} \subseteq L(G_2).
\]
By repeating this step for \( j = 0, 1, \ldots, s-1 \), we get a semialgorithm for the inclusion
\[
\{g_1^n(w) \mid n \geq 0\} \subseteq L(G_2).
\]

4.2 Exponential DF0L systems

Assume that \( G_i = (X, g_i, F_i), i = 1, 2, \) are exponential DF0L systems such that the languages \( L(G_i), i = 1, 2, \) are full.

First, we claim that there is an HD0L sequence \( (S_1(n))_{n \geq 0} \) such that the following conditions hold:

1. \( L(G_1) = \{S_1(n) \mid n \geq 0\}, \)
2. \( S_1(m) \neq S_1(n) \) whenever \( m \neq n, m, n \geq 0, \)
3. there exist positive real numbers \( c_1, d_1, \beta \) with \( \beta > 1 \), and a nonnegative integer \( s \) such that
\[
c_1 n^s \beta^n \leq |S_1(n)| \leq d_1 n^s \beta^n
\]
for almost all \( n \geq 0 \).
To prove the claim, suppose first that
\[ g_i^m(w_1) \neq g_i^n(w_2) \]
whenever \( m, n \geq 0, w_1, w_2 \in F_1 \) and \( w_1 \neq w_2 \). Then the sequence \((S_1(n))_{n \geq 0}\) is obtained simply by merging the DOL sequences \((g_i^n(w))_{n \geq 0}\) for \( w \in F_1 \). Condition 3 follows because the merged sequences have equal growth orders.

In the general case we first construct a DF0L system \( H = (X, h, F_3) \) and a finite set \( F_4 \) such that
\[ L(G_1) = L(H) \cup F_4 \]
and
\[ h^m(v_1) \neq h^n(v_2) \]
whenever \( m, n \geq 0, v_1, v_2 \in F_3 \) and \( v_1 \neq v_2 \). The existence of \( H \) follows by a straightforward induction on the cardinality of \( F_1 \). (This step is explained more fully in [1].) The existence of \((S_1(n))_{n \geq 0}\) then follows as above.

Similarly, there is an HDOL sequence \((S_2(n))_{n \geq 0}\) such that the following conditions hold:
1. \( L(G_1) = \{S_2(n) \mid n \geq 0\} \),
2. \( S_2(m) \neq S_2(n) \) whenever \( m \neq n, m, n \geq 0 \),
3. there exist positive real numbers \( c_2, d_2, \gamma \) with \( \gamma > 1 \), and a nonnegative integer \( t \) such that
\[ c_2 n^t \gamma^n \leq |S_2(n)| \leq d_2 n^t \gamma^n \]
for almost all \( n \geq 0 \).

Suppose now that \( L(G_1) = L(G_2) \). Then necessarily \( \beta = \gamma \) and \( s = t \). Indeed, if either \( \beta = \gamma \) and \( s > t \), or \( \beta > \gamma \), then for a large \( n_1 \) there are at most \( n_1 \) terms in \((S_1(n))_{n \geq 0}\) of length smaller than \( c_1 n_1^t \beta^n_1 \) while the sequence \((S_2(n))_{n \geq 0}\) contains more than \( n_1 \) such terms.

Consequently, if \( L(G_1) = L(G_2) \), there is a positive integer \( a \) such that
\[ S_1(n) \in \{S_2(n + i) \mid -a \leq i \leq a \text{ and } n + i \geq 0\} \]
and
\[ S_2(n) \in \{S_1(n + i) \mid -a \leq i \leq a \text{ and } n + i \geq 0\} \]
for all \( n \geq 0 \). In other words, the HDOL sequence \((S_1(n))_{n \geq 0}\) is covered by the HDOL sequences
\[ (S_2(n + i))_{n \geq 0}, \quad -a \leq i \leq a, \]
and the HDOL sequence \((S_2(n))_{n \geq 0}\) is covered by the HDOL sequences
\[ (S_1(n + i))_{n \geq 0}, \quad -a \leq i \leq a. \]
(Here \( S_1(n + i) = S_2(n + i) = \epsilon \) if \( n + i < 0 \).) By Theorem 5 this gives a semialgorithm for the language equivalence of two exponential DF0L systems generating full languages.
4.3 D0L systems

As a special case the previous subsections give a new algorithm for the D0L language equivalence problem. In particular, the decidability of the language equivalence problem for exponential D0L systems is an almost immediate consequence of the decidability of the D0L covering problem.

References


CS&P'2002
Berlin
October 7-9,
2002

Hans-Dieter Burkhard
Andrzej Skowron
Andrzej Salwicki
Wojciech Penczek
Michael Köhler
Berndt Farwer
Dagmar Monett Díaz

Miriam Minor

Agata Półroła

Matthias Jüngel

Maciej Szreter

Bożena Woźna
Nataly S. Moskalyova

Manfred Kudlek

Simone Tini

Amin Coja-Oghlan

Magdalena Kacprzak

Uri Abraham
Organizing Staff

Gabriela Lindemann
Renate Zirkelbach
Adelheid Lau
Christine Jung

Wilhelm von Humboldt

Alexander von Humboldt

Humboldt Universität zu Berlin
A case against using Stirling's formula
(unless you really need it)

Martin Dietzfelbinger and Manfred Kunde
Technische Universität Ilmenau, Germany
Email: {martin.dietzf\elbinger\manfred.kunde}@tu-ilmenau.de

1 Aim of this note

Stirling's formula gives tight bounds on \( n! \), but it has a nontrivial proof. This famous formula, which reads

\[
n! \sim \sqrt{2\pi n} \left( \frac{n}{e} \right)^n,
\]

is used in many texts on algorithmics and in some research papers to prove statements that have much simpler proofs. We argue that for standard applications in algorithm analysis, slightly weaker estimates with simple proofs (some well known, some seemingly less known) are good alternatives.

2 Estimates without technique

Example 1: Lower bounds for sorting by comparisons. — Assume we have argued that comparison trees that sort \( n \) objects have \( n! \) leaves, one for each order type. It follows that in the comparison model sorting \( n \) objects needs at least \( \lceil \log_2(n!) \rceil \) comparisons in the worst case and \( \log_2(n!) \) comparisons in the average case. Thus, we need lower bounds for \( \log_2(n!) \). A standard argument is that "by Stirling's formula"

\[
\log_2(n!) = n \log_2 n - (\log_2 e)n + \frac{\log_2 n}{2} + \Omega(1).
\]

But do we really need to invoke this strong statement? How about

\[
\frac{n^n}{n!} < \sum_{i=0}^{n} \frac{n^i}{i!} = e^n,
\]

hence

\[
n! > \left( \frac{n}{e} \right)^n,
\]

hence

\[
\log_2(n!) > n \log_2 n - (\log_2 e)n > n \log_2 n - 1.443n,
\]

done?
Even though (3) has a two-line proof, it is quite a tight bound:

**Claim:**

\[ n! \leq \frac{n^{n+1}}{e^{n-1}}, \text{ for } n \geq 1. \]  

(4)

This is proved by induction on \( n \). The case \( n = 1 \) is trivial, hence assume \( n \geq 2 \) and the claim being true for \( n - 1 \). Recall that \( 1 + x \leq e^x \) for all real numbers \( x \) (with equality only for \( x = 0 \)), and hence \( \left( \frac{n-1}{n} \right)^n = \left( 1 - \frac{1}{n} \right)^n < e^{-1} \). Thus,

\[ n! = (n-1)! \cdot n \cdot \left( \frac{n-1}{n} \right)^n \leq \frac{(n-1)^n}{e^{n-2}} \cdot n = \frac{n^{n+1}}{e^{n-2}} \cdot \left( \frac{n-1}{n} \right)^n < \frac{n^{n+1}}{e^{n-1}}. \]

Taking logarithms, we obtain

\[ \log_2(n!) \leq n \log_2 n - (\log_2 e)n + \log_2(e^n), \]

which shows that the lower bound in (3) is only \( O(\log n) \) away from the true value.

(The reader is invited to provide a similar induction proof for the inequality \( n! \geq n^n/e^{n-1} \).)

**Example 2:** Estimating binomial coefficients. — Often, Stirling's formula is invoked as an argument for the following inequality to hold:

\[ \binom{n}{k} < \left( \frac{en}{k} \right)^k, \text{ for } n, k \geq 1. \]

(5)

But again, this is an almost obvious fact, and Stirling's formula is too heavy a tool to be used in its proof: Simply use (1) for \( k \) to conclude

\[ \binom{n}{k} \leq \frac{n^k}{k!} < \frac{e^k \cdot n^k}{k^k}. \]

Alternatively, note that for \( 1 \leq i \leq k \leq n \) we have \((k/n)^k \leq (k/n)^i\) and hence

\[ \sum_{0 \leq i \leq k} \binom{n}{i} \left( \frac{k}{n} \right)^i < \sum_{0 \leq i \leq n} \binom{n}{i} \left( \frac{k}{n} \right)^i = \left( 1 + \frac{k}{n} \right)^n < e^k \]

to conclude that even

\[ \sum_{0 \leq i \leq k} \binom{n}{i} < \left( \frac{en}{k} \right)^k, \text{ for } 1 \leq k \leq n. \]

### 3 Stirling's formula

Stirling's formula is a very elegant classical result that gives a precise description of the asymptotic behaviour of \( n! \). In texts, we find various formulations under the label "Stirling's formula":

\[ n! \sim \sqrt{2\pi n} \left( \frac{n}{e} \right)^n, \text{ i.e., } \lim_{n \to \infty} \frac{n!}{\sqrt{2\pi n} \left( n/e \right)^n} = 1; \]

(6)
These formulas with their asymptotic notation suggest that they make a statement only for \( n \) that are sufficiently large. A more precise statement, which gives information for all \( n \geq 1 \), can be found e.g. in [5]:

\[
\sqrt{2\pi n \cdot \left( \frac{n}{e} \right)^n \cdot e^{1/(12n+1)}} < n! < \sqrt{2\pi n \cdot \left( \frac{n}{e} \right)^n \cdot e^{1/(12n)}} , \text{ for } n \geq 1 .
\] (8)

Even more precise, but again asymptotic, are formulations like in [4, Table 452]:

\[
n! = \sqrt{2\pi n \left( \frac{n}{e} \right)^n \left( 1 + \frac{1}{12n} + \frac{1}{288n^2} - \frac{139}{51840n^3} + O \left( \frac{1}{n^4} \right) \right)} .
\] (9)

These statements are very accurate. Neither of them, though, has a really simple proof, and in classroom there is no other chance than just to quote them. So it is to be feared that the student has no clue as to why the statement should be true. Some authors even seem to be tempted to apply an asymptotic statement like (6) to derive inequalities for finite numbers, like (5).

## 4 Comparing sums and integrals

Using only basic calculus, quite tight upper and lower bounds on \( n! \) may be proved easily. The method is well-known; it applies to arbitrary monotone functions, see e.g., [3], [2], and [1]. So nothing is new here.

Consider the natural logarithm of \( n! \):

\[
S(n) = \ln(n!) = \sum_{1 \leq i \leq n} \ln(i) .
\]
The simple idea is to compare the sum with the integral
\[ I(n) = \int_1^n \ln x \, dx = [x(\ln x - 1)]_1^n = n \ln n - n + 1. \]

Just by looking at Fig. 1, in which two step functions are depicted that form upper and lower bounds for \(\ln x\), we see (for \(n \geq 2\)) that
\[ \sum_{1 \leq i \leq n-1} \ln(i) < I(n) < \sum_{1 \leq i \leq n} \ln(i), \]
hence
\[ I(n) < S(n) < I(n) + \ln n. \]
Exponentiating, we obtain
\[ e \cdot \left(\frac{n}{e}\right)^n \leq n! < e n \cdot \left(\frac{n}{e}\right)^n. \]

The distance between upper and lower bound is only a factor of \(n\). Note that the upper bound coincides with (4).

5 Approximating \(n!\) with constant relative error

Naturally, it would be nice to have estimates for \(n!\) that are correct up to a constant factor. By looking at Fig. 1 again, we see that the curve \(\ln x\) cuts the rectangles between the lower and the upper step function into two areas that closely resemble triangles. So we expect that the lower step function underestimates \(I(n)\) by \(\frac{1}{2} \ln n + O(1)\), the upper step function overestimates it by \(\frac{1}{2} \ln n - O(1)\). Exponentiating, this makes us expect that \(n! = \Theta(\sqrt{n} \cdot (n/e)^n)\). In this section, we give proofs for bounds of this kind, by bounding the area below the curve \(\ln x\) by trapezoids instead of rectangles.

For bounding \(I(n)\) from below we employ a piecewise linear interpolation of the function \(\ln x\), see Fig. 2. Since \(\ln x\) is a concave function (the second derivative

![Figure 2: Lower bounding \(I(n)\) with interpolating line segments](image)

\[ \frac{d^2}{dx^2} \ln x = -\frac{1}{x^2} \text{ is negative throughout}, \text{ the graph of } \ln x \text{ in } [i-1, i] \text{ lies above} \]
the line segment that connects the points \((i - 1, \ln(i - 1))\) and \((i, \ln i)\); hence we have

\[
\int_{i-1}^{i} \ln x \, dx > \frac{\ln(i - 1) + \ln i}{2}, \quad \text{for } i \geq 2.
\]

Thus,

\[
I(n) > \sum_{2 \leq i \leq n} \frac{\ln(i - 1) + \ln i}{2} = S(n) - \frac{\ln n}{2},
\]

or

\[
S(n) < I(n) + \frac{\ln n}{2} = n \ln n - n + 1 + \frac{\ln n}{2}.
\]

Exponentiating, we get

\[
n! < e^{\sqrt{n} \cdot \left(\frac{n}{e}\right)^n}.
\]

(Compare the constant \(e \approx 2.718\) with the asymptotically correct constant \(\sqrt{2\pi} \approx 2.51\).)

For bounding \(I(n)\) from above, we note that since \(\ln x\) is a concave function, in \([i - \frac{1}{2}, i + \frac{1}{2}]\) its graph runs below the tangent in the point \((i, \ln i)\), cf. Figure 3. Obviously, the area of the trapezoid between the lines \(x = i - \frac{1}{2}\) and \(x = i + \frac{1}{2}\),

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure3.png}
\caption{Upper bounding \(I(n)\) with tangents}
\end{figure}

the \(x\)-axis, and the tangent is \(\ln i\). Hence

\[
\int_{i-\frac{1}{2}}^{i+\frac{1}{2}} \ln x \, dx < \ln i, \quad \text{for } i \geq 2.
\]

Summing this for \(2 \leq i \leq n - 1\) and making corrections for the two intervals \([1, \frac{3}{2}]\) and \([n - \frac{1}{2}, n]\) of the \(x\)-axis that are not covered, we obtain

\[
I(n) < \int_{1}^{3/2} \ln x \, dx + \sum_{2 \leq i \leq n-1} \ln i + \int_{n-\frac{1}{2}}^{n} \ln x \, dx
\]

\[
< \frac{3 \ln(3/2) - 3}{2} + 1 + (S(n) - \ln n) + \frac{1}{2} \ln n. \tag{12}
\]

We substitute the value of \(I(n)\) and rearrange to obtain

\[
S(n) > n \ln n - n + \frac{\ln n}{2} + \frac{3 - 3 \ln(3/2)}{2}
\]
Exponentiating yields

\[ n! > \left( \frac{2e}{3} \right)^{3/2} \cdot \sqrt{n} \cdot \left( \frac{n}{e} \right)^n > 2.439 \sqrt{n} \cdot \left( \frac{n}{e} \right)^n. \]

The constant 2.439 in this lower bound deviates from the asymptotically correct constant \( \sqrt{2\pi} \approx 2.51 \) by less than 2.5 percent.

6 Conclusion

There are reasonable and useful upper and lower bounds for \( n! \) that have extremely simple proofs. On the basis of a little calculus, the bounds obtained by comparing \( \ln(n!) \) with the integral \( \int_n^\infty \ln x \, dx \) are more or less obvious, and are correct up to a factor of \( n \). Approximating the integral by trapezoids leads to easy proofs of bounds for \( n! \) that are correct up to a small relative error. Thus, for standard applications in algorithmics, a reference to Stirling’s formula as a black box is usually unnecessary. Direct proofs for the bounds will help students in understanding why the asymptotic behaviour of \( n! \) is \( \Theta(\sqrt{n}(n/e)^n) \), much more than a reference to any exact version of Stirling's formula.

References


On Iterated Scattered Deletion

Michael Domaratzki*
School of Computing Queen's University,
Kingston, ON K7L 3N6 Canada
email: domaratzi@cs.queensu.ca

In this note, we solve an open problem of Ito et al. [1] on iterated scattered deletion. Let \( \Sigma \) be an alphabet. The scattered deletion [2, 4] of two words \( x, y \in \Sigma^* \), denoted by \( x \sim y \), is defined as

\[
x \sim y = \{ x_1 x_2 \cdots x_k : y = y_1 \cdots y_{k-1}; x = x_1 y_1 x_2 y_2 \cdots x_{k-1} y_{k-1} x_k; x_i, y_i \in \Sigma^* \}.
\]

Thus, \( x \sim y \) is the set of words which result from deleting \( y \) as a scattered subword from \( x \). We extend this operation to languages \( L_1, L_2 \subseteq \Sigma^* \) as follows:

\[
L_1 \sim L_2 = \bigcup_{x \in L_1} \bigcup_{y \in L_2} x \sim y.
\]

For unexplained notions in formal language and automata theory, please see Yu [5]. For languages \( L_1, \cdots, L_k \), we use the notation \( \prod_{i=1}^k L_i = L_1 L_2 \cdots L_k \).

We now define an iterated scattered deletion operation [1]. Let \( i > 1 \), \( L \subseteq \Sigma^* \). Then \( (\sim)^i(L) \) is defined recursively as follows:

\[
(\sim)^0(L) = L;
(\sim)^i+1(L) = (\sim)^i(L) \sim (\sim)^i(L) \cup \{ \epsilon \} \quad \forall i \geq 1.
\]

Then the iterated scattered deletion operator \( (\sim)^+(L) \) is given by

\[
(\sim)^+(L) = \bigcup_{i \geq 1} (\sim)^i(L).
\]

We also define an auxiliary operation \( L_1[\sim]^i L_2 \) which is defined recursively for all \( i \geq 0 \) as follows:

\[
L_1[\sim]^0 L_2 = L_1;
L_2[\sim]^{i+1} L_2 = (L_1[\sim]^i L_2) \sim L_2 \quad \forall i \geq 1.
\]

We then set

\[
L_1[\sim]^* L_2 = \bigcup_{i \geq 0} L_1[\sim]^i L_2.
\]

Ito et al. [1] asked whether the regular languages are closed under \( (\sim)^+ \). We show that they are not.

*Research supported in part by an NSERC PGS-B graduate scholarship.
Let \( k \geq 2 \) be arbitrary, and let \( \Sigma_k = \{\alpha_i, \beta_i, \gamma_i, \eta_i\}_{i=1}^k \). Then we define 
\( L_k \subseteq \Sigma_k^* \) as 
\[
L_k = \prod_{i=1}^{k}(\alpha_i\beta_i)^* \prod_{i=1}^{k}(\gamma_i\eta_i)^* + \bigcup_{i=1}^{k}\beta_i\eta_i.
\]
We claim that 
\[
(\sim)^+(L_k) \cap \prod_{i=1}^{k}\alpha_i^* \prod_{i=1}^{k}\gamma_i^* = \{\alpha_1^0\alpha_2^i\cdots\alpha_k^i\gamma_1^i\gamma_2^i\cdots\gamma_k^i : i_j \geq 0\}.
\]
and that 
\[
(\sim)^+(L_k)
\]
cannot be expressed as the intersection of \( k-1 \) context-free languages.

We first establish (1). Let \((i_1, i_2, \ldots, i_k) \in \mathbb{N}^k \). Then note that 
\[
\prod_{j=1}^{k}\alpha_j^j \prod_{j=1}^{k}\gamma_j^j \in (\cdots((\prod_{j=1}^{k}(\alpha_j^j\beta_j)\gamma_j\eta_j)^j)\cdots)\cdots(\sim)^{k}\beta_k\eta_k.
\]
This establishes the right-to-left inclusion of (1). We now show the reverse inclusion. First, note that if \( \alpha \in (\sim)^+(L_k) \), then we can write \( \alpha = x_1x_2\cdots x_ky_1y_2\cdots y_k \) where \( x_i \in \{\alpha_i, \beta_i\}^* \) and \( y_i \in \{\gamma_i, \eta_i\}^* \). To prove the left-to-right inclusion of (1), we will require the following stronger claim:

**Claim 1** Let \( x_1x_2\cdots x_ky_1y_2\cdots y_k \in (\sim)^+(L_k) \) where \( x_i \in \{\alpha_i, \beta_i\}^* \) and \( y_i \in \{\gamma_i, \eta_i\}^* \) for all \( 1 \leq i \leq k \). Then for all \( 1 \leq i \leq k \), the following equalities hold:
\[
|x_i|_{\alpha_i} - |x_i|_{\beta_i} = |y_i|_{\gamma_i} - |y_i|_{\eta_i}.
\]

**Proof.** Let \( z = x_1x_2\cdots x_ky_1y_2\cdots y_k \in (\sim)^+(L_k) \). Then there exists some \( i \geq 1 \) such that \( z \in (\sim)^i(L_k) \). The proof is by induction on \( i \). For \( i = 1 \), \( z \in L_k \). Thus, we see that either

(a) for all \( 1 \leq i \leq k \), \( x_i = (\alpha_i\beta_i)^j_i \) for some \( j_i \geq 0 \) and \( y_i = (\gamma_i\eta_i)^j_i \) for some \( j_i \geq 0 \), in which case \( |x_i|_{\alpha_i} - |x_i|_{\beta_i} = 0 = |y_i|_{\gamma_i} - |y_i|_{\eta_i} \) or

(b) \( z = \beta_k\eta_k \) for some \( 1 \leq i \leq k \). Thus, \( |x_i|_{\alpha_i} - |x_i|_{\beta_i} = -1 = |y_i|_{\gamma_i} - |y_i|_{\eta_i} \) and \( x_j = y_j = \epsilon \) for all \( j \neq i \) with \( 1 \leq j \leq k \).

Thus, the result holds for \( i = 1 \).

Assume the claim holds for all natural numbers less than \( i \). Let \( z \in (\sim)^i(L_k) \). Then there exists some \( \theta \in (\sim)^{i-1}(L_k) \) and \( \zeta \in (\sim)^{i-1}(L_k) + \epsilon \) such that \( z \in \theta \sim \zeta \). If \( \zeta = \epsilon \), then \( z = \theta \) and the result holds by induction. Thus, let 
\[
\theta = u_1u_2\cdots u_kv_1v_2\cdots v_k
\]
\[
\zeta = s_1s_2\cdots s_kt_1t_2\cdots t_k
\]
so that \( u_i, s_i \in \{\alpha_i, \beta_i\}^* \) and \( v_i, t_i \in \{\gamma_i, \eta_i\}^* \) for all \( 1 \leq i \leq k \). Then note that 
\[
|x_i|_{\alpha_i} = |u_i|_{\alpha_i} - |s_i|_{\alpha_i};
\]
\[
|x_i|_{\beta_i} = |u_i|_{\beta_i} - |s_i|_{\beta_i};
\]
\[
|y_i|_{\gamma_i} = |v_i|_{\gamma_i} - |t_i|_{\gamma_i};
\]
\[
|y_i|_{\eta_i} = |v_i|_{\eta_i} - |t_i|_{\eta_i}.
\]
Thus, by induction, we can easily establish that the desired equalities hold.

We now show that \( (\sim) ^+ (L_k) \) cannot be expressed as the intersection of \( k-1 \) context-free languages. Let \( CFL_k \) be the class of languages which are expressible as the intersection of \( k \) CFLs. The following lemma is obvious, since the CFLs are closed under intersection with regular languages and homomorphism.

\textbf{Lemma 2} \( CFL_k \) is closed under intersection with regular languages and homomorphism.

The following result is due to Liu and Weiner [3, Thm. 8]:

\textbf{Theorem 3} Let \( k \geq 2 \). Let \( L'_k = \{ \alpha _1^i \alpha _2^j \cdots \alpha _k^i \alpha _1^j \alpha _2^j \cdots \alpha _k^j : i_j \geq 0 \} \). Then \( L'_k \in CFL_k - CFL_{k-1} \).

Let \( \Delta_k = \{ \gamma _i, \alpha_i \}^k \). Let \( h_k : \Delta_k ^* \rightarrow \Delta_k ^* \) be given by \( h_k (\alpha_i) = h_k (\gamma_i) = \alpha_i \) for all \( 1 \leq i \leq k \). Let \( D_k = (\sim) ^+ (L_k) \). Then \( h_k (D_k \cap \prod _{i=1}^k \alpha_i \prod _{i=1}^k \gamma_i ) = L'_k \). Thus, if \( D_k \in CFL_{k-1} \), then \( L'_k \in CFL_{k-1} \) as well, by Lemma 2. This contradicts Theorem 3. Thus, \( D_k \) cannot be expressed as the intersection of \( k-1 \) CFLs. We have established the following result:

\textbf{Theorem 4} For all \( k \geq 2 \), there exists an \( O(n^{2k-1}) \)-density bounded regular language \( L_k \) such that \( (\sim) ^+ (L_k) \) cannot be expressed as the intersection of \( k-1 \) context-free languages.

\textbf{References}


Periodicity and Unbordered Segments of Words

Tero Harju         Dirk Nowotka

Turku Centre for Computer Science (TUCS)
Department of Mathematics
University of Turku, Finland

We shall give an introduction to the problem area concerning the well known
Duval’s conjecture, which was announced to be solved in [12].

1 Introduction

In 1979 Ehrenfeucht and Silberger published an article [9] where the relationship
between the length of a word and the maximum length of its unbordered factors
(segments) was for the first time investigated. Periodicity and borderedness are
two basic properties of words that play a rôle in many areas of computer science
such as string searching algorithms [13, 3, 7], data compression [20, 6], and codes
[2], which are classical examples, but also computational biology, e.g., sequence
assembly [17] or superstrings [4], and serial data communications systems [5].
It is well known that these two word properties do not exist independently from
each other. However, no clear relationship has been established so far, despite
substantial recent progress in that area. We chose the title of Ehrenfeucht and
Silberger’s paper for our essay to underline that this paper illustrates the way
this line of research has evolved in the last 24 years.

In Section 2 we give a historical overview on this line of research, its main
results and conjectures so far. This will lead to the concept of Duval extensions
which are introduced in Section 3. We conclude with Section 4.

First, we shall introduce the main notations of this paper. We refer the
reader to [14, 15] for more basic and general definitions.

Consider a finite alphabet A of letters. Let A* denote the monoid of all
finite words over A including the empty word, denoted by ε. Let w ∈ A*.
Then we can express w as a sequence of letters w(1)w(2) · · · w(n) where w(i) ∈ A is
a letter, for every 1 ≤ i ≤ n. Let 1 ≤ i ≤ j ≤ n, then w(i)w(i+1) · · · w(j) is called
a factor or segment of w. We denote the length n of w by |w|. Note, that |ε| = 0.
A word w is called primitive if it cannot be factored such that w = uk for some
k ≥ 2. Let w = uv for some words u and v. Then vu is called conjugate of w.

A nonempty word u is called a border of a word w, if w = uv = v'u for some
words v and v'. We call w bordered, if it has a border that is shorter than w,
otherwise w is called unbordered. Suppose w = uv, then u is called a prefix of
w denoted by u ≤ w.
Let $\prec$ be an ordering of $A = \{a_1, a_2, \ldots, a_n\}$, say $a_1 \prec a_2 \prec \cdots \prec a_n$. Then $\prec$ induces a lexicographic order, also denoted by $\prec$, on $A^*$ such that

$$u \prec v \iff u \leq v \text{ or } u = xau' \text{ and } v = xbu' \text{ with } a \prec b$$

where $a, b \in A$.

Let us consider the following examples. Let $A = \{a, b\}$ and $u, v, w \in A^*$ such that $u = abaa$ and $v = baaba$ and $w = abaaba$. Then $u$ and $v$ are primitive, but $w$ is not. Furthermore, $\{aaab, aaba, abaa, baaa\}$ is the set of all conjugates of $u$. Let $a \prec b$. Then $u \prec w \prec v$. The largest unbordered factor of $w$ has length three.

## 2 Maximum Length of Unbordered Factors

When the length of unbordered factors of a word is investigated, that is usually done in terms of the length of the word and its minimum period.

Let us make our terminology more precise. Consider a word $w$ of length $n$ over some alphabet $A$. An integer $1 \leq p \leq n$ is a period of $w$, if $w_i = w_{i+p}$ for all $1 \leq i \leq n-p$. The smallest period of $w$ is called the minimum period (or simply, the period) of $w$, denoted by $\partial(w)$. Let $\mu(w)$ denote the maximum length of unbordered factors of $w$. For example, let $w = abaabbaaba$, then $\partial(w) = 7$ and $\mu(w) = 6$.

Clearly, the maximum length of unbordered factors $\mu(w)$ of $w$ is bound by the period $\partial(w)$ of $w$. We have

$$\mu(w) \leq \partial(w)$$

since for every factor $v$ of $w$, with $\partial(w) < |v|$, the prefix $v_{(1)} v_{(2)} \cdots v_{(|v| - \partial(w))}$ of $v$ is also a suffix of $v$ by the definition of period.

It is a natural question to ask at what length of $w$ is $\mu(w)$ necessarily maximal, that is, $\mu(w) = \partial(w)$. Of course, the length of $w$ is considered with respect to either $\mu(w)$ or $\partial(w)$.

In 1979 Ehrenfeucht and Silberger [9], as well as, Assous and Pouzet [1] addressed this question first. Ehrenfeucht and Silberger [9] stated

**Theorem 1.** If $2\partial(w) \leq |w|$ then $\mu(w) = \partial(w)$.

They also established that every primitive word $w$ has at least $\sigma$-many unbordered conjugates, where $\sigma$ is the number of different letters occurring in $w$, which leads directly to

**Theorem 2.** If $2\partial(w) - \sigma \leq |w|$ then $\mu(w) = \partial(w)$.

However, this result was stated by Duval [8] only in 1982.

The real challenge, though, turned out to be giving a bound on the length of $w$ with respect to $\mu(w)$. It was conjectured in [9] that $2\mu(w) \leq |w|$ implies $\mu(w) = \partial(w)$. However, Assous and Pouzet [1] gave the following counter example.
Example 3. Let
\[ w = a^n b a^{n+1} b a^{n+2} b a^n b a^{n+1} b a^n \]
for which \( \partial(w) = 4n + 7 \) and \( \mu(w) = 3n + 6 \) and \( |w| = 7n + 10 = 7/3 \mu(w) - 4 \).

Assous and Pouzet also gave the following conjecture.

Conjecture 4. Let \( f : \mathbb{N} \to \mathbb{N} \) such that \( f(\mu(w)) \leq |w| \) implies \( \mu(w) = \partial(w) \).
Then
\[ f(\mu(w)) \leq 3\mu(w) . \]

In 1982 Duval [8] established the following result.

Theorem 5. If \( 4\partial(w) - 6 \leq |w| \) then \( \mu(w) = \partial(w) \).

He also stated Conjecture 7 (see Duval’s conjecture in the following section) about what was later called Duval extensions, a conjecture that implies

If \( 3\partial(w) \leq |w| \) then \( \mu(w) = \partial(w) . \)

3 Duval Extensions

In the previous section we recalled a question initially raised by Ehrenfeucht and Silberger [9]. The problem was to estimate a bound on the length of \( w \), depending on \( \mu(w) \), such that \( \mu(w) = \partial(w) \). Duval [8] introduced a restricted version of that problem by assuming that \( w \) has an unbordered prefix of length \( \mu(w) \). Let us fix some more notations first.

Let \( w \) and \( u \) be nonempty words where \( w \) is also unbordered. We call \( wu \) a Duval extension of \( w \), if every factor of \( wu \) longer than \( |w| \) is bordered, that is, \( \mu(wu) = |w| \). A Duval extension \( wu \) is called trivial, if \( \partial(wu) = \mu(wu) \). A nontrivial Duval extension \( wu \) of \( w \) is called minimal, if \( u \) is of minimal length, that is, \( u = u'a \) and \( w = u'bw' \) where \( a, b \in A \) and \( a \neq b \).

Example 6. Let \( w = ababababababb \) and \( u = aaba \). Then
\[ w . u = ababababababab.aaba \]
(for the sake of readability, we use a dot to mark where \( w \) ends) is a nontrivial Duval extension of \( w \) of length \( |wu| = 18 \), where \( \mu(wu) = |w| = 14 \) and \( \partial(wu) = 15 \). However, \( wu \) is not a minimal Duval extension, whereas
\[ w . u' = abababababab.aaba \]
is minimal, with \( u' = aa \leq u \). Note, that \( wu \) is not the longest nontrivial Duval extension of \( w \) since
\[ w . v = ababababababb.abaaba \]
is longer, with \( v = ababa \) and \( |vw| = 20 \) and \( \partial(wu) = 17 \). One can check that \( wv \) is a nontrivial Duval extension of \( w \) of maximum length, and at the same time \( wv \) is also a minimal Duval extension of \( w \).
In 1982 Duval [8] stated the following conjecture.

**Conjecture 7 (Duval).** Let $wu$ be a nontrivial Duval extension of $w$. Then $|u| < |w|$.

It follows directly from this conjecture that for any word $w$, we have that $3\mu(w) \leq |w|$ implies $\mu(w) = \partial(w)$. This conjecture remained popular throughout the years, see for example Chapter 8 in [15]. However, recently the authors of this essay established [12] the following result which implies Duval's conjecture.

**Theorem 8.** Let $wu$ be a nontrivial Duval extension of $w$. Then $|u| < |w| - 1$.

We have the following corollary.

**Corollary 9.** If $3\mu(w) - 2 \leq |w|$ then $\mu(w) = \partial(w)$.

This corollary gives the best bound on the general case so far. However, this result does not give a final answer to the question about the relation between $|w|$ and $\mu(w)$. Since the best example known to us is Example 3 in the previous section which shows that there is an arbitrary long word $w$ such that $|w| = 7/3\mu(w) - 4$ and $\mu(w) < \partial(w)$. In general the precise bound is still unknown.

Nevertheless, the bound of Theorem 8 is tight as the following example shows.

**Example 10.** Let $w = a^i ba^{i+j} bb$ and $u = a^{i+j} ba^i$ where $i, j \geq 1$. It is easy to check that $\ w. u = a^i ba^{i+j} bb.a^{i+j} ba^i$ is a nontrivial Duval extension of $w$ of length $|w| - 2$.

Further knowledge about structural properties of Duval extension such as the following conjecture would certainly help to estimate the precise bound in the general case. Let us call a nontrivial Duval extension $wu$ of $w$ maximal if $|u| = |w| - 2$.

**Conjecture 11.** Let $w = w'ab^k$ for some $k \geq 1$. If $wu$ is a maximal Duval extension of $w$, then $b^k$ does not occur in $w'$.

A further property of Duval extensions was established in [11]. Recall, that a minimal Duval extension of a word $w$ is the smallest prefix of a nontrivial Duval extension of $w$ such that the prefix itself is a nontrivial Duval extension of $w$. The following theorem gives another property of nontrivial Duval extensions.

**Theorem 12.** Let $wu$ be a minimal Duval extension of $w$. Then $u$ is a factor of $w$.

Duval extensions have also become a subject of interest on their own. In particular the set of words having no nontrivial Duval extension has been investigated in [10] and [18]. The first attempt to characterize the set of words
that have no nontrivial Duval extension was done investigating well-known sets of words like unbordered finite factors of Sturmian words and Lyndon words.

Infinite words of minimal subword complexity are called Sturmian words, cf. [19, 15]. Minimal subword complexity means that a Sturmian word contains exactly \( n + 1 \) different factors of length \( n \) for every \( n \geq 1 \). Let us consider finite factors of Sturmian words in the following, and lets simply call them Sturmian words. Mignosi and Zamboni showed the following uniqueness result for Duval extensions in [18].

**Theorem 13.** Unbordered Sturmian words have no nontrivial Duval extension.

This result was improved by the authors of this paper in [10] to Lyndon words. Let a primitive word \( w \) be called Lyndon word if it is minimal among its conjugates, that is, if \( w < v \) for every conjugate \( v \) of \( w \) and some arbitrary order \( < \) on \( A \), cf. [16, 15]. Note, that Lyndon words are unbordered.

**Theorem 14.** Lyndon words have no nontrivial Duval extension.


**Theorem 15.** Every unbordered Sturmian word is a Lyndon word.

The converse of Theorem 15 is certainly not true. Indeed, consider the word \( aabbab \) which is a Lyndon word but not a Sturmian word since it contains four factors of length two. A precise characterization of the set of words that have no nontrivial Duval extension is still unknown.

4 Conclusions

We have recalled the problem of estimating the relationship between the length of a word and the maximum length of its unbordered factors. The final answer is still unknown. However, quite some progress has been made since the problem was raised in 1979. In particular the special case of Duval extensions raised attention and led to new results. For example, the long standing conjecture by Duval was just recently solved. However, open problems remain about the structure of Duval extensions and words that have no nontrivial Duval extension. Further research on those questions are likely to lead to a final answer to the general case.

References


On Minimising P Finite State Automata

Kamala Krithivasan and Sandeep V. Varma

Department of Computer Science and Engineering
IIT Madras, India

1 Introduction

P systems were introduced in [5], and this is an area of current interest [6] in natural computing. However, accepting P systems were only recently considered, [1, 2], etc. In [3] P finite state automata and P pushdown automata were introduced. The model is different from the ones discussed in [1, 2, 4]. In this paper, we consider the problem of constructing P finite state automata with as few states as possible. Membranes are more natural than states in P systems, and hence the interest is in minimising the number of states.

2 Basic Definitions

Definition 2.1. A P finite state automaton (abbreviated PFSA) is an 8-tuple $M = (K, \Sigma, \mu, \Delta, q_0, F, i_0, \mu_{\text{final}})$, where

- $K$ is a finite set of states;
- $\Sigma$ is a finite alphabet;
- $\mu$ is a membrane structure having $k$ membranes labelled by $1, 2, \ldots, k$;
- $\Delta = (\delta_1, \delta_2, \ldots, \delta_k)$, where for $1 \leq i \leq k$, $\delta_i$ is a mapping from $K \times (\Sigma \cup \{\lambda\})$ to $2^{K \times \text{TAR}}$, where $\lambda$ is the empty string, and $\text{TAR} = \{\text{in}_j \mid 1 \leq j \leq k\} \cup \{\text{here, out}\}$;
- $q_0$ is the initial state;
- $F \subseteq K$ is the set of final states;
- $i_0$, where $1 \leq i_0 \leq k$, is the initial membrane;
- $\mu_{\text{final}} \subseteq \{1, 2, \ldots, k\}$ is the set of final membranes.

Note that the above definition differs from the one given in [3] in that $\lambda$-transitions have also been allowed. By definition, any such automaton is nondeterministic. We call it deterministic if the range of each $\delta_i$, $1 \leq i \leq k$, consists of singleton or empty sets, and $\lambda$-transitions are not allowed (i.e., for each $i$, $1 \leq i \leq k$ and each $q \in K$, $\delta_i(q, \lambda) = \emptyset$).

A PFSA consists of a membrane structure whose membranes have associated transition mappings $\delta_i$. The input string is initially placed in a specified membrane labelled by $i_0$. Like in a finite automaton, the symbols of the input string are read one by one, depending on the state of the PFSA and as prescribed by the mapping $\delta_i$ associated with membrane $i$ where the string is placed. One also allows $\lambda$-transitions. After each transition, the remaining string is transferred to the membrane indicated by the target
here, out, in, specified by $\delta_i$. This process is continued until all the symbols are read.

If we exhaust the string in a final state and in a final membrane, then the input string is said to be accepted.

An instantaneous description can be given as a 3-tuple $(q, w, i)$ where $q$ is the current state, $w$ the portion of the input yet to be read and $i$ the membrane in which the input string is. Let $q \in K$, $1 \leq i \leq k$, and $a \in (\Sigma \cup \{\lambda\})$ (if we are considering a deterministic PFSA, then we should have $a \in \Sigma$). If $\delta_i(q, a)$ contains $(p, \text{tar})$ for some $p \in K$, and some tar $\in \{\text{in}, \text{out}\}$, then we write $(q, aw, i) \vdash (p, w, j)$, where (1) if tar $= \text{in}$ and membrane $j'$ is directly inside membrane $i$, then $j = j'$; (2) if tar $= \text{here}$, then $j = i$; (3) if tar $= \text{out}$, then $j$ is the label of the membrane directly containing membrane $i$.

Let $\vdash^*$ be the reflexive and transitive closure of $\vdash$. The language accepted by a PFSA $M$ is defined as

$$L(M) = \{w \mid w \in \Sigma^*, (q_0, w, i_0) \vdash^* (q_f, \lambda, j) \text{ for some } q_f \in F, j \in \mu_{\text{final}}\}.$$

**Remark.** One could think of acceptance defined only via final states or only via final membranes. It is clear that both are special cases of the above construct (the first case being equivalent to $\mu_{\text{final}}$ being the set of all membranes, and the second to $F$ being the set of all states). On the other hand, the regularity of the languages accepted by PFSA (a proof of which is given in the next section) shows that any PFSA can be simulated by a usual FSA, which can be thought of as a PFSA with only one membrane, and hence accepting only via final states. Further, given any FSA we can consider a PFSA with two membranes, the inner one simulating the FSA and such that, if $\delta_{\text{inner}}$ is its rule set, then $\delta_{\text{inner}}(q, \lambda)$ also contains $(q', \text{out})$ for every final state $q$ of the given FSA and a fixed state $q'$, the choice of which does not matter. Then this PFSA, with no rule set for the outer membrane, accepts the same language as the given FSA only via final membranes, on setting the outer membrane to be the only final membrane. Hence, defining acceptance only via final states or only via final membranes does not change the power.

## 3 A Simple Upper Bound

The regularity of the languages accepted by PFSA has been proved in [3] (the addition of $\lambda$-transitions clearly does not affect the result). In [7] the minimization of nondeterministic FSA has been dealt with. The following theorem gives a simple upper bound restricting the product of the number of states and number of membranes in a PFSA.

**Theorem 3.1.** Let a nondeterministic PFSA $M$ with $k$ membranes, $n$ states and accepting a language $L$ be given. If $n_{\text{min}}$ denotes the number of states of a minimal nondeterministic FSA accepting $L$, then we have $kn \geq n_{\text{min}}$.

**Proof.** Let the given PFSA be $M = (K, \Sigma, \mu, \Delta, q_0, F, i_0, \mu_{\text{final}})$, with notation as in Section 2. We first construct a nondeterministic FSA $M' = (Q, \Sigma, \delta', q_0', F')$. Define $Q' = \{[q, j] \mid q \in K, 1 \leq j \leq k\}$. Let $\forall q \in Q, 1 \leq j \leq k, a \in \Sigma$, $\delta'(q, j, a) = \{[q', j'] \mid (q, a, j) \vdash (q', \lambda, j')\}$. Further, set $q_0' = [q_0, i_0]$, and $F' = \{[q, j] \mid q \in F, j \in \mu_{\text{final}}\}$.
It is clear that the above construction encodes the membranes as well as the states of $M$ in the states of $M'$, and that $M'$ simulates $M$, so that $M'$ also accepts $L$ (this proves that $L$ is regular, without having to assume the result in [3]). To complete the proof, just note that, since $M'$ accepts $L$, $n\text{min}$ is less than or equal to the number of states in $M'$, which in turn is less than or equal to $kn$.

**Remark.** Note that the above proof holds if we replace 'nondeterministic' in the above result by 'deterministic', in which case the minimal deterministic FSA would be the one given by the Myhill-Nerode theorem.

Now, we have equality in the above result for the minimal FSA of any regular language, when viewed as a PFSA with just one membrane. But our aim is to keep the number of states at a minimum. It was proved in [3] that $(ab)^*$ is accepted by a PFSA with just one state but two membranes, and the PFSA constructed therein was deterministic. It is known that the minimal state deterministic FSA for $(ab)^*$ requires two states. So, the first question is whether any regular language can be accepted by a PFSA having only one state. The following example answers this question in the negative.

**Example 3.2.** $(aaa)^*$ cannot be accepted by a PFSA with only one state, even by those that are not deterministic and admit $\lambda$-transitions.

**Proof.** Suppose that there is a PFSA with only one state that accepts $(aaa)^*$, and having the least number of membranes among all such PFSAcs. Remove the useless rules if any (i.e., if the removal of any rule does not change the language accepted, then remove that rule; repeat this process until there is no such rule left). Note first that no transition is possible with the target here; this is true because otherwise a loop will be possible, and we will be able to add arbitrarily many $a$'s to some accepted string and still get an accepted string (if it were a $\lambda$-transition it would be a useless rule). Consider the membranes of the PFSA from where there are transitions with higher membranes as target. Such membranes exist, because the transitions cannot be to the same membrane, and the membranes are finite in number. There must be some string in the language $(aaa)^*$ such that such a membrane $i$ can be reached by reading some prefix of it, otherwise this membrane would be useless and can be removed to get a smaller PFSA. Now, when such a string is parsed, at least one transition involving the membrane $i$ should take place on reading an $a$ and not a $\lambda$, since otherwise again this membrane would be useless (as it could be "identified" with its parent, i.e., by removing it, letting its parent take over as the parent of its children, and transferring all the transitions from it to its children, to its parent). Thus we have a loop from the parent of the membrane $i$ to itself, involving one or two $a$'s. We can thus insert one or two $a$'s into the string and still get a string that is accepted, a contradiction to the fact that the number of $a$'s in any string that is accepted has to be a multiple of 3. \[Q.E.D.\]

Therefore, it would be interesting to find out the smallest natural number $k$, if any, such that any regular language can be accepted by a deterministic or nondeterministic PFSA with less than or equal to $k$ states (and any number of membranes). The existence of such a $k$ for nondeterministic PFSA will follow from the next section (it turns out that $k$ is 2 or 3).
It is fairly obvious that whatever is simulated by membranes can also be simulated by states. But membranes are a less powerful feature than states, as from a membrane one can only stay there, go to its parent or go to one of its children. A membrane structure admits a natural tree representation, while the transition graph between various states can be arbitrary. Hence the answer to the above question would be lying shrouded in some graph theoretic property.

4 Minimising Nondeterministic PFSA

In this section we prove that any nondeterministic PFSA is equivalent in its power to one with only three states, provided $\lambda$-transitions are allowed (and there is no restriction on the number of membranes).

Theorem 4.1. Any regular language can be accepted by a nondeterministic PFSA with only three states.

Proof. We use the following fact: if a family of languages on an alphabet contains all singleton sets, the empty set and the set $\{\lambda\}$, and is closed under union, concatenation, and the $*$ operation, then the family contains all regular languages. We construct P automata of the desired form such that the initial as well as the only final membrane is the outermost one, and the states, denoted $q_1$, $q_2$, and $q_3$, are such that the initial state is $q_1$ and the only final state is $q_3$. Let $R'$ be the family of languages accepted by all such automata.

First, it is fairly clear that all singleton sets, the empty set, and $\{\lambda\}$ belong to $R'$—we can rather easily give simple constructions for this purpose. It remains to prove the closure under the three operations given above.

- Closure under union: Suppose that $M_1$ and $M_2$ are PFSA accepting languages $L_1$ and $L_2$, respectively. Then one can consider a PFSA $M'$ with the outermost membrane containing the (disjoint) membrane structures of $M_1$ and $M_2$ inside, enabling us to think of $M_1$ and $M_2$ as parts of $M'$. While in state $q_1$ in the outermost membrane, a $\lambda$ can be read to go to the outermost membrane of $M_1$, so as to be in state $q_1$. Similarly, a $\lambda$ transition from $q_3$ in the outermost membrane of $M_1$ leads to the outermost membrane of $M'$, in the same state $q_3$. Exactly the same can be done for $M_2$. This construction can be pictorially represented as in Figure 1 (a). Clearly $M'$ accepts $L_1 \cup L_2$.

- Closure under concatenation: Let $M_1$, $M_2$, $L_1$ and $L_2$ be as before. Then the construction of $M'$ can be given as in Figure 1 (b). It is clear that $M'$ accepts $L_1 \cdot L_2$.

- Closure under the $*$ operation: If a PFSA $M$ accepts a language $L$, then $L^*$ is accepted by a PFSA $M'$ obtained from $M$ by allowing a $\lambda$-transition from $q_3$ to $q_1$ in the outermost membrane.
Thus, we have proved that the class $K$, in addition to containing the empty set, the set $\{\lambda\}$ and all singleton sets, is closed under the three operations listed above, so that it contains all regular languages.

\[ \square \]

5 Conclusion

We saw that a trade-off holds between states and membranes in P finite state automata. However, the concept of membranes is more natural in the context of P systems. The example in Section 3 showed that there are languages which cannot be accepted by PFSA having only one state, even if $\lambda$-transitions are allowed. In Section 4 it was seen that all regular languages can be accepted by PFSA with only three states, provided $\lambda$-transitions are allowed, and there is no restriction on the number of membranes. What happens if only two states are allowed? Is there a finite $n$ such that any regular language can be accepted by a PFSA with at most $n$ states, not necessarily deterministic, but such that $\lambda$-transitions are not allowed? This question can be raised for deterministic PFSA as well. Similar questions can be raised for P push-down automata. Thus, a number of interesting problems remain open, and it seems that the answers to some of them lie in graph theory.

Acknowledgements

The authors are grateful to the anonymous referee for the valuable comments.

References


Challenges in Business Process Management:
Verification of business processes using Petri nets

W.M.P. van der Aalst

Department of Technology Management, Eindhoven University of Technology
P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands.
w.m.p.v.d.aalst@tue.nl

Abstract. Most scientists working on formal methods are mainly focusing on technical systems such as circuit design, embedded systems, traffic control, etc. Few are working on the application of formal methods to business processes. As a result, interesting problems in the domain of Business Process Management (BPM) are not addressed. To stimulate the application of formal methods to BPM, the following two conferences, taking place in June 2003, are co-located: (1) the International Conference on Applications and Theory of Petri nets (Petri nets 2003) [8] and (2) the International Conference on Business Process Management: On the Application of Formal Methods to Process-Aware Information Systems (BPM 2003) [13]. Both conferences precede the International Colloquium on Automata, Languages and Programming (ICALP 2003) also taking place in Eindhoven (The Netherlands). By co-locating these events we hope to trigger cooperation between people working on BPM and formal methods. This survey/tutorial discusses the need for formal methods in BPM. Although different formal methods could be applied in this domain, we focus on the application of Petri nets in this domain. In particular, we focus on the verification of workflow processes using Petri-net-based results. By this we hope to stimulate scientists working on Petri nets to address some of the challenges posed by BPM.

1 Introduction

The goal of this paper is to discuss the relation between Petri nets and BPM. This way we hope to interest researchers working on formal methods in some of the scientific challenges in this domain. The definition of a BPM system used throughout this paper is: a generic software system that is driven by explicit process designs to enact and manage operational business processes. The system should be process-aware and generic in the sense that it is possible to modify the processes it supports. The process designs are often graphical and the focus is on structured processes that need to handle many cases.

In the remainder of this paper, we will first put BPM and related technology in its historical context. Then, we will discuss models for process design. Since BPM systems are driven by explicit models, it is important to use the right techniques. Next, we will discuss techniques for the analysis of process models. We will argue that it is vital to have techniques to assert the correctness of workflow designs. Based on this we introduce the class of workflow nets: A subclass of Petri nets.
2 Business process management from a historical perspective

*Only the wisest and stupidest of men never change.*

Confucius

To show the relevance of BPM systems, it is interesting to put them in a historical perspective. Consider Figure 1, which shows some of the ongoing trends in information systems. This figure shows that today's information systems consist of a number of layers. The center is formed by the operating system, i.e., the software that makes the hardware work. The second layer consists of generic applications that can be used in a wide range of enterprises. Moreover, these applications are typically used within multiple departments within the same enterprise. Examples of such generic applications are a database management system, a text editor, and a spreadsheet program. The third layer consists of domain specific applications. These applications are only used within specific types of enterprises and departments. Examples are decision support systems for vehicle routing, call center software, and human resource management software. The fourth layer consists of tailor-made applications. These applications are developed for specific organizations.

![Trends in information systems](image)

1. From programming to assembling.
2. From data orientation to process orientation.
3. From design to redesign and organic growth.

*Fig. 1. Trends relevant for BPM.*

In the sixties the second and third layer were missing. Information systems were built on top of a small operating system with limited functionality. Since no generic nor domain specific software was available, these systems mainly consisted of tailor-made applications. Since then, the second and third layer have developed and the ongoing trend is that the four circles are increasing in size, i.e., they are moving to the outside while absorbing new functionality. Today's operating systems offer much more...
functionality. Database management systems that reside in the second layer offer functionality which used to be in tailor-made applications. As a result of this trend, the emphasis shifted from programming to assembling of complex software systems. The challenge no longer is the coding of individual modules but orchestrating and gluing together pieces of software from each of the four layers.

Another trend is the shift from data to processes. The seventies and eighties were dominated by data-driven approaches. The focus of information technology was on storing and retrieving information and as a result data modeling was the starting point for building an information system. The modeling of business processes was often neglected and processes had to adapt to information technology. Management trends such as business process reengineering illustrate the increased emphasis on processes. As a result, system engineers are resorting to a more process driven approach.

The last trend we would like to mention is the shift from carefully planned designs to redesign and organic growth. Due to the omnipresence of the Internet and its standards, information systems change on-the-fly. As a result, fewer systems are built from scratch. In many cases existing applications are partly used in the new system. Although component-based software development still has it problems, the goal is clear and it is easy to see that software development has become more dynamic.

The trends shown in Figure 1 provide a historical context for BPM systems. BPM systems are either separate applications residing in the second layer or are integrated components in the domain specific applications, i.e., the third layer. Notable examples of BPM systems residing in the second layer are workflow management systems [32, 36] such as Staffware, MQSeries, and COSA, and case handling systems such as FLOWer. Note that leading enterprise resource planning systems populating the third layer also offer a workflow management module. The workflow engines of SAP, Baan, PeopleSoft, Oracle, and JD Edwards can be considered as integrated BPM systems. The idea to isolate the management of business processes in a separate component is consistent with the three trends identified. BPM systems can be used to avoid hard-coding the work processes into tailor-made applications and thus support the shift from programming to assembling. Moreover, process orientation, redesign, and organic growth are supported. For example, today’s workflow management systems can be used to integrate existing applications and support process change by merely changing the workflow diagram. Give these observations, we hope to have demonstrated the practical relevance of BPM systems. In the remainder of this paper we will focus more on the scientific importance of these systems. Moreover, for clarity we will often restrict the discussion to clear cut BPM systems such as workflow management systems.

An interesting starting point from a scientific perspective is the early work on office information systems. In the seventies, people like Skip Ellis [22], Anatol Holt [31], and Michael Zisman [44] already worked on so-called office information systems, which were driven by explicit process models. It is interesting to see that the three pioneers in this area independently used Petri-net variants to model office procedures. During the seventies and eighties there was great optimism about the applicability of office information systems. Unfortunately, few applications succeeded. As a result of these experiences, both the application of this technology and research almost stopped for a decade. Consequently, hardly any advances were made in the eighties. In the nineties,
there again was a huge interest in these systems. The number of workflow management systems developed in the past decade and the many papers on workflow technology illustrate the revival of office information systems. Today workflow management systems are readily available [36]. However, their application is still limited to specific industries such as banking and insurance. As was indicated by Skip Ellis it is important to learn from these ups and downs [23]. The failures in the eighties can be explained by both technical and conceptual problems. In the eighties, networks were slow or not present at all, there were no suitable graphical interfaces, and proper development software was missing. However, there were also more fundamental problems: a unified way of modeling processes was missing and the systems were too rigid to be used by people in the workplace. Most of the technical problems have been resolved by now. However, the more conceptual problems remain. Good standards for business process modeling are still missing and even today’s workflow management systems enforce unnecessary constrains on the process logic (e.g., processes are made more sequential).

To summarize we state that, although the relevance of BPM systems is undisputed, many fundamental problems remain to be solved. In the remainder of this paper we will try to shed light on some of these problems.

3 Models for process design

A camel is a horse designed by committee.
Sir Alec Issigonis

BPM systems are driven by models of processes and organizations. By changing these models, the behavior of the system adapts to its environment and changing requirements. These models cover different perspectives. Figure 2 shows some of the perspectives relevant for BPM systems [32]. The process perspective describes the control-flow, i.e., the ordering of tasks. The information perspective describes the data that are used. The resource perspective describes the structure of the organization and identifies resources, roles, and groups. The task perspective describes the content of individual steps in the processes. Each perspective is relevant. However, in this paper we restrict ourselves to the process perspective.

Many techniques have been proposed to model the process perspective. Some of these techniques are informal in the sense that the diagrams used have no formally defined semantics. These models are typically very intuitive and the interpretation shifts depending on the modeler, application domain, and characteristics of the business processes at hand. Examples of informal techniques are ISAC, DFD, SADT, and IDEF. These techniques may serve well for discussing work processes. However, they are inadequate for directly driving information systems since they are incomplete and subject to multiple interpretations. Therefore, more precise ways of modeling are required.

Figure 3 shows an example of an order handling process modeled in terms of a so-called workflow net [2]. Workflow nets are based on the classical Petri-net model invented by Carl Adam Petri in the early sixties [37]. The squares are the active parts of the model and correspond to tasks. The circles are the passive parts of the model and are used to represent states. In the classical Petri net, the squares are named transitions and
the circles places. A workflow net models the life-cycle of one case. Examples of cases are insurance claims, tax declarations, and traffic violations. Cases are represented by tokens and in this case the token in start corresponds to an order. Task register is a so-called AND-split and is enabled in the state shown. The arrow indicates that this task requires human intervention. If a person executes this task, the token is removed from place start and two tokens are produced: one for c1 and one for c2. Then, in parallel, two tasks are enabled: check availability and send bill. Depending on the eagerness of the workers executing these two tasks either check availability or send bill is executed first. Suppose check availability is executed first. If the ordered goods are available, they can be shipped by executing task ship goods. If they are not available, either a replenishment order is issued or not. Note that check availability is an OR-split and produces one token for c3, c4, or c5. Suppose that not all ordered goods are available, but the appropriate replenishment orders were already issued. A token is produced for

Fig. 2. Perspectives of models driving BPM systems.

Fig. 3. WF-net.
c3 and task update becomes enabled. Suppose that at this point in time task send_bill is executed, resulting in the state with a token in c3 and c6. The token in c6 is input for two tasks. However, only one of these tasks can be executed and in this state only receive_payment is enabled. Task receive_payment can be executed the moment the payment is received. Task reminder is an AND-join/AND-split and is blocked until the bill is sent and the goods have been shipped. Note that the reminder is sent after a specified period as indicated by the clock symbol. However, it is only possible to send a reminder if the goods have been actually shipped. Assume that in the state with a token in c3 and c6 task update is executed. This task does not require human involvement and is triggered by a message of the warehouse indicating that relevant goods have arrived. Again check_availability is enabled. Suppose that this task is executed and the result is positive. In the resulting state ship_goods can be executed. Now there is a token in c6 and c7 thus enabling task reminder. Executing task reminder again enables the task send_bill. A new copy of the bill is sent with the appropriate text. It is possible to send several reminders by alternating reminder and send_bill. However, let us assume that after the first loop the customer pays resulting in a state with a token in c7 and c8. In this state, the AND-join archive is enabled and executing this task results in the final state with a token in end.

This very simple workflow net shows some of the routing constructs relevant for business process modeling. Sequential, parallel, conditional, and iterative routing are present in this model. There also are more advanced constructs such as the choice between receive_payment and reminder. This is a so-called implicit choice since it is not resolved by the system but by the environment of the system. The moment the bill is sent, it is undetermined whether receive_payment or reminder will be the next step in the process. Another advanced construct is the fact that task reminder is blocked until the goods have been shipped. The latter construct is a so-called milestone. The reason that we point out both constructs is that many systems have problems supporting these rather fundamental process patterns [11, 12].

Workflow nets have clear semantics. The fact that we are able to play the so-called token game using a minimal set of rules shows the fact that these models are executable. None of the informal informal techniques mentioned before (i.e., ISAC, DFD, SADT, and IDEF) have formal semantics. Besides workflow nets there are many other formal techniques. Examples are the many variants of process algebra [14] and statecharts [29]. The reason we prefer to use a variant of Petri nets is threefold [2]:

- Petri nets are graphical and yet precise.
- Petri nets offer an abundance of analysis techniques.
- Petri nets treat states as first-class citizens.

The latter point deserves some more explanation. Many techniques for business process modeling focus exclusively on the active parts of the process, i.e., the tasks. This is rather surprising since in many administrative processes the actual processing time is measured in minutes and the flow time is measured in days. This means that most of the time cases are in-between two subsequent tasks. Therefore, it is vital to model these states explicitly.
4 Techniques for process analysis

From the errors of others, a wise man corrects his own.

Syrus

BPM systems allow organizations to change their processes by merely changing the models. The models are typically graphical and can be changed quite easily. This provides more flexibility than conventional information systems. However, by reducing the threshold for change, errors are introduced more easily. Therefore, it is important to develop suitable analysis techniques. However, it is not sufficient to just develop these techniques. It is as least as important to look at methods and tools to make them applicable in a practical context.

Traditionally, most techniques used for the analysis of business processes, originate from operations research. All students taking courses in operations management will learn to apply techniques such as simulation, queueing theory, and Markovian analysis. The focus mainly is on performance analysis and less attention is paid to the correctness of models. Verification and validation are often neglected. As a result, systems fail by not providing the right support or even break down [3,40]. Verification is needed to check whether the resulting system is free of logical errors. Many process designs suffer from deadlocks and livelocks that could have been detected using verification techniques. Validation is needed to check whether the system actually behaves as expected. Note that validation is context dependent while verification is not. A system that deadlocks is not correct in any situation. Therefore, verifying whether a system exhibits deadlocks is context independent. Validation is context dependent and can only be done with knowledge of the intended business process.

Fig. 4. An incorrect WF-net.
To illustrate the relevance of validation and verification and to demonstrate some of the techniques available, we return to the workflow net shown in Figure 3. This workflow process allows for the situation where a replenishment is issued before any payment is received. Suppose that we want to change the design such that replenishments are delayed until receiving payment. An obvious way to model this is to connect task `receive_payment` with `replenish` using an additional place `c9` as shown in Figure 4. Although this extension seems to be correct at first glance, the resulting workflow net has several errors. The workflow will deadlock if a second replenishment is needed and something is left behind in the process if no replenishments are needed. These are logical errors that can be detected without any knowledge of the order handling process. For verification, application independent notions of correctness are needed. One of these notions is the so-called soundness property [2]. A workflow net is sound if and only if the workflow contains no dead parts (i.e., tasks that can never be executed), from any reachable state it is always possible to terminate, and the moment the workflow terminates all places except the sink place (i.e., place `end`) are empty. Note that soundness rules out logical errors such as deadlocks and livelocks. The notion of soundness is applicable to any workflow language. An interesting observation is that soundness corresponds to liveness and boundedness of the short-circuited net [2]. The latter properties have been studied extensively [39, 21]. As a result, powerful analysis techniques and tools can be applied to verify the correctness of a workflow design. Practical experience shows that many errors can be detected by verifying the soundness property. Moreover, Petri-net theory can also be applied to guide the designer towards the error.

![Workflow Net](image)

Fig. 5. A sound but incorrect WF-net.

Soundness does not guarantee that the workflow net behaves as intended. Consider for example, the workflow net shown in Figure 5. Compared to the original model, the shipment of goods is skipped if some of the goods are not available. Again this may seem to be a good idea at first glance. However, customers are expected to pay even if the goods are never delivered. In other words, task `receive_payment` needs to
be executed although task \textit{ship\_goods} may never be executed. The latter error can only be detected using knowledge about the context. Based on this context one may decide whether this is acceptable or not. Few analysis techniques exist to automatically support this kind of validation. The only means of validation offered by today's workflow management systems is gaming and simulation.

An interesting technique to support validation is inheritance of dynamic behavior. Inheritance can be used as a technique to compare processes. Inheritance relates subclasses with superclasses [16]. A workflow net is a subclass of a superclass workflow net if certain dynamic properties are preserved. A subclass typically contains more tasks. If by hiding and/or blocking tasks in the subclass one obtains the superclass, the subclass inherits the dynamics of the superclass. The superclass can be used to specify the minimal properties the workflow design should satisfy. By merely checking whether the actual design is a subclass of the superclass, one can validate the essential properties. Consider for example Figure 6. This workflow net describes the minimal requirements the order handling process should satisfy. The tasks \textit{register}, \textit{ship\_goods}, \textit{receive\_payment}, and \textit{archive} are mandatory. Tasks \textit{ship\_goods} and \textit{receive\_payment} may be executed in parallel but should be preceded by \textit{register} and followed by \textit{archive}. The original order handling process shown in Figure 3 is a subclass of this superclass. Therefore, the minimal requirements are satisfied. However, the order handling process shown in Figure 5 is not a subclass. The fact that task \textit{ship\_goods} can be skipped demonstrates that not all properties are preserved.

\begin{figure}[h]
\centering
\includegraphics[width=0.6\textwidth]{fig6.png}
\caption{A superclass WF-net.}
\end{figure}

Inheritance of dynamic behavior is a very powerful concept that has many applications. Inheritance-preserving transformation rules and transfer rules offer support at design-time and at run-time [7]. Subclass-superclass relationships also can be used to enforce correct processes in an E-commerce setting. If business partners only execute subclass processes of some common contract process, then the overall workflow will be executed as agreed. It should be noted that workflows crossing the borders of organizations are particularly challenging from a verification and validation point of view [4]. Errors resulting from miscommunication between business partners are highly dis-

\footnote{We have identified four notions of inheritance [7, 16]. In this paper, we only refer to life-cycle inheritance.}
ruptive and costly. Therefore, it is important to develop techniques and tools for the verification and validation of these processes.

Few tools aiming at the verification of workflow processes exist. Woflan [42] and Flowmake [40] are two notable exceptions. We have been working on Woflan since 1997. Figure 7 shows a screenshot of Woflan. Woflan combines state-of-the-art scientific results with practical applications [10,42,43]. Woflan can interface with leading workflow management systems such as Staffware and COSA. It can also interface with BPR-tools such as Protos. Workflow processes designed using any of these tools can be verified for correctness. It turns out that the challenge is not to decide whether the design is sound or not. The real challenge is to provide diagnostic information that guides the designer to the error. Woflan also supports the inheritance notions mentioned before. Given two workflow designs, Woflan is able to decide whether one is a subclass of the other. Tools such as Woflan illustrate the benefits of a more fundamental approach. Large scale experiments with experienced students show that workflow designers frequently make errors and that these design errors can be detected using Woflan [42].
5 Formalization of sound workflow nets

The most likely way for the world to be destroyed, most experts agree, is by accident. That's where we come in; we're computer professionals. We cause accidents.

Nathaniel Borenstein

In the first part of this paper, an informal introduction was given into the BPM domain. In this introduction, we focused on workflow processes. As demonstrated, the process perspective can be modeled in terms of a WF-net. In this section, we formalize the notions mentioned in previous sections. First, we introduce some Petri net notation. Then we define WF-nets and the soundness property.

5.1 Petri Nets

This section introduces the basic Petri net terminology and notations. Readers familiar with Petri nets can skip this section.2

The classical Petri net is a directed bipartite graph with two node types called places and transitions. The nodes are connected via directed arcs. Connections between two nodes of the same type are not allowed. Places are represented by circles and transitions by rectangles.

Definition 1 (Petri net). A Petri net is a triple $(P, T, F)$:

- $P$ is a finite set of places,
- $T$ is a finite set of transitions ($P \cap T = \emptyset$),
- $F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs (flow relation)

A place $p$ is called an input place of a transition $t$ iff there exists a directed arc from $p$ to $t$. Place $p$ is called an output place of transition $t$ iff there exists a directed arc from $t$ to $p$. We use $t \bullet p$ to denote the set of input places for a transition $t$. The notations $t^\bullet$, $p^\bullet$ and $p^\bullet$ have similar meanings, e.g., $p^\bullet$ is the set of transitions sharing $p$ as an input place. Note that we do not consider multiple arcs from one node to another. In the context of workflow procedures it makes no sense to have other weights, because places correspond to conditions.

At any time a place contains zero or more tokens, drawn as black dots. The state, often referred to as marking, is the distribution of tokens over places, i.e., $M \in P \rightarrow \mathbb{N}$. We will represent a state as follows: $1p_1 + 2p_2 + 1p_3 + 0p_4$ is the state with one token in place $p_1$, two tokens in $p_2$, one token in $p_3$ and no tokens in $p_4$. We can also represent this state as follows: $p_1 + 2p_2 + p_3$. To compare states we define a partial ordering. For any two states $M_1$ and $M_2$, $M_1 \leq M_2$ iff for all $p \in P$: $M_1(p) \leq M_2(p)$

The number of tokens may change during the execution of the net. Transitions are the active components in a Petri net: they change the state of the net according to the following firing rule:

---

2 Note that states are represented by weighted sums and note the definition of (elementary) (conflict-free) paths.
A transition $t$ is said to be enabled iff each input place $p$ of $t$ contains at least one token.

An enabled transition may fire. If transition $t$ fires, then $t$ consumes one token from each input place $p$ of $t$ and produces one token for each output place $p$ of $t$.

Given a Petri net $(P, T, F)$ and a state $M_1$, we have the following notations:

- $M_1 \xrightarrow{t} M_2$: transition $t$ is enabled in state $M_1$ and firing $t$ in $M_1$ results in state $M_2$
- $M_1 \xrightarrow{\sigma} M_n$: the firing sequence $\sigma = t_1 t_2 t_3 \ldots t_{n-1}$ leads from state $M_1$ to state $M_n$ via a (possibly empty) set of intermediate states $M_2, \ldots, M_{n-1}$, i.e., $M_1 \xrightarrow{t_1} M_2 \xrightarrow{t_2} \ldots \xrightarrow{t_{n-1}} M_n$

A state $M_n$ is called reachable from $M_1$ (notation $M_1 \xrightarrow{\sigma} M_n$) iff there is a firing sequence $\sigma$ such that $M_1 \xrightarrow{\sigma} M_n$. Note that the empty firing sequence is also allowed, i.e., $M_1 \xrightarrow{} M_1$.

We use $(PN, M)$ to denote a Petri net $PN$ with an initial state $M$. A state $M'$ is a reachable state of $(PN, M)$ iff $M \xrightarrow{*} M'$.

Let us define some standard properties for Petri nets. First, we define properties related to the dynamics of a Petri net, then we give some structural properties.

**Definition 2 (Live).** A Petri net $(PN, M)$ is live iff, for every reachable state $M'$ and every transition $t$ there is a state $M''$ reachable from $M'$ which enables $t$.

A Petri net is structurally live if there exists an initial state such that the net is live.

**Definition 3 (Bounded, safe).** A Petri net $(PN, M)$ is bounded iff for each place $p$ there is a natural number $n$ such that for every reachable state the number of tokens in $p$ is less than $n$. The net is safe iff for each place the maximum number of tokens does not exceed 1.

A Petri net is structurally bounded if the net is bounded for any initially state.

**Definition 4 (Well-formed).** A Petri net $PN$ is well-formed iff there is a state $M$ such that $(PN, M)$ is live and bounded.

Paths connect nodes by a sequence of arcs.

**Definition 5 (Path, Elementary, Conflict-free).** Let $PN$ be a Petri net. A path $C$ from a node $n_i$ to a node $n_k$ is a sequence $(n_1, n_2, \ldots, n_k)$ such that $(n_i, n_{i+1}) \in F$ for $1 \leq i \leq k - 1$. $C$ is elementary iff, for any two nodes $n_i$ and $n_j$ on $C$, $i \neq j \Rightarrow n_i \neq n_j$. $C$ is conflict-free iff, for any place $n_j$ on $C$ and any transition $n_i$ on $C$, $j \neq i - 1 \Rightarrow n_j \notin \bullet n_i$.

For convenience, we introduce the alphabet operator $\alpha$ on paths. If $C = (n_1, n_2, \ldots, n_k)$, then $\alpha(C) = \{n_1, n_2, \ldots, n_k\}$.

**Definition 6 (Strongly connected).** A Petri net is strongly connected iff, for every pair of nodes (i.e., places and transitions) $x$ and $y$, there is a path leading from $x$ to $y$. 
Definition 7 (Free-choice). A Petri net is a free-choice Petri net iff, for every two transitions \( t_1 \) and \( t_2 \), \( \bullet t_1 \cap \bullet t_2 \neq \emptyset \) implies \( \bullet t_1 = \bullet t_2 \).

Definition 8 (State machine). A Petri net is state machine iff each transition has exactly one input and one output place.

Definition 9 (S-component). A subnet \( PN_s = (P_s, T_s, F_s) \) is called an S-component of a Petri net \( PN = (P, T, F) \) if \( P_s \subseteq P, T_s \subseteq T, F_s \subseteq F, PN_s \) is strongly connected, \( PN_s \) is a state machine, and for every \( q \in P_s \) and \( t \in T \): \((q, t) \in F \Rightarrow (q, t) \in F_s \) and \((t, q) \in F \Rightarrow (t, q) \in F_s \).

Definition 10 (S-coverable). A Petri net is S-coverable iff for any node there exist an S-component which contains this node.

See [21, 39] for a more elaborate introduction to these standard notions.

5.2 WF-Nets

A Petri net which models the control-flow dimension of a workflow, is called a Workflow net (WF-net). It should be noted that a WF-net specifies the dynamic behavior of a single case in isolation.

Definition 11 (WF-net). A Petri net \( PN = (P, T, F) \) is a WF-net (Workflow net) if

\( (i) \) There is one source place \( i \in P \) such that \( \bullet i = \emptyset \).
\( (ii) \) There is one sink place \( o \in P \) such that \( o \bullet = \emptyset \).
\( (iii) \) Every node \( x \in P \cup T \) is on a path from \( i \) to \( o \).

A WF-net has one input place (i) and one output place (o) because any case handled by the procedure represented by the WF-net is created when it enters the WFMS and is deleted once it is completely handled by the WFMS, i.e., the WF-net specifies the life-cycle of a case. The third requirement in Definition 11 has been added to avoid ‘dangling tasks and/or conditions’, i.e., tasks and conditions which do not contribute to the processing of cases.

Given the definition of a WF-net it is easy derive the following properties.

Proposition 1 (Properties of WF-nets). Let \( PN = (P, T, F) \) be Petri net.

\( - \) If \( PN \) is WF-net with source place \( i \), then for any place \( p \in P \): \( \bullet p \neq \emptyset \) or \( p = i \), i.e., \( i \) is the only source place.
\( - \) If \( PN \) is WF-net with sink place \( o \), then for any place \( p \in P \): \( p \bullet \neq \emptyset \) or \( p = o \), i.e., \( o \) is the only sink place.
\( - \) If \( PN \) is a WF-net and we add a transition \( t^* \) to \( PN \) which connects sink place \( o \) with source place \( i \) (i.e., \( \bullet t^* = \{o\} \) and \( t^* \bullet = \{i\} \)), then the resulting Petri net is strongly connected.
\( - \) If \( PN \) has a source place \( i \) and a sink place \( o \) and adding a transition \( t^* \) which connects sink place \( o \) with source place \( i \) yields a strongly connected net, then every node \( x \in P \cup T \) is on a path from \( i \) to \( o \) in \( PN \) and \( PN \) is a WF-net.
Figures 3, 4 and 5 show examples of WF-nets. In each net the source place \( i \) is named \textit{start} and the sink place \( o \) is named \textit{end}. Note that some syntactic sugaring is used. There is a one-to-one correspondence between AND-splits/AND-joins and transitions. However, OR-splits and OR-joins correspond to clusters of transitions: one for each choice. For example, task \textit{check availability} corresponds to three transitions. Each of these three transitions has \( c1 \) as input place and one output place. The first one produces a token for \( c5 \), the second for \( c4 \), and the third for \( c3 \). Using this translation it is easy to see that each of the nets shown in figures 3, 4 and 5 is indeed a WF-net.

5.3 Soundness

In this section we summarize some of the basic results for WF-nets presented in [1]. The remainder of this paper will build on these results.

The three requirements stated in Definition 11 can be verified statically, i.e., they only relate to the structure of the Petri net. However, there is another requirement which should be satisfied:

\begin{quote}
For any case, the procedure will terminate eventually and the moment the procedure terminates there is a token in place \( o \) and all the other places are empty.
\end{quote}

Moreover, there should be no dead tasks, i.e., it should be possible to execute an arbitrary task by following the appropriate route through the WF-net. These two additional requirements correspond to the so-called \textit{soundness property}.

\textbf{Definition 12 (Sound).} A procedure modeled by a WF-net \( PN = (P, T, F) \) is \textit{sound} if and only if:

(i) \textit{For every state} \( M \) \textit{reachable from state} \( i \), \textit{there exists a firing sequence leading from state} \( M \) \textit{to state} \( o \). Formally:\(^3\)

\[
\forall_M (i \rightarrow M) \Rightarrow \left(M \rightarrow o\right)
\]

(ii) \textit{State} \( o \) \textit{is the only state reachable from state} \( i \) \textit{with at least one token in place} \( o \). Formally:

\[
\forall_M (i \rightarrow M \land M \geq o) \Rightarrow (M = o)
\]

(iii) \textit{There are no dead transitions in} \( (PN, i) \). Formally:

\[
\forall_{t \in T} \exists_{M, M'} i \rightarrow M \xrightarrow{t} M'
\]

Note that the soundness property relates to the dynamics of a WF-net. The first requirement in Definition 12 states that starting from the initial state (state \( i \)), it is always possible to reach the state with one token in place \( o \) (state \( o \)). If we assume a strong notion of fairness, then the first requirement implies that eventually state \( o \) is reached. Strong fairness means in every infinite firing sequence, each transition fires infinitely often. The fairness assumption is reasonable in the context of workflow management:

\(^3\) Note that there is an overloading of notation: the symbol \( i \) is used to denote both the \textit{place} \( i \) and the \textit{state} with only one token in place \( i \) (see Section 5.1).
All choices are made (implicitly or explicitly) by applications, humans or external actors. Clearly, they should not introduce an infinite loop. Note that the traditional notions of fairness (i.e., weaker forms of fairness with just local conditions, e.g., if a transition is enabled infinitely often, it will fire eventually) are not sufficient. See [2, 34] for more details. The second requirement states that the moment a token is put in place \( o \), all the other places should be empty. Sometimes the term *proper termination* is used to describe the first two requirements [27]. The last requirement states that there are no dead transitions (tasks) in the initial state \( i \).

Note that the second requirement is implied by the first one. Suppose the second requirement does not hold. This implies that there is a state \( M \) reachable from \( i \) such that \( M \geq o \) and \( M \neq o \), i.e., a state with at least two tokens. There is at least one token in \( o \) which cannot be removed. The other token cannot be consumed without producing a new one. Therefore, it is not possible to reach state \( o \) from \( M \). This contradicts with the first requirement. This shows that the second requirement can be removed. Nevertheless, Definition 12 lists this requirement since it corresponds to an intuitive notion of correctness.

![Fig. 8. Another WF-net for the processing of complaints.](image)

Figure 8 shows a WF-net which is not sound. There are several deficiencies. If \( \text{time\_out\_1} \) and \( \text{processing\_2} \) fire or \( \text{time\_out\_2} \) and \( \text{processing\_1} \) fire, the WF-net will not terminate properly because a token gets stuck in \( c4 \) or \( c5 \). If \( \text{time\_out\_1} \) and \( \text{time\_out\_2} \) fire, then the task \( \text{processing\_NOK} \) will be executed twice and because of the presence of two tokens in \( o \) the moment of termination is not clear.

Figures 3, 4 and 5 also show examples of WF-nets. In each net the source place \( i \) is named *start* and the sink place \( o \) is named *end* and task *check\_availability* corresponds to multiple transitions as described before. The WF-net shown in Figure 3 is sound. The WF-net shown in Figure 4 is not sound: The process may deadlock before reaching the sink place or the process may leave a superfluous token in place \( c9 \). The WF-net shown in Figure 5 is sound (although it is not a subclass of Figure 3).

Given a WF-net \( PN = (P,T,F) \), we want to decide whether \( PN \) is sound. In [1] we have shown that soundness corresponds to liveness and boundedness. To link soundness to liveness and boundedness, we define an extended net \( \overline{PN} = (\overline{P},\overline{T},\overline{F}) \).
$\overline{PN}$ is the Petri net obtained by adding an extra transition $t^*$ which connects $o$ and $i$. The extended Petri net $\overline{PN} = (\overline{P}, \overline{T}, \overline{F})$ is defined as follows: $\overline{P} = P$, $\overline{T} = T \cup \{t^*\}$, and $\overline{F} = F \cup \{(o, t^*), (t^*, i)\}$. In the remainder we will call such an extended net the short-circuited net of $PN$. The short-circuited net allows for the formulation of the following theorem.

**Theorem 1.** A WF-net $PN$ is sound if and only if $\langle \overline{PN}, i \rangle$ is live and bounded.

**Proof.** See [1].

This theorem shows that standard Petri-net-based analysis techniques can be used to verify soundness. Several authors have proposed other (typically weaker) notions of soundness. In [19, 24] the notion of relaxed soundness is used. This notion is weaker since it focuses on the possibility to have sound executions. In [30] the notion of $k$-soundness is introduced. This notion takes $k$ tokens in place $i$ as a starting point. In [35] soundness is also defined for multiple connected WF-nets.

### 6 Structural Characterization of Soundness

*An classic is classic not because it conforms to certain structural rules, or fits certain definitions (of which its author had quite probably never heard). It is classic because of a certain eternal and irrepressible freshness.*

— Edith Wharton

Theorem 1 gives a useful characterization of the quality of a workflow process definition. However, there are a number of problems:

- For a complex WF-net it may be intractable to decide soundness. (For arbitrary WF-nets liveness and boundedness are decidable but also EXPSPACE-hard, cf. Cheng, Esparza and Palsberg [18].)
- Soundness is a minimal requirement. Readability and maintainability issues are not addressed by Theorem 1.
- Theorem 1 does not show how a non-sound WF-net should be modified, i.e., it does not identify constructs which invalidate the soundness property.

These problems stem from the fact that the definition of soundness relates to the dynamics of a WF-net while the workflow designer is concerned with the static structure of the WF-net. Therefore, it is interesting to investigate structural characterizations of sound WF-nets. For this purpose we introduce three interesting subclasses of WF-nets: free-choice WF-nets, well-structured WF-nets, and S-coverable WF-nets.

#### 6.1 Free-Choice WF-Nets

Most of the WFMS's available at the moment, abstract from states between tasks, i.e., states are not represented explicitly. These WFMS's use building blocks such as the AND-split, AND-join, OR-split and OR-join to specify workflow procedures. The
AND-split and the AND-join are used for parallel routing. The OR-split and the OR-join are used for conditional routing. Because these systems abstract from states, every choice is made *inside* an OR-split building block. If we model an OR-split in terms of a Petri net, the OR-split corresponds to a number of transitions sharing the same set of input places. This means that for these WFMS's, a workflow procedure corresponds to a free-choice Petri net (cf. Definition 7).

It is easy to see that a process definition composed of AND-splits, AND-joins, OR-splits and OR-joins is free-choice. If two transitions \( t_1 \) and \( t_2 \) share an input place \((\bullet t_1 \cap \bullet t_2 \neq \emptyset)\), then they are part of an OR-split, i.e., a 'free choice' between a number of alternatives. Therefore, the sets of input places of \( t_1 \) and \( t_2 \) should match \((\bullet t_1 = \bullet t_2)\). Figure 8 shows a free-choice WF-net. The WF-net shown in Figure 3 is not free-choice; *archive* and *reminder* share an input place but the two corresponding input sets differ. We have evaluated many WFMS's and just one of these systems (COSA [41]) allows for a construct which is comparable to a non-free choice WF-net [12, 33]. Therefore, it makes sense to consider free-choice Petri nets in more detail. Clearly, parallelism, sequential routing, conditional routing and iteration can be modeled without violating the free-choice property. Another reason for restricting WF-nets to free-choice Petri nets is the following. If we allow non-free-choice Petri nets, then the choice between conflicting tasks may be influenced by the order in which the preceding tasks are executed. The routing of a case should be independent of the order in which tasks are executed. A situation where the free-choice property is violated is often a mixture of parallelism and choice. Figure 9 shows such a situation. Firing transition \( t_1 \) introduces parallelism. Although there is no real choice between \( t_2 \) and \( t_5 \) (\( t_5 \) is not enabled), the parallel execution of \( t_2 \) and \( t_3 \) results in a situation where \( t_5 \) is not allowed to occur. However, if the execution of \( t_2 \) is delayed until \( t_3 \) has been executed, then there is a real choice between \( t_2 \) and \( t_5 \). In our opinion parallelism itself should be separated from the choice between two or more alternatives. Therefore, we consider the non-free-choice construct shown in Figure 9 to be improper. In literature, the term *confusion* is often used to refer to the situation shown in Figure 9.

![Fig. 9. A non-free-choice WF-net containing a mixture of parallelism and choice.](image-url)

Free-choice Petri nets have been studied extensively (cf. Best [17], Desel and Esparza [21, 20, 25], Hack [28]) because they seem to be a good compromise between expressive power and analyzability. It is a class of Petri nets for which strong theoretical results and efficient analysis techniques exist. For example, the well-known Rank Theorem (Desel and Esparza [21]) enables us to formulate the following corollary.
Corollary 1. The following problem can be solved in polynomial time.
Given a free-choice WF-net, to decide if it is sound.

Proof. Let PN be a free-choice WF-net. The short-circuited net $\overline{PN}$ is also free-choice. Therefore, the problem of deciding whether $(\overline{PN}, i)$ is live and bounded can be solved in polynomial time (Rank Theorem [21]). By Theorem 1, this corresponds to soundness.

Corollary 1 shows that, for free-choice nets, there are efficient algorithms to decide soundness. Moreover, a sound free-choice WF-net is guaranteed to be safe (given an initial state with just one token in i).

Lemma 1. A sound free-choice WF-net is safe.

Proof. Let PN be a sound free-choice WF-net. $\overline{PN}$ is the Petri net PN extended with a transition connecting o and i. $\overline{PN}$ is free-choice and well-formed. Hence, $\overline{PN}$ is $S$-coverable [21], i.e., each place is part of an embedded strongly connected state-machine component. Since initially there is just one token $(\overline{PN}, i)$ is safe and so is $(PN, i)$. □

Safeness is a desirable property, because it makes no sense to have multiple tokens in a place representing a condition. A condition is either true (1 token) or false (no tokens).

Although most WFMS's only allow for free-choice workflows, free-choice WF-nets are not a completely satisfactory structural characterization of 'good' workflows. On the one hand, there are non-free-choice WF-nets which correspond to sensible workflows (cf. Figure 3). On the other hand there are sound free-choice WF-nets which make no sense. Nevertheless, the free-choice property is a desirable property. If a workflow can be modeled as a free-choice WF-net, one should do so. A workflow specification based on a free-choice WF-net can be enacted by most workflow systems. Moreover, a free-choice WF-net allows for efficient analysis techniques and is easier to understand. Non-free-choice constructs such as the construct shown in Figure 9 are a potential source of anomalous behavior (e.g., deadlock) which is difficult to trace.

6.2 Well-Structured WF-Nets

Another approach to obtain a structural characterization of 'good' workflows, is to balance AND/OR-splits and AND/OR-joins. Clearly, two parallel flows initiated by an AND-split, should not be joined by an OR-join. Two alternative flows created via an OR-split, should not be synchronized by an AND-join. As shown in Figure 10, an AND-split should be complemented by an AND-join and an OR-split should be complemented by an OR-join.

One of the deficiencies of the WF-net shown in Figure 8 is the fact that the AND-split register is complemented by the OR-join c3 or the OR-join o. To formalize the concept illustrated in Figure 10 we give the following definition.

Definition 13 (Well-handled). A Petri net PN is well-handled iff, for any pair of nodes x and y such that one of the nodes is a place and the other a transition and for any pair of elementary paths $C_1$ and $C_2$ leading from x to y, $\alpha(C_1) \cap \alpha(C_2) = \{x, y\} \Rightarrow C_1 = C_2$. 
Note that the WF-net shown in Figure 8 is not well-handled. Well-handledness can be decided in polynomial time by applying a modified version of the max-flow min-cut technique described in [9]. A Petri net which is well-handled has a number of nice properties, e.g., strong connectedness and well-formedness coincide.

**Lemma 2.** A strongly connected well-handled Petri net is well-formed.

**Proof.** Let $PN$ be a strongly connected well-handled Petri net. Clearly, there are no circuits that have PT-handles nor TP-handles [26]. Therefore, the net is structurally bounded (See Theorem 3.1 in [26]) and structurally live (See Theorem 3.2 in [26]). Hence, $PN$ is well-formed. $\square$

Clearly, well-handledness is a desirable property for any WF-net $PN$. Moreover, we also require the short-circuited $\overline{PN}$ to be well-handled. We impose this additional requirement for the following reason. Suppose we want to use $PN$ as a part of a larger WF-net $PN'$. $PN'$ is the original WF-net extended with an ‘undo-task’. See Figure 11. Transition undo corresponds to the undo-task, transitions $t1$ and $t2$ have been added to make $PN'$ a WF-net. It is undesirable that transition undo violates the well-handledness property of the original net. However, $PN'$ is well-handled iff $\overline{PN}$ is well-handled. Therefore, we require $\overline{PN}$ to be well-handled. We use the term well-structured to refer to WF-nets whose extension is well-handled.

**Fig. 11.** The WF-net $PN'$ is well-handled iff $\overline{PN}$ is well-handled.
Definition 14 (Well-structured). A WF-net $PN$ is well-structured iff $\overline{PN}$ is well-handled.

Well-structured WF-nets have a number of desirable properties. Soundness can be verified in polynomial time and a sound well-structured WF-net is safe. To prove these properties we use some of the results obtained for elementary extended non-self controlling nets.

Definition 15 (Elementary extended non-self controlling). A Petri net $PN$ is elementary extended non-self controlling (ENSC) iff, for every pair of transitions $t_1$ and $t_2$ such that $\bullet t_1 \cap \bullet t_2 \neq \emptyset$, there does not exist an elementary path $C$ leading from $t_1$ to $t_2$ such that $\bullet t_1 \cap \alpha(C) = \emptyset$.

Theorem 2. Let $PN$ be a WF-net. If $PN$ is well-structured, then $\overline{PN}$ is elementary extended non-self controlling.

Proof. Assume that $\overline{PN}$ is not elementary extended non-self controlling. This means that there is a pair of transitions $t_1$ and $t_k$ such that $\bullet t_1 \cap \bullet t_k \neq \emptyset$ and there exist an elementary path $C = (t_1, p_2, t_2, \ldots, p_k, t_k)$ leading from $t_1$ to $t_k$ and $\bullet t_1 \cap \alpha(C) = \emptyset$. Let $p_1 \in \bullet t_1 \cap \bullet t_k$. $C_1 = (p_1, t_k)$ and $C_2 = (p_1, t_1, p_2, t_2, \ldots, p_k, t_k)$ are paths leading from $p_1$ to $t_k$. (Note that $C_2$ is the concatenation of $(p_1)$ and $C$.) Clearly, $C_1$ is elementary. We will also show that $C_2$ is elementary. $C$ is elementary, and $p_1 \notin \alpha(C)$ because $p_1 \in \bullet t_1$. Hence, $C_2$ is also elementary. Since $C_1$ and $C_2$ are both elementary paths, $C_1 \neq C_2$ and $\alpha(C_1) \cap \alpha(C_2) = \{p_1, t_k\}$, we conclude that $PN$ is not well-handled. \qed

Consider for example the WF-net shown in Figure 12. The WF-net is well-structured and, therefore, also elementary extended non-self controlling. However, the net is not free-choice. Nevertheless, it is possible to verify soundness for such a WF-net very efficiently.

Corollary 2. The following problem can be solved in polynomial time. Given a well-structured WF-net, to decide if it is sound.
Let $PN$ be a well-structured WF-net. The short-circuited net $\overline{PN}$ is elementary extended non-self controlling (Theorem 2) and structurally bounded (see proof of Lemma 2). For bounded elementary extended non-self controlling nets the problem of deciding whether a given marking is live, can be solved in polynomial time (See [15]). Therefore, the problem of deciding whether $(\overline{PN},i)$ is live and bounded can be solved in polynomial time. By Theorem 1, this corresponds to soundness.

**Lemma 3.** A sound well-structured WF-net is safe.

**Proof.** Let $PN$ be the net $PN$ extended with a transition connecting $o$ and $i$. $\overline{PN}$ is extended non-self controlling. $\overline{PN}$ is covered by state-machines (S-components), see Corollary 5.3 in [15]. Hence, $\overline{PN}$ is safe and so is $PN$ (see proof of Lemma 1).

Well-structured WF-nets and free-choice WF-nets have similar properties. In both cases soundness can be verified very efficiently and soundness implies safeness. In spite of these similarities, there are sound well-structured WF-nets which are not free-choice (Figure 12) and there are sound free-choice WF-nets which are not well-structured. In fact, it is possible to have a sound WF-net which is neither free-choice nor well-structured (Figures 3 and 9).

### 6.3 S-Coverable WF-Nets

What about the sound WF-nets shown in Figure 3 and Figure 9? The WF-net shown in Figure 9 can be transformed into a free-choice well-structured WF-net by separating choice and parallelism. The WF-net shown in Figure 3 cannot be transformed into a free-choice or well-structured WF-net without yielding a much more complex WF-net. Place $c7$ acts as some kind of milestone which is tested by the task reminder. Traditional workflow management systems which do not make the state of the case explicit are not able to handle the workflow specified by Figure 3. Only workflow management systems such as COSA [41] have the capability to enact such a state-based workflow. Nevertheless, it is interesting to consider generalizations of free-choice and well-structured WF-nets: *S-coverable WF-nets* can be seen as such a generalization.

**Definition 16 (S-coverable).** A WF-net $PN$ is S-coverable if the short-circuited net $\overline{PN}$ is S-coverable.

The WF-nets shown in Figure 3 and Figure 9 are S-coverable. The WF-net shown in Figure 8 is not S-coverable. The following two corollaries show that S-coverability is a generalization of the free-choice property and well-structuredness.

**Corollary 3.** A sound free-choice WF-net is S-coverable.

**Proof.** The short-circuited net $\overline{PN}$ is free-choice and well-formed. Hence, $\overline{PN}$ is S-coverable (cf. [21]).

**Corollary 4.** A sound well-structured WF-net is S-coverable.

**Proof.** $\overline{PN}$ is extended non-self controlling (Theorem 2). Hence, $\overline{PN}$ is S-coverable (cf. Corollary 5.3 in [15]).
All the sound WF-nets presented in this paper are S-coverable. Every S-coverable WF-net is safe. The only WF-net which is not sound, i.e., the WF-net shown in Figure 8, is not S-coverable. These and other examples indicate that there is a high correlation between S-coverability and soundness. It seems that S-coverability is one of the basic requirements any workflow process definition should satisfy. From a formal point of view, it is possible to construct WF-nets which are sound but not S-coverable. Typically, these nets contain places which do not restrict the firing of a transition, but which are not in any S-component. (See for example Figure 65 in [38].) From a practical point of view, these WF-nets are to be avoided. WF-nets which are not S-coverable are difficult to interpret because the structural and dynamical properties do not match. For example, these nets can be live and bounded but not structurally bounded. There seems to be no practical need for using constructs which violate the S-coverability property. Therefore, we consider S-coverability to be a basic requirement any WF-net should satisfy.

Another way of looking at S-coverability is the following interpretation: S-components corresponds to document flows. To handle a workflow several pieces of information are created, used, and updated. One can think of these pieces of information as physical documents, i.e., at any point in time the document is in one place in the WF-net. Naturally, the information in one document can be copied to another document while executing a task (i.e., transition) processing both documents. Initially, all documents are present but a document can be empty (i.e., corresponds to a blank piece paper). It is easy to see that the flow of one such document corresponds a state machine (assuming the existence of a transition $t^*$). These document flows synchronize via joint tasks. Therefore, the composition of these flows yields an S-coverable WF-net. One can think of the document flows as threads. Consider for example the short-circuited net of the WF-net shown in Figure 3. This net can be composed out of the following two threads: (1) a thread corresponding to the physical process (places $\text{start}$, $c1$, $c3$, $c4$, $c5$, $c7$ and $\text{end}$) and (2) a thread corresponding to the financial process (places $\text{start}$, $c2$, $c6$, $c8$, and $\text{end}$). Note that the tasks $\text{register}$, $\text{reminder}$ and $\text{archive}$ are used in both threads.

Although a WF-net can, in principle, have exponentially many S-components, they are quite easy to compute for workflows encountered in practice (see also the above interpretation of S-component as document flows or threads). Note that S-coverability only depends on the structure and the degree of connectedness is generally low (i.e., the incidence matrix of a WF-net typically has few non-zero entries [9]). Unfortunately, in general, it is not possible to verify soundness of an S-coverable WF-net in polynomial time. The problem of deciding soundness for an S-coverable WF-net is PSPACE-complete. For most applications this is not a real problem. In most cases the number of tasks in one workflow process definition is less than 100 and the number of states is less than 200,000. Tools using standard techniques such as the construction of the coverability graph have no problems in coping with these workflow process definitions.

### 6.4 Summary

The three structural characterizations (free-choice, well-structured and S-coverable) turn out to be very useful for the analysis of workflow process definitions. Based on
our experience, we have good reasons to believe that S-coverability is a desirable property any workflow definition should satisfy. Constructs violating S-coverability can be detected easily and tools can be build to help the designer to construct an S-coverable WF-net. S-coverability is a generalization of well-structuredness and the free-choice property (Corollary 3 and 4). Both well-structuredness and the free-choice property also correspond to desirable properties of a workflow. A WF-net satisfying at least one of these two properties can be analyzed very efficiently. However, we have shown that there are workflows that are not free-choice and not well-structured. Consider for example Figure 3. The fact that task reminder tests whether there is a token in c7, prevents the WF-net from being free-choice or well-structured. Although this is a very sensible workflow, most workflow management systems do not support such an advanced routing construct. Even if one is able to use state-based workflows (e.g., COSA) allowing for constructs which violate well-structuredness and the free-choice property, then the structural characterizations are still useful. If a WF-net is not free-choice or not well-structured, one should locate the source which violates one of these properties and check whether it is really necessary to use a non-free-choice or a non-well-structured construct. If the non-free-choice or non-well-structured construct is really necessary, then the correctness of the construct should be double-checked, because it is a potential source of errors. This way the readability and maintainability of a workflow process definition can be improved.

7 Conclusion

In this paper, the application of Petri nets to BPM was discussed. First, BPM was put in its historical perspective. Then, the topics of process design and process analysis were discussed. These topics have been illustrated using Petri nets. In the second part of the paper, we investigated a basic property that any workflow process definition should satisfy: the soundness property. For WF-nets, this property coincides with liveness and boundedness. In our quest for a structural characterization of WF-nets satisfying the soundness property, we have identified three important subclasses: free-choice, well-structured, and S-coverable WF-nets. The identification of these subclasses is useful for the detection of design errors. We hope that these results demonstrate the relevance of formal methods for BPM in general and workflow management in particular.

References


4 Note that parts of the paper are taken from [5, 6].


Threshold Phenomena: The Computer Scientist's Viewpoint

Yannis C. Stamatiou,
Department of Mathematics, University of the Aegean,
Karlovassi, 83200, Samos, Greece. e-mail: stamatiu@aegean.gr

Preface

In this survey, we attempt to present the main concepts and techniques involved in the study of a very interesting phenomenon in Theoretical Computer Science and Physics (Statistical Mechanics) alike: the emergence of a sudden change of behaviour, when certain conditions are met, in combinatorial objects (e.g. graphs, Boolean formulas) or physical materials (e.g. spin systems).

Let us consider, for example, Boolean formulas with three literals per clause, or 3-SAT formulas. Assume that we fix a number of Boolean variables, say $n$, and we start generating random 3-SAT formulas with $m = rn$ clauses, for various values of the constant $r$, and check whether they are satisfiable. It has been experimentally observed that as the ratio $r$ of the number $m$ of clauses to the number $n$ of variables is increased from a little bit before a value $\approx 4.2$ to a little bit beyond this value, then the probability of satisfiability of a random 3-SAT formula falls abruptly from nearly 1 to nearly 0. Moreover, as $n$ increases towards larger and larger values, the transition of the probability becomes more and more abrupt. It is widely believed that there exists a critical value $r^*$, called the satisfiability threshold, such that for any $\varepsilon > 0$ and sufficiently large $n$, the probability of satisfiability goes down from $1 - \varepsilon$ to $\varepsilon$ within an arbitrarily small neighbourhood of $r^*$. And this is the so-called Satisfiability Threshold Conjecture that will be the focus of the survey!

The structure of the survey is as follows. In Section 1 we give a brief introduction to the satisfiability problem, we state its computational complexity and we introduce the satisfiability threshold conjecture. Then we introduce the first moment method, which is the basic tool for proving upper bounds on the satisfiability threshold and we talk a little about threshold functions in general. In Section 2 we expose the reader to the basic technique for proving lower bounds: the design and analysis of randomized algorithms that attempt to satisfy a formula by assigning values to variables using a specific heuristic rule. In Section 3 we state and use more exact variants of the first moment method introduced in Section 1 and review the various techniques that have been proposed and used in order to rectify its major weakness, that is the inclusion of rare formulas with huge numbers of satisfying truth assignments. Then we

---

1This survey was written as a companion of the author's talk on upper bounds given at the Logic and Interaction seminar series that took place in Marseille from January 28th 2002 to March 1st 2002 and was organized jointly by the Institut de Mathématiques de Luminy (IML) and Laboratoire d'Informatique Fondamentale de Marseille (LIF). The author's work was partially supported by the Research and Academic Computer Technology Institute.
explain in detail the ideas of an approach called the method of locally maximum satisfying truth assignments that can be used to address the weakness mentioned above. Section 4 is an attempt (by, certainly, a non-expert!) to express some results and personal thoughts on how threshold phenomena in Mathematics can be linked to similar phenomena in Statistical Mechanics in Physics in which there is also an element of random interaction between physical particles that can be likened to randomly chosen clauses over Boolean variables. Finally, we conclude in Section 5 by expressing the thought that perhaps both phenomena, in the Mathematical and the Physical world, may really be two sides of the same coin, expressing some universal, and yet to be unveiled, complexity concept.

This survey is, by no means, intended to be an exhaustive review of all the results and techniques involved in the efforts to attack the satisfiability threshold conjecture. Its aim is to present to non-experts in this research area some basic probabilistic tools giving, also, detailed examples of their use for the approximation (from above and below) of the satisfiability threshold. At the same time, the survey also attempts to highlight the similarities between threshold phenomena in mathematics and threshold phenomena in physics and how it is possible to attack the former using techniques developed for the study of the latter. The interested reader may consult [37, 16] for comprehensive reviews concerning the improvements and techniques for deriving upper bounds on the satisfiability threshold as well as [1] for a detailed review of the improvements and relevant techniques for achieving lower bounds on the satisfiability threshold.

Finally, I would like to thank my friends Lefteris Kirousis and Alexis Kaporis for their help in preparing the first draft of this survey (in the form of lecture notes) by pointing out omissions and giving me suggestions for improvements.

1 SAT: The Drosophilla of Complexity Theory

Let us fix a set of Boolean variables, that is variables that may assume only one of two possible values: 0 ("false") and 1 ("true"). Call this set $V$. The Satisfiability Problem consists in deciding whether a Boolean formula given as a conjunction (that is logical AND) of disjunctions (that is logical OR) of variables in $V$ or their negation (literals), evaluates to true for some value assignment to the variables. For example consider the set of variables $V = \{x_1, x_2, x_3\}$ and the following formula ($\land$ = $\land$, $\lor$ = $\lor$):

$$f = (x_1 \lor x_2 \lor x_3) \land (\neg x_1 \lor \neg x_2 \lor \neg x_3).$$

This formula is written in Conjunctive Normal Form (CNF), as we say, and in order to be satisfiable all its constituents, also called clauses, must be satisfiable. It is easy to see that by assigning say $x_1 = 1$ and $x_2 = x_3 = 0$ the formula is made satisfiable. But let us now consider what happens if we add some more clauses:

$$f' = (x_1 \lor x_2 \lor x_3) \land (\neg x_1 \lor \neg x_2 \lor \neg x_3) \land (x_1 \lor \neg x_2) \land (x_2 \lor \neg x_3) \land (x_3 \lor \neg x_1).$$

Let us take the first two clauses. These two clauses simply make up $f$ above, which we just established to be satisfiable. The problem, however, is that the last clause does not evaluate to "true" under this assignment. So let us try another assignment. Will $x_1 = 0, x_2 = x_3 = 1$ do,
so as to remedy the problem with the last clause? It turns out that exactly the same problem
is now caused by the third clause.

If we give the problem a little thought, we will discover that no value assignment for
\(x_1, x_2, x_3\) will make the formula satisfiable since the formula contains a contradiction. The
contradiction will become apparent if we describe the formula in everyday language. The
formula constrains the variables in such a way that:

- Not all variables are assigned the same value (first and second clause).
- All variables are assigned the same values! (three remaining clauses).

In order to discern better the second constraint just rewrite the three remaining clauses as the
equivalence binary operator

\[ x_i = \implies \neg x_i, \quad x_i \implies x_j \]

and, finally,

\[ x_i \implies x_j \implies x_k \]

which says exactly that \(x_1, x_2, x_3\) are equivalent and, therefore, they must be assigned
the same value. We observe that the new formula \(f'\) resulted from a satisfiable one, \(f\), with
the addition of some new clauses. In general, as it is intuitively clear, the more clauses a
formula contains the more difficult it is to satisfy it since each additional clause constrains
some variables in some new way. Conversely, the less clauses it has, the easier it is to satisfy
it. The problem that we will address in the next sections is the way in which the number
of variables and the number of clauses affect the satisfiability of formulas. We will assume
that each clause has the same number, \(k\), of literals. We then speak about the famous \(k\)-SAT
problem: the problem of determining whether a formula written in CNF form with clauses of
exactly \(k\) literals is satisfiable.

1.1 A Little Complexity Primer

Let us now become more specific and fix the value for \(k\) to be equal to 3. That is, we will
consider formulas, over \(n\) variables, which are composed of clauses having exactly 3 literals
per clause. Suppose that \(n\) is very "large", say equal to 1000, and you are given a very large
formula too, say with \(m = 4000\) clauses. The task of deciding whether the formula is satisfiable
seems to be a really formidable one. If you consider the small example we gave in the previous
section, trying to find a truth assignment to the variables that satisfy some subset of the
clauses causes problems in other clauses of the formula. And there seems to be no easy way of
extracting some clever structural information about the clauses of the formula so that locating
a satisfying truth assignment becomes easier than simply testing whether each of the possible
\(2^n\) such assignments satisfy all the clauses. But what if we were to assert that the formula is
satisfiable? Of course you would ask to have a piece of evidence, or certificate, for our assertion.
And what's more, this certificate should be checkable fast. At least faster than it would take
you to check yourself the validity of my assertion by trying all the \(2^n\) truth assignments, a
huge number by any standards (\(n\) is a free parameter and can be unboundedly large!). And
what would be a better and faster checking certificate than giving you a truth assignment on
which the formula evaluates to true? Then you would just evaluate the formula and, indeed
very quickly (just check whether each clause contains a literal that is true under the truth
assignment that we gave you), you would be convinced of my assertion. The problems that
have the property that for each of the instances (e.g. formulas) that have the property sought
by the problem (e.g. is the formula satisfiable?) there exists an easily checkable certificate for this, constitute the important computational complexity class NP. We know that there is some obscure point here: how did we manage to find the truth assignment that we gave you as a certificate in the first place? Well, that is of no concern for the class NP! If one gives you a certificate (satisfying truth assignment) then you can check it easily. But if it happens for a problem to, also, exist a way to discover easily (fast) the certificate, then we say that the problem belongs to the class P. P stands for Polynomially fast solvable problems and NP stands for Non-deterministically Polynomially fast solvable problems. The reason for the word non-deterministically lies in the point of view that giving the certificate to someone is equivalent to him/her guessing it, since we are only interested in the existence of such a certificate and not of actually locating it which may be hard to do fast). Of course, for problems in P not only such a certificate exists, but it can also be found fast. It should be borne in mind that for computational complexity fast means always in a number of steps which is bounded from above by a polynomial in the size of the problem instance. The field of Computational Complexity has its roots in the seminal papers of Cook [12], Karp [32], and Levin [40] and has as its milestone the celebrated Cook-Levin theorem: there is no fast algorithm that can determine whether a Boolean formula is satisfiable unless there is also an algorithm that solves fast all the problems whose “yes” instances possess short certificates. Problems with this property, are called NP-complete. It should be stressed that for this problem class, we only consider worst case complexity. That is, every algorithm that runs in polynomial time and decides correctly these problems must take at least superpolynomial time on some instances. It may well be the case that the algorithm decides fast the majority of the instances. However, for some of them the algorithm must fail to finish quickly (unless, of course, P = NP). (An excellent introduction to the theory of NP-completeness can be found in [23].)

Regarding other views of complexity, Andrei Kolmogorov, Ray Solomonoff, and Gregory Chaitin independently developed ([38, 51, 7]) the concept of complexity of finite objects (now called Kolmogorov complexity), and Leonid Levin pioneered the concept of completeness in average case complexity of problems in his seminal work in the field ([41]). Although these complexity views can be shown to be relevant to the study of threshold phenomena, we will confine ourselves to the realm of NP-completeness and worst case intractability. For example, we will not attempt to examine how formulas “look like” as finite objects near the threshold region, as far as their Kolomogorov complexity is concerned, although this is an interesting line of research.

1.2 Locating “hard” formulas: experiments with 3-SAT

If the complexity of NP-complete problems can be attributed to the fact that for each polynomial time algorithm there is a set of “hard” instances that forces the algorithm to spend much time finding the solution, then an interesting problem that arises is to characterize such instances. This is interesting for the following reasons:

1. It will help us understand better the source of the complexity of NP-complete problems thus understanding the separating line between P and NP-complete problems (if P is different from NP; but no one knows yet).
2. On the practical side, it will help us to recognize whether a given real world instance (formula in the $k$-SAT problem) is likely to require much time for its resolution.

Following observations about "clusters" of hard instances of NP-complete problems ([9]) around some well definable areas, a series of experimental observations was reported in [44] (see also [33, 49]) concerning the behavior of formulas with 3 literals per clause, i.e. instances of 3-SAT. This problem has a distinguished place among all $k$-SAT problems. The reason why is that it marks the beginning of the algorithmic intractability for $k$-SAT: for $k = 1$ or 2 there are fast algorithms (can you see why the case of $k = 1$ is trivial?) which can decide whether any 1-SAT or 2-SAT formula is satisfiable. Now in order to conduct experiments that study Boolean formulas, one has to choose a method by which it is possible to check a formula for satisfiability as fast as possible. Of course, 3-SAT is a hard problem so we cannot have such an algorithm but we can at least use some systematic technique that does significantly better than trying blindly all possible truth assignments. One commonly used algorithm that can be used for deciding satisfiability of any Boolean formula is the famous Davis-Putnam (or DP for short) procedure. This procedure attempts to systematically construct a satisfying truth assignment for all the variables of a formula by satisfying a progressively larger set of clauses. The procedure is as follows, where $f$ is a Boolean formula, not necessarily a 3-SAT one, given as a set of clauses $S$:

\[ \text{DP}(f) \]

- If $S$ is empty then return “satisfiable”.
- If $S$ contains an empty clause then return “unsatisfiable”.
- If $S$ contains a unit clause (a clause consisting of only one literal) assign to the corresponding variable a value such that the literal (unit clause) evaluates to 1. Return $\text{DP}(f')$ where $f'$ is the formula that results from $f$ by deleting all of its clauses containing the literal and removing its complement from the rest of the clauses (UNIT CLAUSE rule).
- Select a variable that has not been assigned a value yet. Assign a value to it and then call $\text{DP}(f')$ where $f'$ is as before. If the call returns “satisfiable” then return “satisfiable” else set the chosen variable to the opposite value and return $\text{DP}(f')$ where again $f'$ is as before (SPLITTING RULE).

Observe that the procedure is recursive and that in the worst case it will explore all the possible truth assignments (step 4 will try both 0 and 1 on each variable). The experiments were conducted as follows: for each value of $n$ (number of variables) a set of formulas was randomly produced for various values of $m$ (number of clauses) and their satisfiability was tested using the DP procedure. For each of them, the number of DP calls was recorded. This number is representative of the running time of the DP procedure because it is these calls that are the most time consuming. The results of the various runs are depicted in Figure 1, with the number of clauses being divided by $n$ so that on the x-axis the ratio $m/n$ appears.

The remarkable observation is that the DP procedure appears to find difficulties with formulas whose $m/n$ ratio clusters around a value between 4 and 4.5. For larger or smaller ratios,
DP decides satisfiability considerably faster. An intuitive explanation of this phenomenon goes as follows: in the region where formulas have a clause to variable ratio less than the value where a peak is observed, the "underconstrained" region, most of the formulas generally have a very large number of solutions so the DP procedure has the odds on its side of locating one of them. In a similar fashion, in the "overconstrained" region, almost all formulas are unsatisfiable, usually containing many contradictions like the one we saw in Section 1. Therefore, the DP procedure is again lucky and will hit such a contradiction soon, without needing to check every possible value assignment in order to expose such a contradiction. But near the peak, the formulas have the property that many partial value assignments can be extended by DP to larger assignments (Steps 3 and 4) without exposing an inconsistency (so far) only to be discovered near the last variables that the partial assignment cannot belong to a global satisfying truth assignment. Therefore the DP procedure seems to get stuck to "would-be" solutions that it is unable to characterize early as non-solutions. The plot in Figure 2 shows the percentage of satisfiable clauses, for various m/n ratios.

With the help of the previous figure, it can be seen that the m/n ratio around which the difficult formulas are clustered, is also the point where the generated formulas suddenly change from being almost all satisfiable to being almost all unsatisfiable. We say that random formulas undergo a phase transition, where the two phases of a formula are the "satisfiable" and the "unsatisfiable". A daring conjecture to which one may be led by the above observations, is that it is possible that the NP-completeness property (or solution difficulty, may we say) of k-SAT stems from those instances that are clustered around this critical m/n ratio and impose the most burden to any polynomial algorithm that solves correctly the problem. We should, however, keep in mind that it is not generally true that phase transitions are a defining characteristic of NP-complete problems. It has been rigorously proved ([10, 25]) that 2-SAT random formulas undergo a phase transition at m/n = 1, even though 2-SAT is in P. Also, for some NP-complete problems, like the Traveling Salesman Problem, hard instances do not
cluster around some parameter like the SAT $m/n$ ratio.

1.3 Rigorous treatment for Phase Transitions: 3-SAT

In order to investigate the asymptotic satisfiability properties of 3-SAT formulas, as a function of the clauses to variables ratio $r = m/n$, we may use the following model for their construction: out of the possible $8\binom{m}{3}$ clauses with 3 literals, we randomly select, with repetitions allowed, $m$ clauses to be the conjuncts of the formula. The formula thus constructed is aptly called random formula. Now, what is the probability that such a random formula $f$ is satisfiable? In order to determine this probability, define for each possible truth assignment $S$ to the $n$ variables an indicator variable that simply indicates whether the formula evaluates to true under $S$:

$$I_S(f) = 1, \text{ if } f \text{ is true under } S \text{ and 0 otherwise.}$$

The parameter $f$ is the random variable in our case. Then if we denote by $\#S(f)$ the number of truth assignments under which $f$ evaluates to "true" it is not hard to see that $f$ is satisfiable if and only if $\#S(f)$ is larger than 0. But:

$$\#S(f) = \sum_S I_S(f)$$

Therefore:

$$\Pr[f \text{ is satisfiable}] = \Pr[\#S(f) > 0] = \Pr\left[\sum_S I_S(f) > 0\right]$$

where the last sum contains $2^n$ terms since this is the number of possible truth assignments to $n$ variables. A very useful result in probability theory, called Markov's inequality, gives us an easy way to bound from above this sum by its expectation $E$ (see, for example, [20, 11]):

$$\Pr[\sum_S I_S(f) > 0] \leq E[\sum_S I_S(f)] = \sum_S E[I_S(f)].$$
The last equality follows from the principle of linearity of expectation. But \( I_S(f) \) is an indicator variable and, therefore, its expectation is equal to the probability that it assumes the value 1. Let us calculate this probability. Fix a truth assignment \( S \). Then observe that out of the possible \( 8(\binom{n}{3}) \) clauses with three literals over \( n \) variables, only \( \binom{n}{3} \) evaluate to “false” under \( S \). This is because for any set of three variables which are assigned some value by \( S \), there corresponds exactly one clause that does not evaluate to “true” under these specific value assignments: the clause that contains the variables that are assigned the value 0 unnegated and the variables that are assigned the value 1 negated. Moreover, for a value assignment \( S \), the indicator variable \( I_S(f) \) takes the value 1, if the formula \( f \) that was randomly constructed evaluates to 1 under \( S \). But in order to evaluate to 1 under \( S \), \( f \) must have avoided all \( \binom{n}{3} \) clauses that are false, under \( S \) again. The probability that during the selection (with repetitions) of \( m = r n \) clauses we have avoided to draw these clauses is:

\[
\left( 1 - \frac{\binom{n}{3}}{8(\binom{n}{3})} \right)^{rn} = \left( \frac{7}{8} \right)^{rn}.
\]

Note that this probability is the same for all truth assignments. Then, returning to Markov's inequality:

\[
\Pr \left[ \sum_S I_S(f) > 0 \right] \leq \sum_S E[I_S(f)] = 2^n \left( \frac{7}{8} \right)^{rn} = \left[ 2 \left( \frac{7}{8} \right)^r \right]^n.
\]

This is a very important inequality. Imagine that the number of variables gets unboundedly large (it tends to infinity). Then if the clauses to variable ratio \( r \) was such so that the quantity \( 2(7/8)^r \) was strictly less than 1, then the above inequality immediately proves that, in the limit, the random formula is unsatisfiable. Thus, if the above condition holds, we say that almost all formulas containing \( m = r n \) clauses are unsatisfiable when \( n \) tends to infinity. Let us then see, what is the smallest value of \( r \) for which Markov’s inequality asserts this fact:

\[
2 \left( \frac{7}{8} \right)^r < 1 \Leftrightarrow \ln 2 + r \ln \frac{7}{8} < 0 \Leftrightarrow r > -\frac{\ln 2}{\ln \frac{7}{8}} \approx 5.1909.
\]

The above technique, that relies on Markov’s inequality in order to bound from above the probability that a nonnegative random variable assumes a value strictly greater than 0, is called the first moment method.

Now we have rigorously proved that, as \( n \) tends to infinity, the probability that a random formula with \( m = r n \) clauses is satisfiable tends to 0, whenever \( r > 5.1909 \). How does this blend with the experimental evidence? Well, at least it approximates (but rigorously) the presumed threshold value of 4.2 that was determined by experiments. But what about \( r < 5.1909 \)? Of course, in this case we cannot conclude anything from the above analysis. We know from experimental data that from 5.1909 down to 4.2 the formulas must still be asymptotically unsatisfiable. But how can this be rigorously proved? We will not go into any details here but we will simply say that one possible approach that has been pursued, resembles the approach we presented with Markov’s inequality and consists in defining more refined versions of it in order to further reduce the base (i.e. \( 2(7/8)^r \)) of the exponent \( (n) \), thus making the convergence to 0 possible for values of \( r \) lower than 5.1909. One such improved Markov inequality was first presented by Kirousis, Kranakis, and Krizanc in [35]. This inequality was,
independently, obtained by Dubois and Boufkhad in [13]. Another approach based on Aldous harmonic mean formula and exact calculations of probabilities of the occupancy problem is presented by Karnath, Motwani, Palem, and Spirakis in [29] while the best currently known upper bound, 4.507, was announced by Dubois, Boufkhad, and Mandler in [14] and given with complete proofs in [15]. In Section 3 we will study in detail some of these approaches.

On the other hand, regarding the approximation of the threshold value from below, the best lower bound up to now is given by Kaporis, Kirousis, and Lalas in [31], where it is shown that a DP-like heuristic based on the number of appearances of literals in the clauses succeeds in finding a satisfying truth assignment for almost all formulas with \( m/n < 3.42 \). The analysis is based on a technique that we will present in Section 2.

1.4 A significant advance

In [21], Friedgut proved a result that essentially establishes the threshold behavior of \( k \)-SAT, without however establishing specific values (such as, for example, the conjectured 4.2 for 3-SAT). More specifically, Friedgut showed the following theorem:

**Theorem 1** If \( \phi \) is a random \( k \)-SAT formula on \( n \) variables with \( rn \) clauses, let \( S_k(n, r) = \Pr[\phi \text{ is satisfiable}] \). Then for every \( k \geq 2 \), there exists a sequence \( r_k(n) \) such that for all \( \epsilon > 0 \)

\[
\lim_{n \to \infty} S_k(n, r_k(n) - \epsilon) = 1 \quad \text{and} \quad \lim_{n \to \infty} S_k(n, r_k(n) + \epsilon) = 0.
\]

Notice that the above does not really resolve the satisfiability threshold conjecture since \( r_k(n) \) depends on \( n \). If \( r_k(n) \) were to converge to a constant, then the conjecture would be resolved, but noone knows if convergence occurs.

On the other hand, the following corollary to Friedgut's theorem is often useful in analyzing heuristics for obtaining, at least, lower bounds to the threshold value:

**Corollary 1** If \( r \) is such that \( \liminf_{n \to \infty} S_k(n, r) > 0 \) then for any \( \epsilon > 0 \), \( \lim_{n \to \infty} S_k(n, r - \epsilon) = 1 \).

1.5 Threshold and "micro-threshold" functions

The concept of a threshold function or transition point in connection with properties of combinatorial objects, such as graphs, is well understood in discrete mathematics and combinatorics (see [5] for a thorough treatment of threshold phenomena in relation to random graph properties). However, the suggestion to look at this concept from a fresh perspective was given by Cheeseman, Kanefsky, and Taylor in [9]. One of the problems they examined was a problem equivalent to 3-SAT, in the complexity theoretic framework of NP-completeness, i.e. they are both computationally intractable and if one of them could be solved efficiently, then a multitude of other problems believed to be computationally intractable would also be solvable efficiently. The problem was that of colouring the vertices of a graph with three colours, also known as 3-COLOURING, in a way such that no two adjacent vertices are assigned the same colour. The graphs that can be coloured with 3 colours are called 3-colourable. In the theory of random graphs (see [5]), we are interested in whether a randomly formed graph possesses a property,
such as being 3-colourable, or not. A random graph with $n$ vertices and $m$ edges is most commonly formed according to the following model: from the set of possible $\binom{n}{2}$ edges, select uniformly and without replacement $m$ edges to belong to the graph. Now a natural question that arises is the following: for various values of $m$ (chosen edges), what is the probability that a random graph of $m$ edges possesses the property in question as $n$ tends to infinity? Let us consider the property of a graph being colourable with 3 colours. If $m = \omega(n)$, meaning that as $n$ grows, $m$ grows so that the ratio $m/n$ tends to infinity, it can be easily proved using the first moment method that was applied above to the 3-SAT problem that with probability tending to 1, a random graph with $m$ edges will not be 3-colourable. On the other hand, if $m = o(n)$, meaning that as $n$ tends to infinity $m/n$ tends to 0, we can use a result from the theory of random graphs (see the book of Bollobás [5], Corollary 5.8, page 105) that states that in this case, with probability tending to 1, every component of a random graph with $m$ edges is either a tree or a unicyclic graph (i.e. a non-chordal ring with trees attached to some of its nodes). But this means that the graph can be coloured with at most three colours.

From the above discussion, we conclude that the function $f(n) = n$ marks a so-called, threshold area, in the sense that if the number of edges in a randomly formed graph grows slower or faster than $f(n)$ then we observe in each case a different behaviour, with probability that tends to 1. We, then, say that $f(n) = n$ is a threshold function for the property of 3-colourability.

What happens, however, when $m = \Theta(n)$, i.e. when $m/n$ tends to a positive constant value $r$? Well, in this case it may or may not be true that, almost certainly, a randomly formed graph with $m$ edges can be coloured with 3 colours. The key factor is the exact value of $r$, the constant itself. Therefore, we shift our attention to the study of the "micro-threshold" behaviour, i.e. we fix the order of growth of $m$ to be $n$ and we focus on discovering the ranges of $r$ that correspond to graphs that are or are not 3-colourable with probability that tends to 1.

To return to Cheeseman, Kanefsky, and Taylor, their experiments demonstrated that for values of $r$ outside the interval $(4,6)$ (approximately), the random graphs with $rn$ edges were either almost all 3-colourable or almost none 3-colourable. This suggests that there may by some value for $r$ in $(4,6)$ for which we may observe an abrupt transition from almost certain 3-colourability to almost certain non 3-colourability of the random graphs with $rn$ edges. Indeed, their experiments indicated that the transition takes place around the value $r = 5.4$.

As we said, however, from a complexity-theoretic perspective their most important observation was that for $r$ in the "threshold" region $(4,6)$ the randomly produced graphs seemed to cause great difficulty to the most efficient graph colouring algorithms. Therefore, it was conjectured that computationally intractable problems, such as 3-COLOURING and 3-SAT are intractable because of problem instances whose parameters (e.g. number of edges of graphs) fall around a threshold point where we have an abrupt change of behaviour in an observed property (e.g. 3-colourability, satisfiability). Although this conjecture does not seem to hold, at least in its full generality (for example 2-SAT, which is a tractable problem, also displays a phase transition behaviour and the Traveling Salesman Problem or TSP, an intractable problem, does not seem to have a clearly defined transition region), the results of Cheeseman et al. were followed by a series of theoretical and experimental studies on various problems (most often 3-SAT) to discover the regions where the transitions take place as well as the relationship
between the transition regions and the problem's hardness. In this review, we will be concerned with the efforts concerning the determination of the threshold region of 3-SAT. We will review the history of the theoretical efforts as well as the successive improvements in the accuracy of the description of the threshold region.

2 Ideas for lower bounds

In this section we will describe the main ideas and techniques used to prove lower bounds to the satisfiability threshold. As we will see, the techniques differ radically from the techniques used for upper bounds. Upper bounds are proved using some variation or improvement of the Markov inequality (we will see more on this in Section 3). On the other hand, lower bounds are proved using a backtrack-free algorithm that in a series of simple steps assigns a value to all the variables, one at a time (although there exist variations to this rule, as we will see below) according to some specific rule that characterizes the algorithm.

Probably the most primitive algorithm for satisfiability is Unit Clause (UC): if there are clauses with one literal (1-clauses), select randomly a literal that appears in one of these clauses and set it to 0 or 1; otherwise, randomly select a literal and set it to 0 or 1. (In what follows, u.a.r. stands for uniformly at random.)

Algorithm: UC (UNIT CLAUSE)

Input: A random 3-SAT formula on \( n \) variables

Output: A satisfying truth assignment on its variables or fail

1. begin
2. if there exist 1-clauses (unit clauses) then
3. select a unit clause u.a.r. and satisfy it (forced step)
4. else
5. pick an unset variable u.a.r. and assign to it 0 or 1 u.a.r. (free step)
6. end

Each time UC sets a literal to true, the clauses that contain it are deleted and the clauses that contain its negation are shrunk by deleting the occurrence of the unsatisfied literal. Thus at each step \( t \) of the execution of the algorithm, and as long as unsatisfiability has not been reported, there are three types of possible clauses: the ones that contain exactly \( i \) literals, or \( i \)-clauses. for \( i = 1, 2, 3 \). These sets will be denoted by \( S_i(t) \) (\( i = 1, 2, 3 \), respectively) and will also be called \( i \)-buckets. If \( |S_1(t)| > 0 \) and UC happens to satisfy the 1-clause \( \{l\} \in S_1(t) \), then if \( \{-l\} \in S_1(t) \) too, then an empty clause will be generated, which signifies the failure of the algorithm. Thus, we also have a set with 0-clauses, denoted by \( S_0(t) \). In order, however, to facilitate the analysis, we will assume that the algorithm continues its execution normally until it has assigned values to all \( n \) variables.

In [1], Achlioptas devised a very nice way to depict the clause flows that result from the actions of the algorithm. This way is reproduced here in Figure 3.

The UC algorithm was first analyzed by Chao and Franco (see [8]) and was proved to succeed a.s. (almost surely) for formulas with density up to 8/3. Many other improved heuristics
Figure 3: Transformations of clauses

that differ from UC only at the free steps, i.e., when there are no 1-clauses available, were subsequently introduced and analyzed, some of which will be described later in this section.

In [1], Achlioptas introduced a generic framework that greatly assists the definition and analysis of algorithms, such as UC. His framework, represents a k-SAT (in the general case) formula with m clauses, with m adjacent stacks of k cards, each representing a k-clause of the formula. Obviously, the cards in each stack represent the literals of the corresponding clause. We also assume that all the cards are initially placed face-down so that the literal represented is concealed. All the manipulations to the stacks of cards are effected through an intermediary who is assumed to know the literals represented by the cards. We (i.e., the algorithm) are allowed to do one of the following:

1. Point to one of the cards.
2. Name one of the variables that has not yet received a value.

If we chose to point to a particular card that happens to represent the literal \( l \), then the intermediary reveals all the cards that contain either \( l \) or \( \neg l \). If, on the other hand, we chose to name one of the variables, say \( v \), then the intermediary flips all the cards containing either \( v \) or \( \neg v \). In both cases, the next thing to do is to decide which value to assign to the chosen variable. After the decision has been made (using, perhaps, the revealed information), the intermediary proceeds to delete all stacks that contain the satisfied literal and to delete all cards that represent the dissatisfied literal. In either case, the result is a set of stacks of cards that do not contain the assigned variable. Figure 4, given by Achlioptas in [1], shows a typical configuration snapshot at some stage of the card game. The key to the analysis of the algorithms that can be expressed in the framework of the card game is the preservation of “randomness” of the formula that results after the manipulations are effected by the intermediary: the resulting formula is *uniformly random*, given the information exposed in the previous steps. This fact is crucial for the probabilistic analysis of the algorithms.

We will now review a number of heuristics that can be expressed and analyzed within the framework of the card game. The Majority Rule (Mj) at a free step chooses randomly a literal \( l \) and sets to 1 either \( l \) or \( \neg l \), depending on whose degree with respect to the 3-bucket is larger (ties are broken randomly). Thus the creation of newly shrunk 2-clauses is slowed down. Mj
succeeds a.s. for formulas with density at most 2.99 ([8]). The Generalized Unit Clause rule, (GUC) randomly selects and sets to 1 a literal that appears among a randomly chosen clause of the clauses that are presently available and have the shortest length. The main feature of GUC is that it guarantees that at a free step the size of the 2-bucket will be reduced by at least one. GUC succeeds for densities up to 3.003 (see, also, [22]).

The two-at-time rule (TT), introduced and analyzed by Achlioptas [1], at a free step selects one clause from the 2-bucket and from the three possible truth assignments to the literals of this clause that satisfy it, selects the one that shrinks the least possible number of 3-clauses (ties are broken randomly; also when no 2-clauses are available a literal is selected and set randomly). TT succeeds for densities up to 3.14.

An important idea that is crucial in the analysis of the heuristics that are used for proving lower bounds is to satisfy the 1-clauses not whenever they are available, but only with a probability larger than (but arbitrarily close to) the flow from the 2-bucket to the 1-bucket (as long as this flow remains below 1). The reason for this variation is only technical: first it guarantees that if the flow from the 2-bucket to the 1-bucket is less than one, the size of the 1-bucket will remain constant, and so a.s. (almost surely) no contradiction (empty clause) will be generated, and second it permits the evolution of the size of the buckets to be expressed by differential equations in a way that can be proved correct in a relatively easier way than when we always set an available random 1-clause. This variation was formalized in [1] by the so-called "lazy-server lemma".

Finally, Achlioptas and Sorkin ([4]), observed that the common characteristic of all these heuristics is that at each step a literal is selected either randomly from a random clause in a specified non-empty bucket or randomly from the totality of literals. Moreover, the truth value given to the selected literal depends only on its degrees (and the degrees of its negation) with respect to each bucket. Considering heuristics where the rules for picking and setting a literal retain the above mentioned characteristics, but are allowed to change even during the course of the algorithm (and even allowing for more than one literal to be selected at each step before setting them), they introduced the framework of the myopic algorithms and proved that the highest density a myopic algorithm may a.s. succeed is 3.26.
2.1 The main idea

Our discussion in the previous section has already indicated the general approach of proving that $r$ is a lower bound to the threshold: we design a suitable (i.e. easy to analyze) heuristic for assigning values to the variables of a formula with $rn$ clauses and then prove that with high probability the algorithm succeeds, i.e. it comes up with a truth assignment that satisfies the formula. And the algorithm succeeds, if no empty clause is ever generated during its execution.

As an empty clause can be generated at step $t + 1$ only if $S_1(t) > 0$, we can conclude that the algorithm used should strive to keep $|S_i(t)|$ as low as possible, in order to minimize the probability that both $l$ and $\neg l$ are in $S_1(t)$ at some execution step. And as the 1-clauses of $S_1(t)$ can only result from shrunk 2-clauses, we will need to analyze $S_i(t)$ for all $t$.

In what follows, we will denote $|S_i(t)|$ by $C_i(t)$, $i = 1, 2, 3$.

It can be proved that the density of the 2-clause subformula, which is equal to $C_2(t)/(n-t)$ at the $t$th step of the algorithm execution, is the crucial parameter that determines the fate of the algorithm, as formalized by the following Lemma (for more explanations and a proof see [1]):

**Lemma 1** Let $A$ be any algorithm that can be expressed within the card game framework such that it always satisfy a unit clause whenever such clauses exist. If $\delta, \epsilon > 0$ and $t_e$ are such that $t_e \leq (1 - \epsilon)n$ and with high probability

$$C_2(t) < (1 - \delta)(n - t), \forall t \in \{0, \ldots, t_e\},$$

then there exists $\rho = \rho(\delta, \epsilon) > 0$ such that $\Pr[S_0(t_e) \cup S_1(t_e)] > \rho$.

Lemma 1 suggests that we need some expression for the quantity $C_2$ which, however, also depend on $C_3$ since it is the 3-clauses to 2-clauses flow that determines the set $C_2$.

2.2 The differential equations method

By now, it should be clear that our main goal should be the determination of $C_2(t)$, that is the number of 2-clauses at each execution step. Looking at the actions that UC performs, we lead to the conclusion that $C_2(t)$ and $C_3(t)$ are, in fact, two sequences of discrete time, discrete random variables or discrete random processes with $C_3(t)$ affecting $C_2(t)$ as shrunk 3-clauses end up in the bucket with the 2-clauses, thus increasing $C_2(t)$. We say that $C_2(t)$ and $C_3(t)$ are two co-evolving random processes.

But now things seem rather disconcerting! There does not seem to be some easy way of determining the two entangled sequences, let alone the fact that they are random and may behave differently at different evolutions of the algorithm on the same initial formula. Fortunately, there is a way out of this problem. It is possible to approximate the “typical trajectories over time” of the two processes with the trajectories of two deterministic functions in the continuous time domain that are the unique solution of a system of differential equations defined by the way the algorithm under analysis works. Theorem 2 below, due to Wormald ([57]) gives us a powerful tool that achieves the goal (see, also, [56]).

In applying Wormald's theorem in a combinatorial setting one usually deals with a random object (e.g. a graph or a formula) on which a randomized algorithm is applied for $O(n)$ steps.
where \( n \) is involved in measuring the object's size (e.g. number of variables in a formula or number of vertices in a graph). At each step of the application of the algorithm one is interested in keeping track of a number of (possibly dependent on each other) counting variables (e.g. number of 2-clauses and 3-clauses in a formula or number of vertices of up to a certain degree in a graph) that change as the algorithm manipulates the object (e.g. chooses and sets to TRUE a variable or deletes a vertex) and do not grow larger than \( O(n) \). As the algorithm is randomized and it is also applied to a random object, the counting variables are random variables. However, in the problem at hand, the values of these variables at each step of the application of the algorithm usually provides the value of another quantity dependent on them that gives the answer to the problem (e.g. the density of 2-clauses in the 3-SAT example). At this point it is where Wormald's theorem gives the solution: provided the random variables in question do not vary "wildly", one can find a deterministic function for each of them such that the corresponding random variable stays within a "narrow region" around the deterministic function, over all time steps for which the algorithm is applied. Thus, for the purposes of the problem, the random variables are, in some sense, "predictable" over all time steps. Also, in order to dispense with the dependence on the size of the object \( n \) and avoid finding deterministic functions for each such \( n \), we divide all quantities (discrete time plus random variables) by \( n \) thus scaling them into real numbers and making them independent of the size of the object.

We end up calculating deterministic, real functions and by multiplying by \( n \) we get back the approximation to the random variables in question in the discrete time domain.

Before embarking on the statement of Wormald's theorem, we will give some auxiliary definitions.

**Definition 1** A function \( f \) satisfies a Lipschitz condition on \( D \subset \mathbb{R}^j \) if there exists some constant \( L > 0 \) such that

\[
|f(u_1, \ldots, u_j) - f(v_1, \ldots, v_j)| \leq L \sum_{i=1}^{j} |u_i - v_i|
\]

for all \((u_1, \ldots, u_j)\) and \((v_1, \ldots, v_j)\) in \( D \).

**Definition 2** Given a random variable \( X \) depending on \( n \), denoted by \( X^{(n)} \), we say that \( X^{(n)} = o(f(n)) \) always if

\[
\max\{z | \Pr[X^{(n)} = z] \neq 0\} = o(f(n)).
\]

**Theorem 2** Let \( Y_i^{(n)}(t), n \geq 1, \) be a sequence of real-valued random variables, \( 1 \leq i \leq k \) for some fixed \( k \), such that for all \( i, t \) and all \( n \), \( |Y_i^{(n)}(t)| \leq Bn \) (\( n > 0 \)) for some constant \( B \). Let \( H(t) \) be the history of the sequence, i.e. the matrix \( (Y(0), \ldots, Y(t)) \), where \( Y(t) = (Y_1^{(n)}(t), \ldots, Y_k^{(n)}(t)) \).

Let \( I = \{(y_1, \ldots, y_k) : \Pr[Y(0) = (y_1n, \ldots, y_kn)] \neq 0 \text{ for some } n\} \). Let \( D \) be some bounded connected open set containing the intersection of \( \{(s, y_1, \ldots, y_k) : s \geq 0\} \) with a neighbourhood of \( \{(t/n, y_1, \ldots, y_k) : (y_1, \ldots, y_k) \in I\} \). (That is, after taking a ball around the set \( I \), \( D \) is required to contain the part of the ball in the halfspace corresponding to \( s = t/n, s \geq 0 \).)

Let \( f_i : \mathbb{R}^{k+1} \rightarrow \mathbb{R}, 1 \leq i \leq k, \), and suppose that for some \( m = m(n) \),
(i) for all \( i \) and uniformly over all \( t < \ell \),

\[
E[Y^{(n)}_i(t + 1) - Y^{(n)}_i(t)|H(t)] = f_i(t/n, Y^{(n)}_0(t)/n, \ldots, Y^{(n)}_k(t)/n) + o(1), \text{ always,}
\]

(This condition ensures that the expected rate of change of each one of the random variables of interest can be approximated well by a deterministic function.)

(ii) for all \( i \) and uniformly over all \( t < \ell \),

\[
\Pr[|Y^{(n)}_i(t + 1) - Y^{(n)}_i(t)| > n^{1/3}] = o(n^{-3}), \text{ always,}
\]

(This condition ensures that the random variables of interest evolve in a "smooth" way from step to step.)

(iii) for each \( i \), the function \( f_i \) is continuous and satisfies a Lipschitz condition on \( D \). (This condition ensures that the expected rate of change stated in (i) above does not change much over time or as the values of the random variables change.)

Then

(a) for \((0, \tilde{z}^{(0)}, \ldots, \tilde{z}^{(k)}) \in D\) the system of differential equations

\[
\frac{dz_i}{ds} = f_i(s, z_0, \ldots, z_k), 1 \leq i \leq k
\]

has a unique solution in \( D \) for \( z_i : \mathbb{R} \to \mathbb{R} \) passing through \( z_i(0) = \tilde{z}^{(i)}, 1 \leq i \leq k \), and which extends to points arbitrarily close to the boundary of \( D \);

(b) almost surely

\[
Y^{(n)}_i(t) = z_i(t/n) \cdot n + o(n),
\]

uniformly for \( 0 \leq t \leq \min\{n, m\} \) and for each \( i \), where \( z_i(s) \) is the solution in (a) with \( \tilde{z}^{(i)} = Y^{(n)}_i(0)/n \), and \( \sigma = \sigma(n) \) is the supremum of those \( s \) to which the solution can be extended.

Although the statement of the theorem is rather daunting, the truth is that once understood at the intuitive level (see the explanations before the statement of the theorem), its use arises naturally from the application in hand.

As we said above, what the theorem essentially says is that if we are confronted with a number of (possibly) interrelated random variables (associated with some random process) such that they satisfy a Lipschitz condition and their expected fluctuation at each time step is known, then the value of these variables can be approximated using the solution of a system of differential equations. Furthermore, the system of differential equation results directly from the expressions for the expected fluctuation of the random variables describing the random process. (See, also, [1] for a particularly enlightening discussion on the intuition behind the theorem.)
2.3 Analyzing UC

Let $C_i(t)$ denote the number of $i$-clauses remaining in the formula after $t$ variables have been set by the algorithm. Let us summarize some facts:

- The expected number of 1-clauses that are generated in round $t$ of the algorithm is $\frac{C_1(t)}{n-t} + o(1)$.
- If there exists $t$ such that $\frac{C_1(t)}{n-t} > (1 + \delta)$ then the algorithm will a.s. fail.
- If for every round $t$ it holds $\frac{C_1(t)}{n-t} < (1 - \delta)$ then the algorithm succeeds with probability bounded below by a non-zero probability $p(\delta)$.

It turns out that if one can show that for a specific value $r$ of the clause to variable ratio $m/n$ it holds

$$\frac{C_2(t)}{n-t} < (1 - \delta)$$

for all rounds $t$ then $r$ is a lower bound for the satisfiability threshold.

Let us analyze the Unit Clause (UC) heuristic. At each step, we assign a random value (0 or 1) to a variable chosen at random from the remaining variables. Then the expected difference in the number of 2-clauses is

$$\mathbb{E}[\Delta C_2(t)] = \mathbb{E}[C_2(t + 1) - C_2(t)] = f(t, C_2(t), C_3(t)). \quad (1)$$

We can use differential equations to approximate the mean path of the underlying Markov chain by replacing the discrete parameter $n$ by $x = t/n$, as follows:

$$\mathbb{E}[\Delta C_3(t)] = \frac{-3C_3(t)}{n-t} \quad \text{gives} \quad c_3(x) = \frac{-3c_3(x)}{1-x}$$

$$C_3(0) = rn \quad \text{gives} \quad c_3(0) = r$$

$$\mathbb{E}[\Delta C_2(t)] = \frac{1}{2n-t} \frac{3C_2(t)}{n-t} - \frac{2C_2(t)}{n-t} \quad \text{gives} \quad c_2(x) = \frac{3c_2(x)}{2(1-x)} - \frac{2c_2(x)}{1-x}$$

$$C_2(0) = 0 \quad \text{gives} \quad c_2(0) = 0. \quad (2)$$

The solutions to the above differential equations for $c_2(t)$ and $c_3(t)$ are the following:

$$c_2(t) = \frac{3}{2} r x (1-x)^2, c_3(t) = r (1-x)^3.$$

Applying Theorem 2 gives us the following (see also [1]):

**Lemma 2** Fix $\epsilon > 0$ and let $t_\epsilon = \lfloor (1 - \epsilon)n \rfloor$. If UC is applied to a random 3-SAT formula with $rn$ clauses on $n$ variables, then with high probability, for each $t \in \{0, \ldots, t_\epsilon\}$,

$$C_i(t) = c_i(t/n) \cdot n + o(n), i = 2, 3$$

with $c_2(x) = \frac{3}{2} r x (1-x)^2, c_3(x) = r (1-x)^3$. 

215
PROOF Using Chernoff bounds (see [1] for details), we can easily bound $\Pr[|C_i(t) - n^{1/5}|nH(t)]$, $i = 2, 3$. Then for each $\epsilon > 0$ Theorem 2 is applied with the following correspondences:

$$k = 2, Y_i^{(n)}(t) = C_{i+1}(t), B = r, m = n - 3$$

$$f_1(s, z_1, z_2) = \frac{3z_1}{2(1-s)} - \frac{2z_2}{1-s}, f_2(s, z_1, z_2) = -\frac{3z_2}{2(1-s)},$$

and $D$ defined by $-\epsilon < s < 1$ and $-\epsilon < z_i < r$ with $i = 1, 2$.

Let us now determine the maximum value of $r$ for which UC succeeds with high probability. Remember that the key quantity is the density of the 2-clauses at time $t$, $C_2(t)/(n-t)$, which corresponds to $c_2(x)/(1-x) = \frac{3}{2}r(x(1-x))$ in the deterministic domain. We now solve the inequality $c_2(x) < 1$. Observe that $c_2(x)$ is maximized for $x = 1/2$. Therefore, it suffices to ensure $c_2(1/2) < 1$, which holds true if $r < 8/3$.

Set, now, $r = \frac{8}{3}(1+\delta)$, for $\delta > 0$, and $t_0 = [n/2]$. Then Lemma 2 tells us that $C_2(t_0) = c_2(t_0/n) \cdot n + o(n)$. Omitting $o(n)$ (as we are dealing with ratios of expressions with leading terms $O(n)$), we have

$$C_2(t_0) = \left(1 + \frac{\delta}{2}\right)(n - t_0),$$

which tells us that UC fails with high probability for $r > 8/3$.

We now need to show that for $r < 8/3$ the algorithm succeeds with high probability. This is left as a (not easy) exercise for the reader (see [1]) that requires the use of Corollary 1.

2.4 Analyzing GUC

After the success of the differential equations method with UC, let us now try our luck with another heuristic for satisfying a given formula: the Generalized Unit Clause (GUC) heuristic:

**Algorithm:** GENERALIZED SHORT CLAUSE (GUC)

**Input:** A random 3-SAT formula on $n$ variables

**Output:** A satisfying truth assignment on its variables or fail

1. begin
2. if there exist 1-clauses (unit clauses) then
3. select a unit clause u.a.r. and satisfy it (forced step)
4. else (free step)
5. select one clause of shortest length and satisfy one of its literals chosen u.a.r.
6. end

Notice that for GUC, the expected difference in $C_3(t)$ is the same as in UC. Thus, replacing $n$ by $x = t/n$, we have the following:

$$E[\Delta C_3(t)] = -\frac{3C_3(t)}{n-t} \quad \text{gives} \quad c_3' = \frac{3c_3(x)}{1-x}$$

$$C_3(0) = r \quad \text{gives} \quad c_3(0) = r.$$
For $C_2$:

\[
E[\Delta C_2(t)] = \frac{3 C_3(t)}{2 n - t} - \frac{2 C_2(t)}{n - t} - \left(1 - \frac{C_2(t)}{n - t}\right)
\]
gives

\[
c_2(x)' = \frac{3c_3(x)}{2(1 - x)} - \frac{c_2(x)}{1 - x} - 1
\]

\[
C_2(0) = 0\text{ with } x = \frac{t}{n}\text{ gives } c_2(0) = 0.
\]

Solving the above equation we obtain

\[
c_2(t) = \frac{3}{2} r x \left(1 - \frac{x}{2}\right) + \ln(1 - x).
\]

Finally, solving $c_2(t) < 1$ we obtain $r > 3.003$ and reasoning similarly as in Section 2.3 we conclude that a lower bound for the threshold is the value 3.003.

### 2.5 Why the improvement of GUC over UC?

From the descriptions of UC and GUC, it should be apparent that if no 2-clauses exist, their behaviour is similar. They deviate from each other only if 2-clauses exist, in which case GUC always chooses to select and satisfy a 2-clause rather than blindly choosing to satisfy a variable that may end up pushing shrunk 3-clauses into the 2-clause bucket, thus increasing their density. Giving priority on satisfying 2-clauses has the advantage of keeping the density of 2-clauses low, which, in turn, keeps the flow into 1-clauses low. This is exactly the “secret” of success of GUC, that is translated into the improvement of the lower bound.

### 2.6 Two variables at a time

In algorithms UC and GUC, we saw that at each step only one variable was assigned a value. A deviation from this rule is exhibited in the algorithm below that was given by Achlioptas in [2]:

**Algorithm: TT (TWO AT A TIME)**

**Input:** A random 3-SAT formula on $n$ variables

**Output:** A satisfying truth assignment on its variables or fail

1. begin
2. if there exist 1-clauses (unit clauses) then
3. select a unit clause u.a.r. and satisfy it
4. else
5. if 2-clauses exist then select a 2-clause u.a.r. and gently-satisfy it
6. else select an unset variable u.a.r. and randomly assign 0 or 1 to it
7. end

where *gently-satisfy* a 2-clause with literals $l_1, l_2$ has the meaning that *among the three possible truth assignments that satisfy the 2-clause, choose the one that minimizes the number of 3-clauses that shrink into 2-clauses.*
However, for technical reasons, a variation of the above algorithm was analyzed (see [2] for details). The analysis (as expected!) was more difficult to carry out with the differential equations method than it was for UC and GUC but it led to a better lower bound, equal to 3.145.

2.7 Considering alternatives

A careful consideration of the alternatives available to an algorithm set forth to satisfy a formula suggests that there exist two reasonable strategies that one may follow:

1. Minimize the flow from the bucket of 3-clauses to the bucket of 2-clauses in order to keep the density of the 2-clauses low.

2. Minimize the flow from 2-clauses to 1-clauses in order to have available more free steps, i.e., steps that the algorithm has the freedom to select the best continuation for it.

If a variable can be set so that both the 3-clause to 2-clause flow as well as the 2-clause to 1-clause are minimized, then there is not much to think about. However, there may be situations where the variable can be set in more than one way minimizing either of the two flows. Then what should the algorithm do? In [4], Achlioptas and Sorkin considered this trade-off in detail and defined the family of myopic heuristics that pushed the lower bound up to 3.26.

3 Ideas for upper bounds

As we discussed in Section 1 of the survey, the mathematical tool that was first employed for attacking the upper bounds side of the unsatisfiability threshold conjecture, was the first moment method that makes use of Markov's inequality. We will first, briefly, repeat the main idea here in a more general setting. Let $X$ be a nonnegative integer random variable and $E[X]$ its expectation. Then, according to Markov's inequality, it holds that $\Pr[X \geq 1] \leq E[X]$. Now, in general, $X$ depends on some parameter that tends to infinity. The first moment method uses Markov's inequality to assert the following: if we find some condition that makes $E[X]$ converge to 0, then the probability that the random variable $X$ assumes a nonzero value tends to zero too, whenever the condition holds. Despite its simplicity, the first moment method is a powerful and simple to apply tool that quickly provides us with some condition (most often not the tightest possible) for proving that some random variable almost certainly is zero, asymptotically.

The connection of the first moment method with the unsatisfiability threshold conjecture was observed by a number of researchers, including Franco and Paull [20], Simon et al. [50], Chvátal and Szemerédi [11]. The argument, in few words is the following: let $\phi$ be a random 3-SAT formula and $X$ the random variable that counts the number of satisfying truth assignments of $\phi$. Then the probability that a truth assignment satisfies $\phi$ is equal to $(\frac{3}{2})^m$, where $m = \sum_n$ is the number of selected clauses. Therefore, the expected number of truth assignments that satisfy a random formula $\phi$ is $2^m \left(\frac{3}{2}\right)^m$. Since $\Pr[\phi \text{ is satisfiable}] = \Pr[X \geq 1]$, from Markov's
inequality we get that

\[ \Pr[\phi \text{ is satisfiable}] \leq E[X] = 2^n \left(\frac{7}{8}\right)^r. \quad (3) \]

If by \( r_M \) we denote the exact solution of the equation \( 2^r \left(\frac{7}{8}\right)^r = 1 \), where \( r_M \approx 5.1909 \), then we observe that under the condition \( r > r_M \), the right-hand side of Equality (3) tends to zero, which establishes the value \( r_M \approx 5.19 \) as an upper bound for the unsatisfiability threshold (as it was done in Section 1).

Now, the first observation that the condition \( r > r_M \) is not the best possible, came from Broder, Frieze, and Upfal in [6], where they showed that the condition \( r > r_M - 10^{-7} \) suffices to guarantee that \( \Pr[\phi \text{ is satisfiable}] \) tends to zero.

El Maftouhi and Fernandez de la Vega obtained a further improvement, by showing in [17] that the condition can be relaxed to \( r > 5.08 \). Kamath, Motwani, Palem, and Spirakis in [29] obtained the improved condition \( r > 4.758 \) using a numerical computation while they also gave an analytic proof of the condition \( r > 4.87 \). We will stay, for a moment, at their result because they were the first to observe the main weakness of Markov's inequality and point out a way to overcome it. More specifically, they remarked that, according to (3), when \( r \in (4.2, 5.19) \) the expectation of the number of satisfying truth assignments of formulas with \( rn \) clauses becomes unboundedly large while, at the same time, experimental evidence shows that satisfiable formulas with that many clauses become increasingly rare. The only plausible explanation of this "contradictory" phenomenon is that, indeed, when \( r \in (4.2, 5.19) \) the probability that a formula is satisfiable vanishes but this fact cannot be deduced from Markov's inequality because there still exist rare formulas with a huge number of satisfying truth assignments. These rare formulas contribute so much to the expectation in (3), that it is impossible for this expectation to vanish when \( r \in (4.2, 5.19) \). The authors in [29] provided a beautiful mathematical proof supporting this explanation in the following way: they showed that any satisfiable formula contains a small set of variables such that if set properly, then all the clauses of the formula are satisfied. These variables form a cover set in the sense that setting them in a specific way suffices to make the formula satisfiable with no "help" from the rest of the variables. But this means that the remaining variables can be set freely to any truth value without making the formula unsatisfiable. This offers to the formula \( 2^n \) satisfying truth assignments in all and since the expected value of \( i \) is \( \Theta(n) \) (see the analysis in [29]), the formula is expected to have an exponentially large number of satisfying truth assignments. Therefore, to come back to the explanation we considered above, for \( r \in (4.2, 5.19) \) it is true that unsatisfiable formulas abound but the very few satisfiable formulas that exist, contribute to the expectation involved in Markov's inequality an exponential number of truth assignments. To continue the outline of the argument in [29], let \( A \) be any truth assignment and \( \Phi_A \) the set of formulas with \( rn \) clauses that are satisfied by \( A \). Also, for a formula \( \phi \), we denote by \#\( \phi \) the number of truth assignments that satisfy it. The authors of [29] show that the following "harmonic mean" formula holds:

\[ \Pr[\phi \text{ is satisfiable}] = 2^n \left(\frac{7}{8}\right)^r \sum_{\phi \in \Phi_A} \frac{1}{\#\phi} \frac{1}{|\Phi_A|}. \quad (4) \]

It is not difficult to see that the expression in (4) is the expectation of the inverse (that is why we call it "harmonic" mean) of the number of satisfying truth assignments of the
formulas satisfied by the particular truth assignment \( A \). The choice of \( A \) is immaterial however because of symmetry properties of logical formulas. We observe that Equation (4) attempts to exploit rare formulas with many satisfying truth assignments by having their contribution multiplied by \( \frac{1}{\#\phi} \). This has as an effect the following: the more satisfying truth assignments a formula has, the less it contributes to the expectation in (4) that bounds from above the probability that \( \phi \) is satisfiable. This behaviour of the “harmonic” mean is in sharp contrast to Markov’s inequality where the contribution of formulas with many satisfying truth assignments is significant. Starting with (4), the authors in [29] use some techniques that make use of the sharp probability bounds they have derived earlier in the same paper for the occupancy problem in random allocation schemes of balls into bins, in order to bound from above the “harmonic” expectation in (4). Their arguments are elegant but lengthy and we will not reproduce them here.

The section to follow, will be devoted to the presentation of a method proposed in [36] that attempts to reduce the effect of rare formulas with many satisfying truth assignments, by isolating from among these truth assignments only the ones that are locally maximum in a certain, well-defined sense. Then only these truth assignments are used in bounding from above the probability that a random formula is satisfiable thus overcoming, to some extent, the deficiency of Markov’s inequality. We close the section by describing briefly some other works that are based on the concept of local maximality and reduce further the upper bound for the unsatisfiability threshold.

3.1 The method of local maxima

In the previous section, we saw how the first moment method is applied in order to establish rigorously that the unsatisfiability threshold is less than 5.19. We observed that this upper bound is far from the experimentally determined threshold value because Markov’s inequality does not have the power to take into account the following fact: as the number of variables \( n \) tends to infinity and a random formula is formed by selecting \( m = r n \) clauses with replacement, there exist formulas that are highly improbable to be formed yet they contribute to the expectation of the number of solutions of a random formula significantly, thus increasing the right-hand side of Markov’s inequality. This, in turn, has as an effect that the right-hand side needs unnecessarily large values for \( r \) to become negligible. Something analogous, but in a more real-life situation, occurs in lottery games. There, it is obvious that as the number of players grows, the probability of one individual winning becomes negligible. However, the expectation of the amount the individual will win remains high.

The above considerations lead us naturally to the following two possible approaches in order to address the weakness of Markov’s inequality: we can either single out some of the formulas that are unlikely to occur and disregard their contribution to the expectation of the number of solutions or reduce somehow the cardinality of the class of satisfying truth assignments so that each satisfiable random formula contributes to the expectation less. In the remaining of this section, we will describe a technique that follows the second approach.

More formally, let \( S_n \) be the class of all \( 2^n \) truth assignments and \( A_n \) the random class of truth assignments that satisfy a random formula \( \phi \). From now on, we will denote the fact that a truth assignment \( A \) satisfies a formula \( \phi \) either by \( A \models \phi \) or by \( A \in A_n \).
Since the deficiency of Markov's inequality can be attributed to rare formulas having a large number of satisfying truth assignments, a plausible approach is to replace $\mathcal{A}_n$ with a class of smaller cardinality in order to use the expectation of this new, smaller class as an upper bound to the probability that $\phi$ is satisfiable. This, in turn, will lead to an improved upper bound for the unsatisfiability threshold since the right-hand side of the new version of Markov's inequality will be now smaller. There is one critical issue though. We have to ensure that the expectation of the new random class still bounds from above the probability that $\phi$ is satisfiable. We will see more on this issue below.

The idea we are going to describe is quite simple. Take a random formula $\phi$ and consider the set $\mathcal{A}_n$ of truth assignments that satisfy it. If we view the truth assignments as strings composed of zeros and ones (the values assigned to the $n$ variables), then a lexicographic (dictionary) ordering is induced among the 0-1 strings of the set $\mathcal{A}_n$. According to this ordering, the value 0 or FALSE is smaller than the value 1 or TRUE and the values of variables with higher index within the truth assignment are less significant.

**Definition 3** Given a random formula $\phi$, take assignments $A$ such that the following two conditions hold:

- $A$ satisfies $\phi$, and
- any assignment obtained from $A$ by changing a single FALSE value to TRUE does not satisfy $\phi$. Such a change is called a sin.

Now according to the total ordering of the truth assignments, the class $\mathcal{A}_n^1$ contains the elements of $\mathcal{A}_n$ that are local maxima within the set of all satisfying truth assignments differing in one bit position. In other words, a truth assignment belongs to $\mathcal{A}_n^1$ if it satisfies $\phi$ and if it is impossible to obtain from it a satisfying truth assignment by changing a single FALSE value to TRUE.

Since $\mathcal{A}_n^1 \subseteq \mathcal{A}_n$, it is true that $E[|\mathcal{A}_n^1|] \leq E[|\mathcal{A}_n|]$. Therefore, in order to relax Markov's inequality we only need to establish that the expectation of $\mathcal{A}_n^1$ still bounds from above $Pr[\phi$ is satisfiable]. Before we continue, we will fix some notation. Given a truth assignment $A$, we will denote by $sf(A)$ the number of possible single flips it contains which is, of course, equal to the number of zeros in $A$. Also if $sf$ is some single flip, by $A^f$ we will denote the truth assignment that results from $A$ after the application of the single flip.

**Lemma 3** The following equality is true:

$$Pr[\phi \text{ is satisfiable}] = Pr[|\mathcal{A}_n^1| \geq 1].$$

**Proof** It suffices to prove that $\phi$ is satisfiable iff $|\mathcal{A}_n^1| \geq 1$. Trivially, if $|\mathcal{A}_n^1| \geq 1$, then there exists a truth assignment that satisfies $\phi$. Therefore, from the first branch of the definition, $\phi$ is satisfiable.

Suppose now that $\phi$ is satisfiable. Then there exists a truth assignment $A$ such that $A \models \phi$. Now if $A$ satisfies Definition 3, then $A \in \mathcal{A}_n^1$ and, so, $|\mathcal{A}_n^1| \geq 1$. If, however, $A \notin \mathcal{A}_n^1$, then there...
exists a single flip $s^f$ such that $A^s^f \models \phi$, i.e. the second branch of Definition 3 is violated. Let us now consider $A^{s^f}$. If this truth assignment still violates Definition 3, we continue applying single flips until at some point we reach a truth assignment that satisfies the definition. And we are guaranteed to reach such an assignment because, in the worst case, the application of flips will lead to the all-ones assignment that vacuously satisfies Definition 3.

Applying Markov's inequality to $|A_n^1|$, we obtain that $\Pr[|A_n^1| \geq 1] \leq E[|A_n^1|]$ and, consequently, from Lemma 3 it is easy to deduce the following inequality that "refines" Inequality (3):

$$\Pr[\phi \text{ is satisfiable}] = \Pr[|A_n^1| \geq 1] \leq E[|A_n^1|].$$

We can now prove the following theorem:

**Theorem 3** The expectation of the random class $|A_n^1|$ is at most $(7/8)^n(2 - e^{-3\sqrt{7}} + o(1))^n$. Therefore, the unique positive solution of the equation $(7/8)^n(2 - e^{-3\sqrt{7}}) = 1$ is an upper bound for the unsatisfiability threshold. (This solution is less than 4.667.)

**Proof** The expected value of the random variable $|A_n^1|$ is given by the formula

$$E[|A_n^1|] = (7/8)^n \sum_{A \in A_n} \Pr[A \in A_n^1 | A \in A_n].$$

To see this, observe that the random variable $|A_n^1|$ is the sum of indicator variables that assume the value 1 when $A \in A_n^1$ under the condition $A \models \phi$. Therefore, from (6), in order to compute the expectation of $|A_n^1|$, it suffices to compute $\Pr[A \in A_n^1 | A \in A_n]$ or, equivalently

$$\Pr[\forall s^f, A^{s^f} \not\models \phi | A \in A_n].$$

First, we observe that the only effect of the conditional $A \in A_n$ is that it excludes from the set of the $8{n \choose 3}$ clauses that $\phi$ may contain, the $n \choose 3$ clauses not satisfied by $A$. Therefore, when $\phi$ is formed, its conjuncts are assumed to belong to this reduced set of $7{n \choose 3}$ clauses.

From (7), we are required to compute the probability of a conjunction of events. We will, instead, compute the probability of one of these events and observe that we can bound the probability of their intersection from above with the product of probabilities of each of the events.

Let $s_{f_0}$ be one of the possible single flips of $A$, i.e. one of its FALSE values. Assume, also, that this single flip changes from FALSE to TRUE the value of the variable $x_i$. For the event $\Pr[A^{s_{f_0}} \not\models \phi]$ to be true, it is necessary and sufficient that $\phi$ contains at least one clause of the form $(x_i, x, y)$, where $x, y$ are two literals different from $x_i, \neg x_i$, that evaluate to FALSE under $A$. We call this set of clauses blocking clauses of the flip $s_{f_0}$. The number of possible such clauses is $\binom{n-1}{2}$ and for the event $A^{s_{f_0}} \not\models \phi$ to be true, under the condition $A \in A_n$, it is necessary and sufficient that during the selection of the $n = rn$ clauses of $\phi$ from the set of $7{n \choose 3}$ clauses, at least one of the $\binom{n-1}{2}$ blocking clauses of $s_{f_0}$ is selected. In numbers

$$\Pr[A^{s_{f_0}} \not\models \phi] = 1 - \left(1 - \frac{n-1}{7{n \choose 3}}\right)^m = 1 - e^{-3\sqrt{7}} + o(1).$$

Now to get to the joint probability involving all $s^f(A)$ single flips, we first observe that the sets of blocking clauses of distinct single flips are disjoint. However, this is not sufficient to
justify the multiplication of the corresponding probabilities we computed in (8) in order to compute the probability in (7). This is because the events $A^{sf_1} \not\models \phi$, $sf$ a single flip of $A$, are not independent although the sets of clauses that make them true are disjoint. To see this, we observe that once a clause is selected that realizes the event $A^{sf_0} \not\models \phi$ for some single flip $sf_0$, then this clause holds a position in $\phi$, thus reducing the probability of another event $A^{sf_1} \not\models \phi$ being true for another single flip $sf_1$. Therefore, the events $A^{sf} \not\models \phi$, $sf$ a single flip of $A$, are negatively correlated and we can use the product of the probabilities of the events in the conjunction in (7) to bound from above the probability of the conjunction. The fact that the events $A^{sf} \not\models \phi$ are negatively correlated, can be formally proved using the main theorem in [U].

Since the probability in (8) does not depend on the particular single flip, we have the following:

$$\Pr[\forall s, f. A^{sf} \not\models \phi \mid A \in A_n] \leq (1 - e^{-3r/l} + o(1))^{s f (A)}.$$  

(9)

And since the above probability depends only on the number of zeros in a truth assignment, we may group the truth assignments accordingly and use Newton's binomial theorem to obtain from (6) the following:

$$E[|A_n^{A}|] = \left(\frac{7}{8}\right)^r \sum_{A \in A_n} \Pr[A \in A_n^A \mid A \in A_n]$$

$$= \left(\frac{7}{8}\right)^r \sum_{k=0}^{n} \binom{n}{k} \Pr[A \in A_n^A (A \text{ with } k \text{ zeros }) \mid A \in A_n]$$

$$= \left(\frac{7}{8}\right)^r \sum_{k=0}^{n} \binom{n}{k} \left(1 - e^{-\frac{3r}{l}} + o(1)\right)^k$$

$$= \left[\left(\frac{7}{8}\right)^r \left(2 - e^{-\frac{3r}{l}} + o(1)\right)\right]^n.$$  

(10)

But $E[|A_n^{A}|]$ in (10) vanishes for values of $r$ larger than the unique positive solution of the equation $(7/8)^r (2 - e^{-3r/l}) = 1$. By Inequality (5), this solution is an upper bound for the unsatisfiability threshold. Using the symbolic and numerical computation package Maple (see, e.g., [45] for a description of the package and its capabilities), we found that the solution is less than 4.667, which establishes this value as an upper bound for the unsatisfiability threshold.

Armed with the successful application of the method of locally maximum satisfying truth assignments, it is natural to wonder how far we can get by defining even more restricted classes of truth assignments. Let us consider locally maximum truth assignments within a neighbourhood of two bit positions ([36]). We define as a double flip the change of exactly two variables $x_i$ and $x_j$ where $x_i$ is changed from FALSE to TRUE and $x_j$ from TRUE to FALSE. We also impose the restriction $i < j$, so that a double flip always leads to a lexicographically larger assignment. By $A^{df}$ we will denote the truth assignment that results from $A$ after the application of the double flip $df$.

Let $A_n^{df}$ be the set of truth assignments $A$ that have the following three properties:

- $A \models \phi$,
for all possible single flips $s f$ of $A$, it holds $A^{sf} \not \models \phi$, and

- for all possible double flips $df$ of $A$, it holds $A^{df} \not \models \phi$.

Before we continue, observe that our double flips do not include changes of values that are both FALSE although these changes still produce lexicographically larger assignments. The reason is that this kind of flip can be proved to have a negligible contribution towards improving Markov's inequality, therefore its omission is of no importance for our purposes.

It can be proved (see [36]), that the following inequality holds:

**Lemma 4**

\[
\Pr[\phi \text{ is satisfiable}] \leq E[|A_n^{2q}|] = \sum_{A \in S} \Pr[A \in A_n^{2q}, A \models A, A \models \phi] = (7/8)^n \sum_{A \in S} \Pr[A \in A_n^{2q} | A \models \phi] = \sum_{A \in S} \Pr[A \in A_n^1 | A \models \phi] \cdot \Pr[A \in A_n^{2q} | A \models A].
\]  

(11)

On the face of it, our task seems easy again. In order to find an upper bound for the unsatisfiability threshold, it suffices to find the smallest possible value for $r$ for which the right-hand side of (11) tends to 0. Moreover, we have already computed one of the probabilities in (11) when we considered the single flips case. It turns out, however, that the double flips case is not as easy. The required computations are described in full detail in [36] and their length does not allow us to reproduce them here. We will only confine ourselves to say that, rather interestingly, (11) can be rewritten as

\[
\Pr[\phi \text{ is satisfiable}] \leq 3m^{1/2}(7/8)^n \sum_{A \in S} X(s f(A)) Y(A)
\]

(12)

where

\[
X = \Pr[A \in A_n^1 | A \models \phi], \quad \text{and}
\]

\[
Y = 1 + \frac{z}{n}, \quad \text{for some negative constant } z,
\]

with \( \binom{n}{k} \), the \textit{q-binomial} or Gaussian coefficients (see [24, 39]). The main difficulty in the derivation of (12) was that the dependencies among distinct double flips were not as simple as the dependencies among single flips that we saw in the first part of this section. In order to compute probabilities under such dependencies (that lead to the expression for $Y$ above), the authors in [36] employed a version of Suen's correlation inequality (see [52]) that was proved by Janson in [27].

With (12), we also enter the world of \textit{q-hypergeometric} series, which is the name of the sum in (12). Now our success in finding a good upper bound for the unsatisfiability threshold relies on our success in finding a good, manageable upper bound to this sum. This is exactly what was done in [36], where a closed form upper bound to (12) was given (see, also, [34]).
that was later improved in [28] by Janson, Stamatiou, and Varnvakari. This upper bound improvement for the \( q \)-binomial coefficients, resulted in the establishment of the value 4.596 as an upper bound to the unsatisfiability threshold. This value was achieved through two different approaches that may be of interest on their own right. In the first approach, the summation that appears in the first line of 12 was rewritten in the form of the partition function of a physical system of \( n \) spin particles, each spin assuming the values 0 or 1. Then an asymptotic expression was obtained for the partition function, using an optimization technique commonly used in physics. The second approach, started by rewriting the same sum as an expression that defines the \( q \)-hypergeometric series (also known as Rogers-Szego polynomials). Using the Eulerian generating function of this series, an upper bound was obtained that led to the same improvement as the first approach. Finally, in [30], Kaporis, Kirousis, Stamatiou, Varnvakari, and Zito obtained the value 4.571 using the tight probability computations obtained in [29] for the occupancy problem in random allocation schemes of balls into bins, in conjunction with the establishment of a new upper bound for the \( q \)-binomial coefficients.

The technique of locally maximum satisfying truth assignments, that gave rise to the works and the series of improvements mentioned above, can be generalized, at least in theory, to any degree of locality. We may define triple, or even higher-order flips, each time obtaining a tighter Markov-type inequality and, thus, a better upper bound for the unsatisfiability threshold. The general inequality is the following:

\[ \Pr[\phi \text{ is satisfiable}] \leq \mathbf{E}[\vert \mathcal{A}_n \vert] \]

where

\[
\mathbf{E}[\vert \mathcal{A}_n \vert] = \left( \frac{7}{8} \right)^n \sum_{A \in \mathcal{A}_n} \Pr[A \in \mathcal{A}_n ^1 \mid A \models \phi] \cdot \Pr[A \in \mathcal{A}_n ^2 \mid A \in \mathcal{A}_n ^0] \cdot \ldots \cdot \Pr[A \in \mathcal{A}_n ^n \mid A \in \mathcal{A}_n ^{n-1}].
\]

However, the reader is warned that the probability calculations become daunting and very cumbersome to handle, even for the triple-flips case.

4 Phase transitions in physical systems

Let us now change context and move from Mathematics and Boolean formulas to Physics and real world. Nature abounds in examples where a substance may exhibit different properties depending on the conditions it finds itself in. Consider a simplified model of the familiar magnet. Magnets can thought of as substances that consist of particles characterized by their spin. Spin determines the orientation of each individual particle’s magnetization and can take two values, “up” or “down”. When the spins are of this “binary” form, we call them Ising spins. Let us denote by \( S_i \) the \( i \)th spin, where \( S_i \) may be equal to +1 or -1 depending of whether the orientation of the spin is “up” or “down” respectively. Also among each pair of neighbouring spins, there exists an interaction force: the spins of the pair have the tendency to “push” each other towards the same or the opposite direction. In general, this tendency may have different values for different pairs of atoms. Let us denote by \( J_{i,j} \) the strength of this tendency for pair \((i,j)\). If \( J_{i,j} \) is positive, we say that the spins \( i \) and \( j \) tend to align ferromagnetically (to have
the same spins) while if \( J_{i,j} \) is negative the spins tend to align antiferromagnetically (to have opposite spins). The corresponding Ising Hamiltonian, that represents the energy of the Ising spin system, is

\[
H = - \sum_{i,j} J_{i,j} S_i S_j.
\]

If all \( J_{i,j} \)'s are the same and are greater than 0, we have a model for a pure Ising ferromagnet. If all \( J_{i,j} \)'s are the same and are less than 0, we have a pure Ising antiferromagnet. Let us consider the case of an Ising ferromagnet with all \( J_{i,j} \)'s equal to \( J \). Now whether this kind of magnet will display or not magnetic properties, and if so in what orientation, depends on the collective behavior of the spins that compose it. Consider that initially it is subjected to a high, compared to the strength of \( J \), temperature. Then the thermal agitation, will result in random orientations of the spins, overrunning the alignment tendency between neighbouring pairs. That is, for each individual spin, there is no preferred spin direction and alignment with nearby spins: each spin fluctuates rapidly, and equally, between the “up” and “down” states. The overall result of this randomness is that the substance displays no magnetization properties, e.g. it cannot attract (or repel) little pieces of iron. The substance is then in the paramagnetic phase.

Now if we start lowering the temperature there is a certain point \( T_c \) (the Curie point) that depends on \( J \), where the exchange interactions come into play. Correlations among even very distant spins appear and the net result of this is that most of the spins are “frozen” in either the “up” or “down” state and since the vast majority of the atoms point to the same direction, magnetization emerges. We say that there is a spontaneous magnetization, since it arises without the intervening action of any external magnetic field, that could orient the spins towards the same direction. (For a detailed account of such transitions in magnetic materials as well as other substances, see any book on statistical mechanics or thermodynamics e.g. [18, 54, 55]. Also, a very good survey article is [19].) Now magnetization is an importance parameter since it signals the onset of a phase transition from the “non-ordered” phase (essentially a “mixture” of the ferromagnetic and antiferromagnetic phases) to the ordered phase of uniform spin direction. System parameters that behave in a drastically different way as a system conditioning variable (e.g. temperature) crosses a certain point (e.g. \( T_c \), are called order parameters. Continuing our discussion, as the temperature decreases towards 0, the spins remain locked in the state they acquired during the passage from \( T_c \) (because of the exchange interactions) and the magnetic properties of the substance are preserved. We say that the system has settled in a minimum energy state or, as it is called, a Ground State. If you look at the formula for the energy of the Ising model, you will see that the energy is minimized if all the spins point to the same direction and that the energy remains invariant under global spin flips. That is, if all spins flip their orientation simultaneously, the energy remains the same. This implies that there can only exist two Ground States: the “all spins up” and the “all spins down” states. Furthermore, on the Curie point, there is also an interesting behavior of the substance regarding its sensitivity to externally applied weak magnetic fields – its magnetic susceptibility. In high temperatures above \( T_c \), the field has no significant effect on the magnetization properties of the substance because the spin fluctuations prevail over the effort of the field to orient them all towards the same direction. Likewise, in low temperatures below \( T_c \), the field has also not significant effect on the, now, well established uniform magnetization direction, because of the strong interactions between the aligned pairs of spins. However, during the transition over \( T_c \) the
substance’s spins are extremely sensitive even to very small external fields. A graph of the susceptibility near the Curie temperature clearly shows a divergence above $T_c$ (considering that no field is externally applied: $H = 0$) similar to the one that is shown in the Figure 5 (for a detailed analysis of this phase transition see [55]):

![Graph of susceptibility](image)

**Figure 5: Susceptibility**

And even when a non-zero field is applied, the peak of the substance’s susceptibility is observed at $T_c$. Let us now consider another Ising spin system that shares many interesting properties with the pure Ising spin model we have presented and it is more relevant to our purposes: the Spin Glass. Back in the seventies, it was experimentally observed that some solid metallic alloys containing thin concentrations of some magnetic metal like Manganese (Mn) into a non-magnetic metallic host material (which comprises the bulk of the alloy) like copper (Cu) exhibited a strange behavior if they were heated around certain temperatures. One aspect of their behavior is that as the temperature tends to 0, no long-range ferromagnetic or antiferromagnetic state is observed on the alloy, as it was observed on pure, ferromagnetic or antiferromagnetic materials. In such alloys, where there are few spins compared to the concentration of the non-magnetic host, the interaction between neighbouring atoms of the magnetic material is essentially random, in the sense that it fluctuates in magnitude (i.e. strength) and sign (i.e. tendency to align the atom spins so as to point in the same or the opposite direction) according to the distance between the atoms. This has an effect that for each a pair of neighbouring atoms the interaction that exists between them may tend to align them either in the same direction (ferromagnetically) or in opposite directions (antiferromagnetically) and, what is more, this cannot be preset (it is decided at “random”). The Hamiltonian is the same as it was in the ferromagnet’s case. A simplified spin glass model, that suffices for illustration purposes, is a 2-dimensional mesh with nearest neighbour local interactions. In the Figure 6 we see an Ising spin system where the spins are aligned either “up” or “down”. A continuous line that connects a pair of spins, model an antiferromagnetic exchange interaction while a dashed line models a ferromagnetic one. If we assume that the exchange interactions are given (e.g. they have been formed during the process of physically creating the spin glass) then we observe that given the orientation of the rest of the spins, the
right lower corner spin cannot choose an orientation so as to satisfy the exchange interaction constraint imposed by its two neighbours.

Figure 6: A simplified model of spin glasses

This important phenomenon is called frustration and it is a consequence of the exchange interaction randomness that characterizes spin glasses. These systems have been subjected to a variety of experimental studies and have displayed a wide variety of characteristic properties. Unlike ferromagnetic systems, which we studied first in this section, they display a cusp (right angle) in the susceptibility rather than a divergence, at a certain temperature. Then, as the temperature is reduced towards 0, the system shows no ferromagnetic or paramagnetic properties. This means that it does not tend to a “uniform” or ordered ground state, like the one with all spins up or down for a ferromagnet. Of course since scientists are not merely satisfied with observing and cataloguing interesting phenomena, what was called for was a theory that could account for these strange observations. After all, many of the greatest and deepest theories in physics were the result of some strange experimental observations that led scientists to abandon long-standing classical theories and perceptions about reality.

In a few words, a fairly accurate explanation of the properties of spin glasses is the following: as long as the temperature is high, each spin oscillates rapidly and equally between the states +1 and -1. We say that, on the average, the spin is in the “state” 0 (the mean value of the spin, that is). This is also true in the case of a ferromagnet. Now as the temperature is lowered, short range interactions start to gain strength among nearby spins. This happens because as the temperature decreases, the exchange interactions start to take command and affect the spins accordingly. Remember that these interactions can be either ferromagnetic or antiferromagnetic. Then large clusters may form that contain ferromagnetically aligned spins that give rise to some localized magnetization. Or there may exist clusters containing a mixture of ferromagnetically and antiferromagnetically aligned spins. This cluster formation is a consequence of the randomness of the exchange interactions. Now as the temperature continues to decrease, there is a certain value, called the freezing temperature ($T_f$) near which
the spin correlations spread rapidly over larger and larger areas. As the temperature approaches 0, the spin system must settle on a state of lowest energy, a ground state. In the paramagnet case, that was easily decided and the choice was limited: either all spins “up” or all spins “down”. But now the situation is not that easy. A lowest energy state is far from being so simple and ordered. It turns out that such a “frozen” state is highly disordered and spins may point to any direction. And what is more, there may, now, exist many spin configurations that have the lowest, or near lowest, energy. This is a remarkable characteristic of spin glasses: the spin configurations landscape is “rough” and contains many nearest ground state configurations that are separated by high energy barriers. This last property means that once the spin system has “locked” on a particular configuration, it cannot easily escape. This “rough landscape” picture has a direct analogue in computational complexity of the k-SAT problem we examined earlier: near the threshold point there are many would-be solutions, that appear to satisfy more and more of the clauses, only to be discovered near the end of the search that they are really not solutions. Therefore, the ground state of a spin glass resembles the “ground state” (zero unsatisfiable clauses) of a Boolean formula. Further details as well as an extensive historical account of the theoretical and experimental investigations in the spin glass phenomenon, can be found in [43, 48]. Before we end this section, let us offer an explanation for the “glass” characterization. Of course, spin glasses are not “glasses” in the every day sense of the word. They resemble glasses in the sense that in a way similar to the way glass atoms are immobilized (frozen) in random positions of the space, the magnetic spins of the atoms in spin glasses become fixed in random orientation as the material is subjected to temperatures below a critical value. To make another analogy, crystals are contrasted to glasses in that their atoms are arranged in a regular lattice structure. Likewise, pure magnets (like the ones we are accustomed to use) are contrasted to spin glasses in that the spins of their atoms are oriented towards the same direction, giving rise to magnetism. In the next section we will finally see the connection between spin glasses and k-SAT.

4.1 Random formulas as Ising Spin Glasses

And now it is time see how the two different phase transition phenomena we have described can be brought under the same perspective. Let us take a random k-SAT formula \( f \) with \( m \) clauses over \( n \) variables. We will map (see [46]) the random clause onto a problem involving the energy of an Ising spin system with random interactions, like the one we considered in the previous section. For a truth assignment \( S \) and each variable \( x_i \) we create a spin variable \( S_i \) with value +1 if \( x_i \) is “true” and value -1 if \( x_i \) is “false”. The clauses are represented by an \( m \times n \) random matrix \( A \) where \( A_{i,j} = 1 \) if the \( l \)-th clause of the formula contains the literal \( x_i \) and \( A_{i,j} = -1 \) if the \( l \)-th clause contains the literal \( \neg x_i \). Otherwise, \( A_{i,j} = 0 \). Now consider the following function:

\[
E[\Delta, S] = \sum_{l=1}^{m} \delta(\sum_{i=1}^{n} A_{l,i}S_i, -k)
\]

where \( \delta(x, y) = 1 \) if \( x = y \) and \( \delta(x, y) = 0 \) otherwise. In addition, to ensure that the number of literals in each clause is exactly equal to \( k \), we introduce the following constraints:

\[
\sum_{i=1}^{n} A_{l,i}^2 = k, \text{ for each } l = 1, \ldots, m.
\]
Now $k$ has a nice interpretation as a correlation range: it is the number of coupled spins within a each clause. Now notice that if a $k$-SAT formula $f$ is transformed in the way we just described, then given a truth assignment (which is represented in a spin orientation) the value of $E[Δ, S]$ is equal to the number of unsatisfiable clauses, under this assignment. Therefore, $f$ is unsatisfiable if and only if $E[Δ, S] = 0$. Up to now, this may seem just like a nice alternative way to express a random Boolean formula and its satisfiability properties. But in doing so, we have also succeeded in transforming the original formula into some kind of mysterious magnetic substance that is amenable to investigation using the methods and models available for dealing with real substances! In [46], a suitable order parameter for transformed Boolean formulas is defined, and its behavior as a function of the $m/n$ ratio is examined. With the aid of tools and arguments of statistical mechanics, it is shown that a phase transition is indeed exhibited around the experimentally determined threshold values. Now $k$-SAT may be attacked in a completely different way, using the understanding, models and mathematical tools developed to deal with Spin Glass systems. We will not go into the details of how this is done, because it is beyond the introductory scope of this writing. However, the interested reader may find the relevant analysis in [46]. Of relevance to the statistical mechanics analysis of $k$-SAT are also [3] and [47], where the $(2 + p)$ problem is analyzed (a version of satisfiability allowing formulas containing a mixture of clauses with 2 and 3 literals), as well as [53]. A popular account of all the things we discussed in this article may be found in [26].

5 Some thoughts and conclusions

Although it still remains a tantalizingly open question as to what is the cause that gives rise to computationally intractable problems, the efforts for its resolution have never ceased. Since the beginning of the 70's, where the notion of intractability was formalized by the seminal papers of Cook [12], Karp [32], and Levin [40] the efforts for understanding complexity have led to the development of important mathematical theories that approach the problem from many different perspectives. But this time, through a new, innovative perspective, the problem of complexity can be linked to real world, physical phenomena and their behavior. We took as our generic example of complexity the "archetypal" intractable problem: the problem of testing a Boolean formula for satisfiability. We saw that an instance of this problem can be mapped onto a physical system that appears to have some similarities with this problem. Perhaps $k$-SAT is intractable because a Spin Glass has a complex behavior. But the interaction between physics and mathematics is two-way. Perhaps Spin Glasses behave the way they do because their behavior resembles the behavior of a computationally intractable problem. Or, perhaps, Spin Glasses and Intractable problems are simply two sides of the same coin: behind both of them, there lies a deeper and universal complexity concept that governs the behavior of both.

Acknowledgement

We would like to thank the anonymous referee for his/her invaluable contribution towards the improvement of the presentation.
References


Dear Reader,

EdMat goes ICALP. When the workshop coordinator of ICALP 2003 asked me to organize an Education Matters panel discussion, I accepted with pleasure. I am happy to announce the following event.

Education Matters –
The Challenge of Teaching Theoretical Computer Science

Around the world, courses on theoretical issues and on formal methods in general are often not very popular among the computer science students. Many of them do not see the significance of formal concepts and theoretical results and have difficulties to understand and to use the underlying machinery. This is the challenge of teaching theoretical computer science. On one hand, the topic is important, but not easy to explain. On the other hand, a good part of the audience may be hesitant, sceptic, reserved, or ignorant.

Four panelists will consider what can be done about this. In their initial statements they will address the questions along the following lines, after which there will be an open plenary discussion.

- Is there a didactic way out?
- Do good lectures help?
- Does a good textbook solve the problem?
- May digital tools bring the breakthrough?
- Or will the use of multimedia revolutionize the education?

Panelists: Jan Friso Groote (Eindhoven), Jozef Gruska (Brno), Jetty Kleijn (Leiden), and Markus Roggenbach (Swansea).

The Education Matters panel discussion at ICALP 2003 in Eindhoven will take place on
Sunday, 29 June 2003, 16:00-17:30

after the pre-conference workshops. Everybody interested is invited.

The main contribution to this issue of the column is a proposal by Henning Fernau how to teach Cook's theorem on the NP completeness of the satisfiability problem in an introductory course on the theory of computation. The key idea is to break down the proof in a series of separate reductions, each of which exemplifying the basic principle in a nicely presentable form. I wish a good reading.

More about education matters can be found on the EdMat webpage

http://www.informatik.uni-bremen.de/theorie/EducationMatters.

If you want to contribute to the column, you may use the submission page on the webpage or send your contributions preferably in electronic form to

Hans-Jörg Kreowski
Fachbereich Mathematik und Informatik
Universität Bremen
Postfach 33 04 40
D-28334 Bremen
Email: kreo@informatik.uni-bremen.de
Fax: +49-421-218-4322
http://www.informatik.uni-bremen.de/theorie
Abstract

In nowadays computer science curricula, many students will only take one course dedicated to the theory of computation. Since for many pedagogical reasons, such an introductory course will put the complexity theory part at its very end, many students find it hard to follow the reduction argument of the proof of the basic theorem of Cook. Here, we mainly give a proof of Cook’s theorem broken into a couple of lemmas. The main advantage of this approach is that this way students will find much more examples of reduction arguments than they will usually encounter in such a course, and each of the reduction arguments is presentable on one or two slides.

1 Introduction

The main purpose of this paper is to stir up a discussion, possibly to be continued in a newsgroup as comp.theory, on how to introduce basic notions and results of complexity theory, especially the notion of a reduction and Cook’s theorem, to undergraduate students.

My personal teaching experience comes from a German and Australian background, but it is my strong feeling that similar problems arise all over the world.

What are the particular problems that we, as teachers of a course COMP Theo, typically face?

Reduced mathematical background. Many computer science curricula (and even more, this applies to related subjects as software engineering or information technology) have largely reduced the amount of mathematics students have to some minimum. This makes COMP Theo for many students their second
encounter with mathematical thinking and mathematical argumentation within their whole career. If the obligatory mathematics course is more taught in a recipe style, then COMPTHEO might even be the students' very first encounter with mathematical proofs at all.

While it is well possible to question this tendency of reducing the mathematical education of computer science students to a minimum, I am not going in this direction here. Rather, I would like to discuss how to cope with this situation as such. This approach should also be welcomed by teachers who (still) find themselves in the lucky situation that they can rely on certain mathematical skills of their students. After all, also in the more mathematics-oriented curricula, many students find it hard to follow the arguments of reduction proofs, let alone the problem of finding reductions themselves.

There is another aspect here: for many students, COMPTHEO will remain the only course on issues of theoretical computer science they ever take. This puts further responsibility on us as teachers, since material regarding theoretical issues in computer science that students don't learn with us they will probably never learn.

Contents of CompTheo. The typical material covered by an introductory course like COMPTHEO is:

**formal languages** This part of the course mostly focusses on regular and context-free languages. It can be well-motivated by everyday experience of computer science students, working with tools like egrep or writing Document Type Declarations for XML documents. From a mathematical point of view, this might be the place in the curriculum where students are taught how to use the induction principle. This part will also prepare them for a future course on compiler construction which they might wish to take, although compiler construction tends to be optional in nowadays curricula.

**recursion theory** Here, students learn about Turing machines and their principle limitations. Undecidable problems like the Halting Problem are also presented. While the immediate practical value of this material might be questionable, it is of uttermost importance for future computer scientists to know about the principal limitations of their subject. From a mathematical standpoint, they will learn about diagonalization and dovetailing techniques. Moreover, here they will first encounter the notion of a reduction.

**complexity** Students will learn about $P$ versus $NP$ and about polynomial time reductions. Sometimes, other complexity classes can be touched, as well. This should tell them about feasible and unfeasible computational tasks, a borderline which is very important to grasp for understanding and designing algorithms.

Although it would be very nice and in my opinion also very useful if a short introduction into logic could be integrated into this program, as well, this would most likely not fit into a 3 hours lecture.
Organization of CompTheo. The order in which we listed the different topics covered by a standard course of COMP Theo will probably the one in which they are also taught. The reasoning behind this is that especially regular expressions and finite automata are the notions with which students might be even already familiar, so that this part of the lecture is a good warm-up for the rest of it. A disadvantage of this order might be that students could get the impression that the presented material is getting more and more difficult throughout the semester, so that some of the students might even give up any hope of grasping the further material after having been confronted with say the Halting Problem. This makes teaching basics of complexity theory at the end a special challenge.

Mostly, lectures will be accompanied with tutorials. This way of teaching can only be recommended since it will offer the students the possibility of training the notions they encountered in the lecture by solving exercises. If the tutorials are not given by the lecturer him/herself, the students will often have the advantage of hearing the material explained a second time, maybe in a different fashion. Conversely, if the lecturer is giving the tutorials him/herself, this will tell him/her more directly what and how well the students understood of the material presented.

Anyways, the material covered in the tutorials will be essentially and necessarily the one covered in the lectures two weeks ago. In essence, this means that there are no tutorial exercises on the very last weeks of the lecture, so that this material would be poorly covered if we do not put special attention to the teaching then. As explained in the previous paragraphs, this possibly poorly covered material will be complexity theory.

Finally, sometimes “mid-semester” exam(s) are offered to show the students what kind of questions they might expect in the final exam. Not surprisingly, this kind of exam usually only covers the first parts of the lecture, especially excluding complexity theory.

All this might explain why, when it comes to the final exams, questions touching complexity theory use to be omitted by the students or are answered only poorly.

What can we as teachers do about this? The basic strategy I propose is providing as many examples as possible especially in the last two weeks, hence trying to compensate for the lack of tutorials covering that material.

2 Teaching reductions through many examples


The main reason for choosing this book was in fact the “gentleness” it provides. This can be especially seen by the deliberate omission of much of the material found in older textbooks dating back to the times when students were provided with a solid mathematical background before studying COMP Theo. Most probably, in those times also more lecture time was dedicated to COMP Theo than it now uses to be. This leads to the observation that many other textbooks on this subject are simply not “coverable” in a decent way. Of course, this textbook is not completely unique in this respect. For example, in Germany we

One of the things which was omitted in Kinber and Smith's book is a proof of Cook's theorem. In my opinion, this was an understandable omission if you look at the lengthy standard proofs of this theorem. Conversely, this of course deprives the brighter students of one of the most beautiful results in the area.

The solution out of this dilemma which I am going to offer presents a proof of Cook's theorem as a series of "small" reductions, each posed in the form of a lemma. Each of these small proofs is easily presentable on one or two slides. This way, the students can not only follow the proof step by step, but they also get lots of examples of "easy reductions," the type of reduction which most likely is going to be asked for in a final exam.

A second source of reductions comes from the fact that, in order to obtain $NP$-hardness results, mostly variants of SAT are used. The way to obtain these variants offers new opportunities for showing "easy reductions."

Let us mention that Kinber and Smith only introduce the simplest form of reduction: The language $L \subseteq \Sigma^*$ is hence polynomial-time (many-one) reducible to the language $R$ if there exists a polynomial-time computable reduction function $r$ such that, for any $w \in \Sigma^*$, $w \in L$ iff $r(w) \in R$. The notion of $NP$-hardness is based on this notion of reducibility.

### 3 Presenting a proof of Cook's theorem

Recall that a proof of Cook's theorem usually relies on a normal form representation for (computations of) nondeterministic Turing machines (NTM for short)

In actual fact, on the way for obtaining these normal form, we can present many "easy" polynomial-time reductions.

#### 3.1 Normal forms for Turing machine computations

**The n-step Accepting Problem I**

$$ \text{ACC}_n = \{(T) \langle w \rangle \langle n \rangle | \text{NTM } T \text{ accepts } w \text{ in at most } n \text{ steps} \} $$

**Thm. 1** $\text{ACC}_n$ is $NP$-hard.

**Proof.** If $L \in NP$, then there is an NTM $T$ deciding $w \in L$ in time $p(|w|)$. So, $w \in L$ iff $(T) \langle w \rangle \langle p(|w|) \rangle \in \text{ACC}_n$.

As an aside, let us remark that in the book of Kinber and Smith, the class $NP$ is not introduced by polynomial-time computations of nondeterministic Turing machines, but via deterministic verification algorithms of a polynomial-sized certificate. The previous theorem would allow us to conclude that both ways of defining $NP$ are in fact equivalent:

"Guess and Check" for $\text{ACC}_n$

A certificate of $x = (T) \langle w \rangle \langle n \rangle \in \text{ACC}_n$ is a (codified) sequence

$$(q_1, w_1), (q_2, w_2), \ldots, (q_m, w_m)$$

of configurations with $m \leq n$. Such a sequence is of length polynomial in $|x|$.

In the "checking phase", the algorithm tests the following:
• Is \( w = w_1 \)?
• Is \( q_m \) accepting?
• For each \( 1 \leq k < m \), test if \( (q_k, w_k) \rightarrow_T (q_{k+1}, w_{k+1}) \) (by using a universal TM in a deterministic fashion)?

The certificate is verified if each of this questions is answered with YES.

**Cor. 2** \( \text{ACC}_\leq \) is \( \mathcal{NP} \)-complete.

**Cor. 3** A language is accepted by an NTM in polynomial time iff it is in \( \mathcal{NP} \) according to the definition given by Kimber and Smith.

**Proof.** The previous theorem shows that any language \( L \) accepted by a NTM in polynomial time can be translated into an instance of \( \text{ACC}_\leq \). The preceding reasoning hence shows how to decide \( L \) with the "Guess-and-Check" paradigm. Conversely, an NTM simulating a "Guess-and-Check" algorithm only has to firstly guess the certificate bit by bit (which is possible since it has only polynomial size) and then "run" the deterministic verification algorithm.

**ACC Variants I**
\[ \text{ACC}_\leq = \{ (T) \langle w \rangle \$^n | \text{NTM } T \text{ accepts } w \text{ in exactly } n \text{ steps } \} \]

**Thm. 4** \( \text{ACC}_\leq \) is \( \mathcal{NP} \)-complete.

**Proof.** Membership in \( \mathcal{NP} \) similar as before.

We reduce \( \text{ACC}_\leq \) to \( \text{ACC}_\leq \):

Given \( T \), modify it by adding rules \( ((q, a), (q, a)) \) for all states \( q \) and all tape symbols \( a \) in order to obtain TM \( T' \). This allows \( T' \) to "wait" if \( T \) happens to accept "too fast". Hence,

\[ (T) \langle w \rangle \$^n \in \text{ACC}_\leq \iff (T') \langle w \rangle \$^n \in \text{ACC}_\leq \]

**ACC Variants II**
\[ \text{ACC}'_\leq = \{ (T) \langle w \rangle | \text{NTM } T \text{ accepts } w \text{ in exactly } |w| \text{ steps } \} \]

**Thm. 5** \( \text{ACC}'_\leq \) is \( \mathcal{NP} \)-complete.

**Proof.** Membership in \( \mathcal{NP} \) similar as before (see also below).

We reduce \( \text{ACC}_\leq \) to \( \text{ACC}'_\leq \):

If \( |w| \geq n \), translate the \( \text{ACC}_\leq \)-instance \( (T) \langle w \rangle \$^n \) into an equivalent \( \text{ACC}'_\leq \)-instance \( (T) \langle w \rangle \$^n \) by adding wait loops.

Otherwise, proceed as follows: Given \( w \), define \( w' = w \$^{|w| - n} \). Hence,

\[ (T) \langle w \rangle \$^n \in \text{ACC}_\leq \iff (T) \langle w' \rangle \in \text{ACC}'_\leq \]

**Computation Carpet Certificates**
Convention: We now assume (w.l.o.g.) that the tape alphabet and the state set of a Turing machine are disjoint. A configuration \( (q, a_\omega w) \) can then be written as \( a_\omega w \).
Given an instance \((T, w)\) of \(\text{ACC}_m\), with \(\varphi(sw) = a_{1,1} \ldots a_{1,n+2}\), a certificate (for some Guess and Check algorithm) can be viewed as a 2-dimensional "carpet" (of size polynomial in \(|w|\) and \((T)\):

\[
CC = \begin{array}{cccc}
a_{11} & a_{12} & \cdots & a_{1,n+2} \\
a_{21} & a_{22} & \cdots & a_{2,n+2} \\
\vdots & \vdots & \ddots & \vdots \\
a_{n+1,1} & a_{n+1,2} & \cdots & a_{n+1,n+2}
\end{array}
\]

Verifying Computation Carpet Certificates

A computation carpet certificate \(CC\) will be verified if

1. \(\varphi(sw) = a_{1,1} \ldots a_{1,n+2}\),
2. \(\varphi(h 1 \cup_{n-1} = a_{n+1,1} \ldots a_{n+1,n+2}, \text{ and}
3. each \(j\)th row in the carpet is a successor configuration of the \((j-1)\)th row \((1 < j \leq n + 1)\).

"Guess and Check" for \(\text{ACC}_m\)

Guess some computation carpet certificate \(CC\) and then verify it deterministically. \(\leadsto\) \(\text{ACC}_m \in \mathcal{NP}\).

3.2 SAT and computation carpets

SAT is a language of Boolean expressions (B.E.), more precisely, \(\text{SAT} = \{ (E) \mid E\text{ is satisfiable} \}\) is a language over \(\{\land, \lor, \neg, (,), x, 0, 1\}\).

Lem. 6 If \(E\) is a B.E. of "length" \(n\), then \(|(E)| = O(n \log n)|.

For polynomial-time computations, it is hence insignificant whether referring to the length of \(E\) or of \((E)\).

By guessing a satisfying assignment, we get:

Lem. 7 \(\text{SAT} \in \mathcal{NP}\).

Computation carpets are employed in order to prove Cook's Theorem:

Thm. 8 \(\text{SAT}\) is \(\mathcal{NP}\)-complete.

Proof. We already know: \(\text{SAT} \in \mathcal{NP}\).

On the following slides, we show how to describe computation carpets by a Boolean expression (having a size polynomially related to the size of the carpet).

This way, we will reduce \(\text{ACC}_m\) to SAT.

To an instance \((T, w)\) of \(\text{ACC}_m\), we associate an expression

\[
E(T, w) = E_{S_1} \land E_{S_2} \land E_{S_3}(w) \land E_{S_4} \land E_T
\]

which is satisfiable iff \(T\) accepts \(w\) in exactly \(|w|\) steps.
Describing Carpet Patterns

We introduce the variables $C_{i, j, X}$ which are “true” iff there is an $X$ at position $(i, j)$ of a computation carpet. As a shorthand, we write

$$C_{i, j, a_1 \ldots a_t} : \iff C_{i, j, a_1} \land C_{i, j+1, a_2} \land \ldots \land C_{i, j+t-1, a_t}$$

There are some immediate “syntactical conditions” to be posed:

- At each position, a symbol must be specified: $E_{S_1} = \bigvee_{i, j} \forall X \ C_{i, j, X}$.
- At each position, not more than one symbol may be specified:

$$E_{S_2} = \bigwedge_{i, j} \bigwedge_{X, Y, X \neq Y} \overline{C_{i, j, X}} \lor \overline{C_{i, j, Y}}$$

Describing The Start And The End

- The initial configuration, given by the start state $s$ and the input word $w = a_1 \ldots a_n$, is correctly described: $E_{S_3}(w) = C_{1, 1, \ldots, a_n}$.
- The final configuration is correctly described: $E_{S_4} = C_{n+1, 1, \ldots, h_{1, 1}, a_{n-1}}$.

Describing Transitions

Depending on $T$, we can design a predicate $P_T$ to check whether or not symbols $u = u_0 u_1 u_2$ with $C_{i, j, u}$ (one of them holding a state) can possibly be there by looking at the symbols $v = v_0 v_1 v_2$ with $C_{i-1, j, v}$.

More precisely, the following items describe all possible cases when $P_T(u; v)$ yields true (1). Consider a configuration $C = xaqby$ of $T$.

- If $((q, b), (p, c))$ is a transition of $T$, then $xapcy$ is a successor of $C$ if $q$ is not a halting state. Hence, set $P_T(aqb; apc) = 1$ for any $a$.
- If $((q, b), (p, \rightarrow))$ is a transition of $T$, then $xabpy$ is a successor of $C$ if $q$ is not a halting state. Here, set $P_T(aqb; abp) = 1$ for any $a$.
- If $((q, b), (p, \leftarrow))$ is a transition of $T$, then $zqaby$ is a successor of $C$ if $q$ is not a halting state. Now, set $P_T(aqb; pab) = 1$ for any $a$.

This describes a predicate $P_T(u; v)$. All other symbols stay the same from row $i - 1$ to row $i$.

$$E_T = \bigwedge_{i > 1} \bigvee_{j} \bigvee_{P_T(UVW; XYZ)} C_{i, j, UVW} \land C_{i-1, j, XYZ}$$

$$\land \bigwedge_{k, k \notin \{j, j+1, j+2\}} \bigvee_{X} C_{i, k, X} \iff C_{i-1, k, X}$$

Here, $\bigvee_{P_T(UVW; XYZ)}$ is a shorthand for $\bigvee_{U, V, W, X, Y, Z, P_T(UVW; XYZ)}$.

4 Further $NP$-hardness results

To classify other combinatorial problems as $NP$-hard, mostly variants of SAT are used in the proofs which allow only restricted forms of B.E.
4.1 Variations of SAT

LIT-SAT
Let us call a B.E. to be in literal normal form (literal NF) if any negation may only directly refer to a variable occurrence.
Ex.: \( \neg(\neg(x_1 \lor x_2) \land x_2) \) is not in literal NF.

\[ \text{LIT-SAT} = \{(E) \mid E \text{ is a B.E. in literal NF which is satisfiable} \} \]

Thm. 9 LIT-SAT is polynomial-time reducible to SAT.

Proof. Apply de Morgan's laws "exhaustively". \( \square \)
Ex.: \( \neg(\neg(x_1 \lor x_2) \land x_2) \)
= \( \neg\neg(x_1 \lor x_2) \lor \neg x_2 \)
= \( x_1 \lor x_2 \lor \neg x_2 \).

An interesting question to (or from) students might be: What about polynomial-time?

A B.E. is said to be in conjunctive normal form (CNF) if it is of the form

\[ C_1 \land C_2 \land \ldots \land C_k \]

and each \( C_j \) (called a clause) is of the form

\[ C_j = \alpha_{j1} \lor \alpha_{j2} \lor \ldots \lor \alpha_{j_{r_j}} \]

where each \( \alpha_{ij} \) is a literal, i.e., either \( x \) or \( \neg x \) for some variable \( x \).
We also write \( \bar{x} \) instead of \( \neg x \).
Ex.: \( (x_1 \lor x_2 \lor x_3) \land (\bar{x}_1 \lor \bar{x}_2 \lor x_3) \) is in CNF.

\[ \text{CSAT} = \{(E) \mid E \text{ is a B.E. in CNF which is satisfiable} \} \]

Thm. 10 CSAT is polynomial-time reducible to LIT-SAT.

CSAT \( \rightarrow \) LIT-SAT
Let \( V_1 \) and \( V_2 \) be sets of variables with \( V_1 \subseteq V_2 \). An assignment \( \alpha_2 \) of \( V_2 \) extends an assignment \( \alpha_1 \) of \( V_1 \) if \( \alpha_1(x) = \alpha_2(x) \) for all \( x \in V_1 \).

By induction on the number \( r \) of occurrences of \( \land \) and \( \lor \), we shall prove:

CLAIM: Any B.E. \( E \) in literal NF can be transformed into a B.E. \( E' \) in CNF consisting of at most \( n \) clauses over a set of variables \( V' \) that include the variables \( V \) of \( E \) and at most \( n \) other variables s.t. \( E \) is satisfiable iff \( E' \) is satisfiable.
Moreover, if an assignment \( \alpha \) of \( V \) satisfies \( E \), then there is an extension \( \alpha' \) of \( \alpha \) that evaluates to 1 on \( E' \).
Conversely, if \( \alpha' \) is an assignment to \( V' \) that evaluates to 1, then its restriction to \( V \) satisfies \( E \).

Proof of CLAIM:

Anchor: \( r = 0 \)

Induction step: We distinguish two cases:
1. \( E = E_1 \land E_2 \). By I.H., \( E_1 \) can be transformed into \( E'_1 \) and \( E_2 \) into \( E'_2 \).
   W.l.o.g., assume that no variable that is not present in \( E \) appears both in \( E'_1 \) and in \( E'_2 \). Then, \( E' := E'_1 \land E'_2 \) satisfies the requirements.
2. \( E = E_1 \lor E_2 \). By I.H., \( E_1 \) can be transformed into \( E'_1 \) and \( E_2 \) into \( E'_2 \).

Let

\[ E'_j = C_{j,1} \land \ldots \land C_{j,r_j} \]

Let \( y \) be a new variable and set \( E' := E'_1 \lor \bar{E}'_2 \), where

\[ \bar{E}_j = \bar{C}_{j,1} \land \ldots \land \bar{C}_{j,r_j} \]

with \( \bar{C}_{1,i} := y \lor C_{1,i} \) and \( \bar{C}_{2,i} := \bar{y} \lor C_{2,i} \).

**Transitivity:** \( \text{SAT} \rightarrow \text{CSAT} \)

Ex.: Consider

\[ \neg(\neg(x_1 \lor x_2) \land (\neg x_1 \lor x_2)) \]

De Morgan's law transforms into LIT-SAT:

\[
E = (x_1 \lor x_2) \lor (x_1 \land \bar{x}_3) =: E_1 =: E_2
\]

\( E_1 \) is then transformed into

\[
E'_1 = (x_1 \lor y_1) \land (x_2 \lor \bar{y}_1),
\]

while \( E'_2 = E_2 \).

Hence, \( E' = (x_1 \lor y_1 \lor y_2) \land (x_2 \lor \bar{y}_1 \lor y_2) \land (x_1 \lor \bar{y}_2) \land (\bar{x}_3 \lor \bar{y}_2) \)

\( x_1 = 0, x_2 = 1 \) and \( x_3 = 1 \) is an assignment \( \alpha \) satisfying \( E \).

By setting \( y_1 = 1 \) and \( y_2 = 0 \), we can extend \( \alpha \) as to satisfy \( E' \).

**3-SAT**

A B.E. in CNF is in 3NF if each clause contains at most three variables or negated variables.

**Thm. 11** 3-SAT is polynomial-time reducible to CSAT.

**Proof.** Consider the following procedure:

**WHILE** \( E \) contains a "large" clause with more than 3 variables **DO**

Pick some "large" clause \( C = \ell_1 \lor \ldots \lor \ell_k \). Replace \( C \) in \( E \) by

1. \( C_{\text{short}} = \ell_1 \lor \ell_2 \lor y \) and
2. \( C_{\text{tail}} = y \lor \ell_3 \lor \ldots \lor \ell_k \)

**OD** (where \( y \) is a "new" variable)

Observe: \( C_{\text{tail}} \) is shorter than \( C \! \! \! \)  

**Ex-3-SAT**

A B.E. in CNF is in exact 3NF if each clause contains exactly three variables or negated variables.

**Thm. 12** Ex-3-SAT is polynomial-time reducible to 3-SAT.

**Proof.** Consider the following procedure:

**WHILE** \( E \) contains a "small" clause with less than 3 variables **DO**

Pick some "small" clause \( C \). Replace \( C \) in \( E \) by

1. \( C_1 = C \lor y \) and
2. \( C_2 = C \lor \bar{y} \)

**OD** (where \( y \) is a "new" variable)

Observe: \( C_1, C_2 \) are longer than \( C \! \! \! \)
4.2 An application: CLIQUE

The standard proof for showing \( \mathcal{NP} \)-hardness of CLIQUE actually uses this latter variant of SAT.

CLIQUE revisited

\[ C = \{(G) \in \mathcal{NP} \mid G \text{ has a clique of size } k\} \]

Thm. 13 \( C \) is polynomial-time reducible to Ex-3-SAT.

Proof. Let \( E = C_1 \land \ldots \land C_n \) be a B.E. in exact 3NF. Construct a graph \( G \) s.t. \( E \) is satisfiable iff \( G \) has a clique of size \( n \).

For each clause \( C_r = (\ell_1^r \lor \ell_2^r \lor \ell_3^r) \), create three vertices \( v_1^r, v_2^r \) and \( v_3^r \) in \( G \). Connect \( v_1^r \) and \( v_2^r \) by an edge iff

1. \( r \neq s \) and
2. \( \ell_1^r \) is not the negation of \( \ell_1^s \).

The reduction can be computed in polynomial time.

The correctness of the construction is seen as follows:

Let \( a \) be a satisfying assignment of \( E \). Each clause must contain some literal \( \ell_1^r \) evaluating to 1. Pick the corresponding vertex \( v_1^r \). The vertices picked this way form a clique of size \( n \).

Let \( G' \) be a clique in \( G \) of size \( n \). By the first condition, no two vertices of \( G' \) "belong" to the same clause. Making the literals "true" which correspond to clique-vertices (this yields no contradiction by the second condition) gives a (partial) assignment which can be extended arbitrarily to give a satisfying assignment of \( E \). \( \square \)

5 Conclusions

We aimed at presenting the basics of complexity theory in a very fine-sliced fashion, compared to what can usually be found in textbooks. For example, in M. Sipser: Introduction to the Theory of Computation. PWS Publishing, 1996, there are only two proofs showing Cook's theorem and showing the \( \mathcal{NP} \)-completeness of Ex-3-SAT, respectively. We think that this approach offers some advantages, especially, since it provides lots of "easy" examples for the crucial notion of a reduction. Moreover, proofs which are going over 6 pages in a textbook simply shy away most students. Finally, the transitivity of the reduction relation can be neatly explained this way.

Although a validation of the proposal is hard to make in my case (the previous lecture of COMPTHEO in Newcastle was hampered by a long-term illness of the lecturer, and the (mathematical) preconditions of students taking this sort of course in Australia and in Germany are too different to make a fair comparison), I am very confident that this fine-sliced approach helps overcome some of the difficulties in teaching complexity theory to undergraduate students, as discussed in the introduction.
CS&P'2002, the Workshop on Communication, Specification & Programming took place at the Institute of Informatics of the Humboldt University at the old site in the centre of Berlin, from October 7-9, 2002. It was the 11th in this series, started in 1992 at Berlin by Peter Starke and Ludwik Czaja.

It was attended by about 50 participants from 7 countries (PL 21, D 15, I 7, RUS 3, IL 2, F 1, SF 1).

The program covered 33 talks, and ranged from theory to practice of that field.

The contributions came from 9 countries:

<table>
<thead>
<tr>
<th>Country</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>8</td>
</tr>
<tr>
<td>IL</td>
<td>1</td>
</tr>
<tr>
<td>CDN</td>
<td>3</td>
</tr>
<tr>
<td>PL</td>
<td>14</td>
</tr>
<tr>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>GB</td>
<td>5</td>
</tr>
<tr>
<td>RUS</td>
<td>2</td>
</tr>
<tr>
<td>SF</td>
<td>1</td>
</tr>
<tr>
<td>I</td>
<td>4</td>
</tr>
</tbody>
</table>

The detailed program is given below.

The workshop was opened on Monday morning by Hans-Dieter Burkhard. It was well organized by Hans-Dieter Burkhard, Ludwik Czaja, Andrzej Skowron, and Peter Starke, as well as Gabriela Lindemann, Christine Jung, Adelheid Lau, Renate Zirkelbach, and took place in a relaxed atmosphere.

The proceedings, containing all papers, have been published as Informatik-Bericht 161 (2 volumes) of Humboldt-Universität zu Berlin, with editors the organizers of the workshop. It is planned to publish a selection of the papers in Fundamenta Informaticae.

In the breaks coffee, tea, mineral water, and biscuits were served. Lunch was available in the university restaurant nearby. There were also three PC's for Internet connection.

Most foreign participants stayed in the guest house of the Humboldt University, in 10 minutes walking distance from the conference site.

Program of Concurrency, Specification and Programming (CS&P 2002)

Speakers are given in bold face.

Monday, October 7, 2002

9.00 - 9.15 Opening

   Approximations in Information Nets

9.45 - 10.15 W. Penczek, A. Lomuscio
   Bounded Model Checking for Interpreted Systems
10.15 - 10.45 A. Salwicki, M. Warpechowski
    Combining Inheritance and Nesting of Classes together: Advantages and Problems
10.45 - 11.15 Coffee break
11.15 - 11.45 M. Köhler, H. Rölke
    Mobile Object Net Systems: Concurrency and Mobility
11.45 - 12.15 B. Farwer, K. Misra
    Hierarchical Object Systems
12.15 - 14.00 Lunch break
14.00 - 14.30 M. Minor
    The Communication of a System's Treasury of Experiences
14.30 - 15.00 D. Monett Díaz
    On a Method for Deciding on Future Actions in Agent Negotiations
15.00 - 15.30 J. Bach, M. Jüngel
    Using Pattern Matching on a Flexible, Horizon-aligned Grid for Robotic Vision
15.30 - 16.00 Coffee break
16.00 - 16.30 A. Póróla, W. Penczek, M. Szreter
    Refining partitioning for checking reachability in Timed Automata
16.30 - 17.00 B. Woźna, W. Penczek, A. Zbrzezny
    Reachability for Timed Systems based on SAT-Solvers
17.00 - 17.30 M. Szreter
    Partial Order Reductions for TCTL in Automata Model Checking

Tuesday, October 8, 2002

9.00 - 9.30 L. Czaja
    Proving Petri Nets Correct via Cause-Effect Structures
9.30 - 10.00 H.-S., Nguyên, S.-H. Nguyên
    Lazy Classification Method Based on Boolean Reasoning Approach
10.00 - 10.30 A. Dańko, G. Mirkowska
    Multivalued Boolean-Fuzzy Logics
10.30 - 11.00 Coffee break
11.00 - 11.30 K. Pancerz, Z. Suraj
    The Synthesis of Concurrent Systems Specified by Information Systems with Using the Colored Petri Nets
11.30 - 12.00 N. De Francesco, A. Santone (presented by Luca Tesei)
    Checking Secure Information Flow in Concurrent Languages by Abstract Interpretation + Model Checking
12.00 - 14.00 Lunch break
14.00 - 14.30 P. Chrzastowski-Wachtel
    Recovery Nets: Model for Dynamic Workflows
14.30 - 15.00 K. Schmidt
    Distributed Verification with LoLa
15.00 - 15.30 S. Haar
    Probabilizing Parallelism in Cluster Unfoldings
15.30 - 16.00 Coffee break
16.00 - 16.30  L. Popova-Zeugmann, M. Werner, J. Richling  
A State Equation for Timed Petri Nets  
16.30 - 17.00  A. Doros, A. Janowska, P. Janowski  
From Specification Languages to Timed Automata  
17.00 - 17.30  R. Barbuti, L. Tesei  
A Decidable Notion of Timed Non-interference  

Wednesday, October 9, 2002  
9.00 - 9.30  K. Varpaaniemi  
Towards Ambitious Approximation Algorithms inStubborn Set  
Optimization  
9.30 - 10.00  I. G. Tabakov  
An Introduction to the Place-Transition Nets k-Distinguishability  
10.00 - 10.30  N. S. Moskalyova, I. Virbiskaite  
Observing Timed Partial Order Equivalences Categorically  
10.30 - 11.00  Coffee break  
11.00 - 11.30  B. Farwer, M. Kudlek, K. Misra  
Some Considerations on Higher Order Petri Nets  
11.30 - 12.00  R. Lanotte, A. Schettini, A. Peron, S. Tini  
Dynamic Hierarchical Machines  
12.00 - 14.00  Lunch break  
14.00 - 14.30  A. Coja-Oghlan, M.-O. Stehr  
Revisiting the Algebra of Petri Net Processes under the Collective Token  
Philosophy  
14.30 - 15.00  M. Kacprzak  
On Undecidability of Multiagent Logic  
15.00 - 15.30  U. Abraham, T. Matuska (Pinhas)  
Exercises in Style (Alpha Specifications)  
15.30 - 16.00  Coffee break  
16.00 - 16.30  R. Latkowski  
On Decomposition for Incomplete Data  
16.30 - 17.00  V. Bashkin, I. Lomazowa  
Resource bisimulations in Nested Petri Nets  
17.00 - 17.30  M. Bellia, E. Occhiuto  
New Bounds in Parallel Unifications  
17.30  Concluding remarks about CS&P'2002
The theme of the meeting was Combinatorics and Combinatorial Aspects of Biology. Talks were held at the Plymouth International Hotel in New Plymouth and attendees were housed at a number of motels in the New Plymouth area. Talks were held from Sunday to the following Saturday with Wednesday being kept free. As usual the daily program was lectures in the morning and in the evening with afternoons being kept free to work on mathematics or to relax and enjoy the stunning weather. David Gauld led an intrepid group of mountaineers on a summit attempt of Mt Taranaki on Wednesday.

We had an outstanding group of speakers. These were Karl Broman (Johns Hopkins University, Recombination Mapping); Andreas Dress (University of Beilefeld, Overview of Combinatorial Biology); Martin Grohe (University of Edinburgh, Logical Aspects of Graphs); Mike Hallett (McGill University, Parametric Aspects of Computational Biology); Lior Pachter (University of California, Berkeley, Genefinding); Neil Robertson (Ohio State University, The Graph Minors Project); Paul Seymour (Princeton University, The Perfect Graph Theorem); Terry Speed (University of California, Berkeley, Mathematical Aspects of Gene Expression); Richard Stanley (MIT, Enumerative Combinatorics); and Tandy Warnow (University of Texas, Mathematical Aspects of Phylogeny).

Overall the standard of talks was excellent. A highlight of the meeting was the breadth of the talks, ranging from very applied to quite pure. This diversity certainly made for interesting and stimulating sessions. Diversity was also reflected in the attendees who included pure mathematicians, statisticians, computer scientists and biologists.

About 90 people were accommodated for the meeting; this included families of attendees. Typically talks had an attendance of about 55. There were about 20 graduate students.

Rod Downey, Geoff Whittle
DIMACS Workshops and Events Calendar

This calendar lists scheduled workshops upcoming at DIMACS. DIMACS' WWW pages have additional information as well as programs scheduled on short notice. Programs will be held at DIMACS facilities at Rutgers University (CoRE Building) unless otherwise noted.

DIMACS 2000-2003 Special Focus on Computational Molecular Biology

June 2003:
- BioMaPS/DIMACS tutorial: Introduction to Modern Concepts in Biology for Mathematical and Physical Scientists
  Dates: June 23 - July 3, 2003
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizers: William Sofer and Paul Ehrlich, Rutgers University
  Email: sofer@waksman.rutgers.edu, pehrlich@lutece.rutgers.edu

October 2003:
- Data Mining Techniques in Bioinformatics
  Dates: October 30 - 31, 2003
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizer: Mona Singh, Princeton University; Mark Gerstein, Yale University
  Email: mona@cs.princeton.edu, Mark.Gerstein@yale.edu

Workshops Under Development:
- Interface Between Biology and Game Theory
  Organizers: Adam Arkin, Lawrence Berkeley Labs and UC Berkeley; Denise Wolf, Lawrence Berkeley Labs; Vijay Vazirani, Georgia Tech
  Email: dwolf@lbl.gov, vazirani@cc.gatech.edu

DIMACS 2001-2004 Special Focus on Computational Information Theory and Coding

June 2003:
- Complexity and Inference
  Dates: June 2 - 5, 2003
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizers: Mark Hansen, Bell Labs; Paul Vitanyi, University of Amsterdam; Bin Yu, University of California, Berkeley
  Email: cocteau@research.bell-labs.com, Paul.Vitanyi@cwi.nl, binyu@stat.berkeley.edu

December 2003:
- Coding Theory and Discrete Mathematics
  Dates: December 15 - 18, 2003
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizers: Alexander Barg, DIMACS/Rutgers University; Alexei Ashikhmin, Bell Labs; Iwan Duursma, University of Illinois
  Email: abarg@dimacs.rutgers.edu, aea@research.bell-labs.com, duursma@math.uiuc.edu

Workshops Under Development:
- Mini-workshop: Information Theory and Symbolic Dynamics
  Dates: Spring 2003
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizers: Emina Soljanin, Bell Labs; Paul Siegel, Univ. of California - San Diego, Bane Vasic, University of Arizona
  Email: emina@lucent.com, psiegel@ucsd.edu, vasic@ece.arizona.edu
• Working Group Meeting: Optical/Magnetic Recording and Optical Transmission
  Dates: first meeting Spring 2003, second meeting Spring 2004
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizers: Emina Soljanin, Bell Labs; Paul Siegel, Univ. of California - San Diego, Bane Vasic, University of Arizona
  Email: esmina@lucent.com, psiegel@ucsd.edu, vasic@ece.arizona.edu

DIMACS 2000-2003 Special Focus on Mathematics and the Foundations of Computer and Information Science

July 2003:

• Applications of Lattices and Ordered Sets to Computer Science
  Dates: July 8 - 10, 2003
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizers: Jonathan Farley, Massachusetts Institute of Technology; Melvin Janowitz, DIMACS / Rutgers University, Jimmie Lawson, Louisiana State University; Michael Mislove, Tulane University
  Email: melj@dimacs.rutgers.edu, lawson@math.lsu.edu, mwm@math.tulane.edu

DIMACS 2001-2004 Special Focus on Data Analysis and Mining

June 2003:

• Management and Processing of Data Streams
  (In conjunction with ACM SIGMOD/PODS and FCRC 2003)
  Date: Sunday, June 8, 2003
  Location: San Diego, California, USA
  Organizers: S. Muthukrishnan, Rutgers University; Divesh Srivastava, AT&T Labs - Research
  Email: muthu@cs.rutgers.edu, divesh@research.att.com

• Algorithms for Multidimensional Scaling II
  Dates: June 11-12, 2003
  Location: Doubletree Hotel in Tallahassee, Florida
  Organizers: J. Douglas Carroll (chairman), Rutgers University; Phipps Arabie, Rutgers University; Larry Hubert, University of Illinois; Michael Trosset, The College of William & Mary; Mike Brusco, Florida State University; Mel Janowitz, DIMACS
  Email: dcarroll@rci.rutgers.edu, arabie@andromeda.rutgers.edu, llhubert@psych.uiuc.edu, trosset@math.wm.edu, mbrusco@garnet.acns.fsu.edu, melj@dimacs.rutgers.edu

September 2003:

• Working Group Meeting: Computer-Generated Conjectures from Graph Theoretic and Chemical Databases II
  Dates: September 8 - 12, 2003
  Location: Centre de Recherches Mathematiques (CRM), Universite de Montreal, Quebec, Canada
  Organizers: Patrick Fowler, University of Exeter; Pierre Hansen, GERAD - University of Montreal
  Email: P.W.Fowler@exeter.ac.uk, pierreh@crt.umontreal.ca
  This meeting is by invitation only.

October 2003:

• Data Mining Techniques in Bioinformatics
  Dates: October 30 - 31, 2003
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizers: Mona Singh, Princeton University; Mark Gerstein, Yale University
  Email: mona@cs.princeton.edu, Mark.Gerstein@yale.edu
November 2003:

- **Data Cleaning/Noisy Data**
  Dates: November 3-7, 2003
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizers: Parvi Dasu, AT&T
  Email: tamr@research.att.com

**Events Under Development:**

- **Workshop: Geographical Information Systems: Representation Algorithms for Terrain**
  Dates: TBA (tentatively 2003)
  Location: Research Triangle, North Carolina
  Organizers: Lars Arge, Duke University; Jack Snoeyink, University of North Carolina, Chapel Hill
  Email: large@cs.duke.edu, snoeyink@cs.unc.edu

- **Workshop: Algorithms for Clustering in Data Mining**
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizers: Pierre Hansen, GERAD - University of Montreal; Moses Charikar, Princeton University
  Email: pierreh@crt.umontreal.ca, moses@cs.princeton.edu

**DIMACS 2002-2005 Special Focus on Computational Geometry and Applications**

October 2003:

- **Computer-Aided Design and Manufacturing**
  Dates: October 7 - 9, 2003
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizers: Debasish (Deba) Dutta, University of Michigan; Ravi Janardan, University of Minnesota; Michiel Smid, Carleton University
  Email: dutta@engin.umich.edu, janardan@cs.umn.edu, michiel@scs.carleton.ca

**Workshops Under Development:**

- **Geographical Information Systems: Representation Algorithms for Terrain**
  Dates: TBA (tentatively 2003)
  Location: Research Triangle, North Carolina
  Organizers: Lars Arge, Duke University; Jack Snoeyink, University of North Carolina, Chapel Hill
  Email: large@cs.duke.edu, snoeyink@cs.unc.edu

  Organizers: William Steiger, Rutgers University
  Email: steiger@cs.rutgers.edu

- **Algorithm Implementation Challenge: Surface Reconstruction**
  Organizers: Tamal Dey, Ohio State University; Claudio Silva, AT&T Research
  Email: tama@dey.cs.ohio-state.edu, csilva@research.att.com
June 2003:

- **Tutorial: Statistical and Other Analytic Health Surveillance Methods**
  Dates: June 17 - 20, 2003
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizer: David Madigan, Rutgers University; Henry Rolka, CDC; Martin Kulldorff, University of Connecticut
  Email: madigan@stat.rutgers.edu, hrr2@cdc.gov, martink@neuron.uchc.edu

- **Working Group Meeting: The Ecology and Evolution of Influenza and Related Viruses**
  Dates: June 29 - July 2, 2003
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizer: Simon Levin, Princeton University
  Email: slevin@eno.princeton.edu
  This meeting is by invitation only

September 2003:

- **Working Group Meeting: Phylogenetic Trees and Rapidly Evolving Diseases**
  Dates: September 3 - 6, 2003
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizers: Edward Holmes, Oxford University; Mike Steel, University of Canterbury
  Email: Edward.Holmes@zoology.oxford.ac.uk, M.Steel@math.canterbury.ac.nz

October 2003:

- **Genetics and Evolution of Pathogens**
  Dates: October 20 - 24, 2003
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizers: Megan Murray, Harvard University; David Krakauer, Institute for Advanced Study; Marc Lipsitch, Harvard University
  Email: mmurray@hsph.harvard.edu, krakauer@ias.edu, mlipsitehsph.harvard.edu

May 2004:

- **Methodologies for Comparing Vaccination Strategies (Working Group Meeting)**
  Dates: May 10 - 14, 2004
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizers: John Glasser, Centers for Disease Control and Prevention; Herbert Hethcote, University of Iowa
  Email: jwg3@cdc.gov, herbert-hethcote@uiowa.edu

Events Under Development:

- **Workshops/Working Group Meetings on Modeling of Infectious Diseases**
- **Distributed Computing, Social Networks and Disease Spread Processes (Working Group Meeting) (first meeting, 2003)**
  Organizers: Vasak Chvatal and Fred Roberts, Rutgers University; Aleksandar Pekec, Duke University
  Email: chvatal@cs.rutgers.edu, froberts@dimacs.rutgers.edu, pekec@duke.edu

- **Models/Methodological Problems of Botanical Epidemiology (2005-2006)**
  Organizers: Chris Gilligan, Cambridge
  Email: cag1@cus.cam.ac.uk

- **Statistical, Mathematical, and Modeling Issues in the Analysis of Marine Diseases (Working Group Meeting) (first meeting, Fall 2003)**
  Organizers: Andy Dobson, Princeton University; Steve Ellner, Cornell University; C. Drew Harvell, Cornell University
  Email: andy@eno.princeton.edu, spe2@cornell, cdh5@cornell.edu
Workshops/Working Group Meetings on Modeling of NonInfectious Diseases

• Computational Biology of Tumor Progression (Joint with Institute for Advanced Study) (Working Group Meeting) (first meeting, 2003)
  Organizer: Martin Nowak, Institute for Advanced Study
  Email: nowak@ias.edu

• Disease Clusters (2004 - 2005)
  Organizer: Daniel Wartenberg, Robert Wood Johnson Medical School, UMDNJ
  Email: dew@eohsi.rutgers.edu

Workshops/Working Group Meetings on Evolution and Epidemiology

• The Evolution and Control of Drug Resistance
  Organizers: Sally Blower, UCLA Medical School; Tom Leitman, UCSF Medical School
  Email: SBlower@biomath.medsch.ucla.edu, tml@itsa.ucsf.edu

• The Epidemiology and Evolution of Influenza (2004-2005)
  Organizers: Catherine Macken and Alan Perelson, Los Alamos
  Email: cam@synapse.lanl.gov, asp@lanl.gov

• Models of Co-Evolution of Hosts and Pathogens (2006-2007)
  Organizers: Viggo Andreasen, Roskilde; Andrea Pugliese, Trento
  Email: viggo@fatou.ruk.dk, pugliese@science.unitn.it

Workshops/Working Group Meetings on Methodological Issues in Computational and Mathematical Epidemiology

• Spatial Epidemiology and Geographical Information Systems (2005-2006)
  Organizer: Daniel Wartenberg, Robert Wood Johnson Medical School, UMDNJ

• Predictive Methodologies for Infectious Diseases (Working Group Meeting) (first meeting, 2005-2006)
  Organizers: Sally Blower, UCLA Medical School; Christl Donnelly, Imperial College
  Email: SBlower@biomath.medsch.ucla.edu, c.donnelly@ic.ac.uk

• Capture-recapture Models in Epidemiology (2005 - 2006)
  Organizer: David Madigan, Rutgers University
  Email: madigan@stat.rutgers.edu

• Ecologic Inference (2006-2007)
  Organizers: Tom Webster, Boston University; Slyvia Richardson, Imperial School of Medicine
  Email: twebster@bu.edu, sylvia.richardson@ic.ac.uk

• Combinatorial Group Testing (2006-2007)
  Organizers: Ding-zhu Du, University of Minnesota; Frank Hwang, Chiaotung University
  Email: dzd@cs.umn.edu, fhwang@math.nctu.edu.tw

• Order-theoretic Aspects of Epidemiology (Working Group Meeting) (first meeting, 2004-2005)
  Organizers: David Ozonoff, Mel Jancowitz, Fred Roberts, Rutgers University
  Email: dozonoff@bu.edu, melj@dimacs.rutgers.edu, froberts@dimacs.rutgers.edu

Tutorial Program

• Discrete Math and Theoretical Computer Science for Epidemiologists and Biologists (Summer 2004)
  Organizers: Alun Lloyd, Institute for Advanced Study; Fred Roberts, Rutgers University; Katherine St. John, CUNY
  Email: alun@alunlloyd.com, froberts@dimacs.rutgers.edu, stjohn@lehman.cuny.edu
DIMACS 2003-2006 Special Focus on Communication Security and Information Privacy

August 2003:

- Tutorial: Computer Security
  Date: August 4 - 7, 2003
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizer: Rebecca Wright, Stevens Institute of Technology
  Email: rwright@cs.stevens-tech.edu

September 2003:

- Large-scale Internet Attacks
  Date: September 24 - 25, 2003
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizer: Vern Paxson, ICSI Center for Internet Research
  Email: vem@icir.org

December 2003:

- Working Group Meeting: Privacy / Confidentiality of Health Care Data
  Date: December 10 - 12, 2003
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizer: Rakesh Agrawal, IBM Almaden; Larry Cox, CDC; Joe Fred Gonzalez, CDC; Harry Guess, Merck
  Email: ragrawal@acm.org, lcpx@cdc.gov, jfg2@cdc.gov, harry.guess@merck.com

March 2004:

- Working Group: Secure, Efficient Extraction of Joint Information from Multiple Datasets
  Date: March 15-17, 2004
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizer: Rebecca Wright, Stevens Institute of Technology
  Email: rwright@cs.stevens-tech.edu

May 2004:

- Electronic Voting - Theory and Practice
  Date: May 27 - 28, 2004
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizer: Markus Jakobsson and Ari Juels, RSA Laboratories
  Email: mjacobsson@rsasecurity.com, ajuels@rsasecurity.com

Events Under Development:

- Intellectual Property Protection
  Date: TBD
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizer: Drew Dean, SRI International
  Email: ddean@cs.sri.com

- Security of Web Services and E-Commerce
  Date: TBD
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizer: Brian LaMacchia, Microsoft
  Email: bal@msoufte.com

- Cryptography: Theory Meets Practice
  Date: TBD
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizer: Dan Boneh, Stanford
  Email: dabo@cs.stanford.edu
• Security Analysis of Protocols
  Date: 2003
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizer: John Mitchell, Stanford and Ran Canetti, IBM Hawthorne
  Email: mitchell@cs.stanford.edu, canetti@watson.ibm.com

• Mobile and Wireless Security
  Date: TBD
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizer: Bill Arbaugh, University of Maryland
  Email: waa@cs.umd.edu

• Working Group Meeting: On-Line Privacy: Threats and Tools
  Date: 2003
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizer: Lorrie Cranor, AT&T
  Email: lorrie@research.att.com

• Working Group Meeting: Intrusion Detection and Network Security Management Systems
  Date: TBD
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizer: Joe McHugh, CERT, and S. Raj Rajagopalan, Telcordia Technologies
  Email: TBA, sraj@research.telcordia.com

• Security and Trust Issues Associated with Ad-Hoc Computing/Pervasive Networking
  Date: TBD
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizer: S. Raj Rajagopalan, Telcordia Technologies
  Email: sraj@research.telcordia.com

• Database Security: Query Authorization and Information Inference
  Date: TBD
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizer: H.V. Jagadish, University of Michigan
  Email: jag@eecs.umich.edu

• Working Group Meeting: Mobile Code Security
  Date: TBD
  Location: DIMACS Center, CoRE Building, Rutgers University
  Organizer: Dan Wallach, Rice University
  Email: dwallach@cs.rice.edu
CALL FOR PARTICIPATION

The series of MFCS symposia, organized alternately in the Czech Republic, Poland and Slovakia since 1972, has a well-established tradition. The MFCS symposia encourage high-quality research in all branches of theoretical computer science. Their broad scope provides an opportunity to bring together specialists who do not usually meet at specialized conferences.

MFCS 2003 is organized by the Slovak Society for Computer Science and the Comenius University in Bratislava, in cooperation with other institutions in Slovakia. It is supported by the European Association for Theoretical Computer Science and the Slovak Research Consortium for Informatics and Mathematics.

The scientific program of MFCS 2003 (http://www.mfcs.sk) consists of 7 invited talks and 55 contributed talks. The latter have been selected by the Program Committee consisting of Julian Bradfield (Edinburgh), Janos Csirik (Szeged), Pierpaolo Degano (Pisa), Mariangiola Dezani-Ciancaglini (Turin), Krzysztof Diks (Warsaw), Juhani Karhumaki (Turku), Marek Karpinski (Bonn), Mojmir Kretinsky (Brno), Werner Kuich (Vienna), Jan van Leeuwen (Utrecht), Christoph Meinel (Trier), Marcin Pacholski (Wroclaw), David Peleg (Rehovot), Jose Rolim (Geneve), Branislav Rovan - chair (Bratislava), Jan Rutten (Amsterdam), Paul Spirakis (Patras), Jiri Sima (Prague), Ulrich Untes-Nitsche (Fribourg), Peter Vojtas - vice-chair (Kosice), and Igor Walukiewicz (Bordeaux) from a total of 137 submitted papers.

Invited talks will be delivered by Harry Buhmann (Amsterdam), Roberto Gorrieri (Bologna), Giancarlo Mauri (Milano), Burkhard Monien (Paderborn), Don Sannella (Edinburgh), Wolfgang Thomas (Aachen), and Ingo Wegener (Dortmund).

Bratislava (population 540 000) is the capital of Slovak Republic, located close to the Austrian, Hungarian and Czech borders on both banks of the river Danube. Bratislava is a very old town with more than 2000 years history. Full town privileges were granted to Bratislava in 1291. The historic centre is dominated by the Bratislava Castle. Bratislava is the seat of Comenius University, Slovak University of Technology, and University of Economy. More information about the history and sightseeing in Bratislava can be found on the MFCS 2003 homepage http://www.mfcs.sk

The conference will be held in the hotel Sorea (http://www.sorea.sk) which is located about 3km west of the town centre. By regular tram connections it is possible to reach the town centre in about 10 minutes. As the capacity of the hotel is limited, additional rooms in the Hotel Druzba, the guest house of the Comenius University are also available. The booking will be performed on the first-come-first-served basis.

For registration and payment details see the conference web page http://www.mfcs.sk.

Conference address:
Branislav Rovan
MFCS 2003
Department of Computer Science
Comenius University
Mlynska dolina, FMFI UK
842 48 BRATISLAVA, Slovak Republic
Phone: (+421 2) 654 26 635
Fax: (+421 2) 654 27 041
E-mail: mfcs2003@dcs.fmph.uniba.sk
WWW: http://www.mfcs.sk
(Please, visit often for current information.)
Call for Papers

Compositional Verification of UML Models

Workshop of the UML 2003 Conference

The definition of UML has been motivated by the need for a standard notation for modelling system architectures and behaviours at functional and implementation level. The main focus has been essentially on terminology, notation and syntax without addressing semantic, validation and methodology issues which are important for formal design and verification techniques. This workshop addresses the application of formal methods and techniques that exploit the architectural structure of UML models in a compositional manner.

TOPICS

The workshop topics include (but are not limited to):

- semantic foundations of architectural and component-based design within UML
- compositional techniques for the analysis embedded and real-time systems in UML
- compositional model checking of UML behavioural models
- compositional deductive methods based on OCL
- methodologies based on compositional formal techniques

FORMAT OF THE WORKSHOP

The workshop will consist of presentations of the accepted papers, which will be the basis for an intensive discussion on the workshop topics. Extended abstracts of the presentations will be published after the workshop by Elsevier Science as a volume of the Electronic Notes in Theoretical Computer Science.

For an up-to-date program and invited talks see the workshop web-site http://fmco.liacs.nl/compuuml.html

SUBMISSIONS

Authors are invited to submit by August 25th an extended abstract not exceeding 20 pages electronically to F.S.de.Boer@cwi.nl. Submissions must be either in Postscript or PDF format and prepared for USLetter or A4 page sizes.

Submissions will be evaluated by the program committee for inclusion in the proceedings, which will be published by Electronic Notes in Theoretical Computer Science series. Papers must contain original contributions, be clearly written, and include appropriate reference to and comparison with related work. Simultaneous submissions to other conferences are not allowed.

IMPORTANT DATES

25 August Submission deadline
10 September Notification to authors
1 October Deadline for preliminary version
20 October Workshop date
21 November Deadline for final version

PROGRAM COMMITTEE

Frank de Boer (CWI, NL)
Marcello Bonsangue (LIACS, NL)
Warner Damm (OFFIS, DE)
Susanne Graf (Verimag, France)
David Harel (Weizmann Institute, Israel)
Jozef Hooman (University of Nijmegen, NL)
Bernhard Josko (OFFIS, DE)
Amir Pnueli (Weizmann Institute, ISR)
Willem-Paul de Roever (Kiel University, DE)
Joseph Sifakis (Verimag, FR)

SPONSORS

This workshop is sponsored by the European R&D project OMEGA – Correct Development of Real-time Embedded Systems (http://www-omega.imag.fr), and the German-Dutch project Mobi-J (main sponsor of FMCO, http://fmco.liacs.nl).
CALL FOR PARTICIPATION


Theme
Large and complex software systems provide the necessary infrastructure in all industries today. In order to construct such large systems in a systematic manner, the focus in the development methodologies has switched in the last two decades from functional issues to structural issues: both data and functions are encapsulated into software units which are integrated into large systems by means of various techniques supporting reusability and modifiability. This encapsulation principle is essential to both the object-oriented and the more recent component-based software engineering paradigms.
The objective of this symposium is to bring together researchers and practitioners in the areas of software engineering and formal methods to discuss the concepts of reusability and modifiability in component-based and object-oriented software systems.

Format
The symposium is a four days event in the style of the former REX workshops; organised to provide an atmosphere that fosters collaborative work, discussions and interaction. The program consists of keynote and technical presentations, and contains an exquisite social event.

Keynote presentations
Desmond D'Souza (Kinetium, Austin, USA)
E. Allen Emerson (University of Texas at Austin, USA)
Andrew D. Gordon (Microsoft Research, UK)
Yuri Gurevich (Microsoft Research, USA)
Tony Hoare (Microsoft Research, UK)
Yuri Gurevich (Microsoft Research, USA)
Tony Hoare (Microsoft Research, UK)
David Parnas (University of Limerick, IE)
Joseph Sifakis (Verimag, FR)

Technical presentations
Albert Benveniste (IRISA/INRIA - Rennes, FR)
Frank de Boer (CWI, NL)
Egon Boerger (Pisa University, IT)
Werner Damr (University of Oldenburg, DE)
Razvan Diaconescu (IMAR, RO)
Gregor Engels (University of Paderborn, DE)
Jose Luis Fiadeiro (University of Leicester, UK)
Jan Friso Groote (Eindhoven University of Technology, NL)
Jean-Marc Jezequel (IRISA, Rennes, FR)
Bengt Jonsson (Uppsala University, SE)
Yasme Lakhnech (University of Grenoble, FR)
Rob van Ommering (Philips Research Laboratories, NL)
Amir Pnueli (The Weizmann Institute of Science, ISR)
Willem-Paul de Roever (University of Kiel, DE)
Jan Rutten (CWI, Amsterdam, NL)
Philippe Schnoebelen (CNRS, Cachan, FR)
Natalia Sidorova (Eindhoven University of Technology, NL)
Heike Wehrheim (University of Oldenburg, DE)
Jeanette Wing (Carnegie Mellon University, USA)

Organizing committee
F.S. de Boer (CWI and Utrecht University)
M.M. Bonsangue (LIACS-Leiden University)
S. Graf (Verimag)
W.-P. de Roever (Christian-Albrechts University)

Speakers' contributions will be published after the workshop in a proceeding of Lecture Notes in Computer Science by Springer-Verlag.

Registration
Participation is limited to about 80 people, using a first-in first-served policy. To register, please fill in the registration form at http://fmco.liacs.nl/fmco03.html and submit it before September 15, 2003. The registration fee is 375 euro for regular participants and 250 euro for students. It includes the participation to the symposium, a copy of the proceedings, all lunches and refreshments, and a social event (with dinner).

For more information about participation and registration see the FMCO site above or consult either F.S. de Boer (frb@cwi.nl) or M.M. Bonsangue (marcello@liacs.nl).
The symposium invites papers on fundamental aspects of data management. Original research papers on the theory, design, specification, or implementation of data management tools are solicited. Papers emphasizing new topics or foundations of emerging areas are welcome. The symposium will be held at the Maison de la Chimie, Paris, in conjunction with the ACM SIGMOD Intl. Conference on Management of Data (SIGMOD’04).

Suggested topics include the following (this list is not exhaustive and the order does not reflect priorities):

- Access Methods & Physical Design
- Active Databases
- Complexity & Performance Evaluation
- Data Integration & Interoperability
- Data Mining
- Data Models
- Data Stream Management
- Database Programming Languages
- Databases & Information Retrieval
- Databases & Workflows
- Deductive Databases & Knowledge Bases
- Distributed Databases
- Information Processing on the Web
- Logic in Databases
- Multimedia Databases
- Object-oriented Databases
- Query Languages
- Query Optimization
- Real-time Databases
- Security & Privacy
- Semistructured Data & XML
- Spatial & Temporal Databases
- Theory of recovery
- Transaction Management
- Views & Warehousing
- Web Services & Electronic Commerce
- XML Databases

Important Dates:

- November 17, 2003: Paper abstracts due
- November 24, 2003: Full papers due
- February 9, 2004: Notification of acceptance/rejection
- March 15, 2004: Camera-ready due

Program Committee:

- Divyakant Agrawal, University of California at Santa Barbara, USA
- Michael Benedikt, Bell Labs, USA
- Thomas Eiter, Technische Universitat Wien, Austria
- Christos Faloutsos, Carnegie Mellon University, USA
- Wenfei Fan, Bell Labs, USA
- Johannes Gehrke, Cornell University, USA
- Martin Grohe, University of Edinburgh, United Kingdom
- Yannis E. Ioannidis, University of Athens, Greece
- Sanjeev Khanna, University of Pennsylvania, USA
- Yossi Matias, Tel Aviv University, Israel

PODS General Chair: Catriel Beeri, Hebrew University, Israel
PODS Program Chair: Dan Suciu, University of Washington, USA
PODS Proceedings Chair: Alin Deutsch, University of California at San Diego, USA
PODS Publicity Chair: Frank Neven, University of Limburg LUC, Belgium

Submission format and instructions:

Authors are required to submit a paper title and short abstract (about 100 words) before submitting the paper. The address, telephone number, FAX number, and e-mail address of the contact author should appear on the title page of the submission. Submissions should be limited to 10 pages (with font size at least 11 pts) and may consist of extended abstracts. Each submission must provide sufficient detail to allow the program committee to assess its merits and should include appropriate references to and comparisons with the literature.

It is recommended that each submission begins with a succinct statement of the problem, a summary of the main results, and a brief explanation of their significance and relevance to the conference, all suited for the best paper. The submission must include a technical report that is clearly marked appendix to be read at the discretion of the committee. Only electronic submissions are accepted. Instructions for (electronic) submissions will be posted on PODS's home page.

Awards:

Best Newcomer Award: An award will be given to the best submission, as judged by the program committee, written solely by authors who have never published in earlier PODS proceedings.

Best Paper Award: There will also be an award for the best of all papers submitted, as judged by the program committee. The program committee reserves the right to give both awards to the same paper, not to give an award, or to split an award among several papers. Papers authored or co-authored by PC members are not eligible for an award.

http://www.sciences.univ-nantes.fr/inr/SIGMODPODS04/
CALL FOR PAPERS
ICALP 2004
31st International Colloquium on Automata, Languages and Programming
July 12–16, 2004, Turku, Finland

Conference Chair
Juhani Karhumäki
Department of Mathematics and
Turku Centre for Computer Science
University of Turku
FIN-20014 Turku, Finland
email: karhumak@cs.utu.fi

Program Committee
Track A
A. Atserias, Barcelona (ES)
G. Brodal, Aarhus (DK)
J. Cassaigne, Marseille (FR)
J. Diaz, Barcelona (ES), chair
R. Fleischer, Hong Kong (HK)
H. Gabow, Boulder (US)
L. Goldberg, Warwick (UK)
J. Hromkovic,achen (DE)
G. Italiano, Roma (IT)
T. Jiang, Riverside (US)
C. Kaklamanis, Patras (GR)
J. Kari, Turku (FI)
C. Moore, Santa Fe (US)
P. Podelski, Prague (CZ)
P. Raghavan, Stanford (US)
M. Santha, Paris (FR)
B. Voelking, Dortmund (DE)
G. Woeginger, Twente (NL)
M. Yung, Columbia U. (US)

Track B
R.-J. Back, Turku (FI)
P.-L. Curien, Paris (FR)
A. Gordon, Microsoft, Cambridge (UK)
S. Hayashi, Kobe (JP)
T. Høegger, Berkeley (US)
M. Hofmann, Munich (DE)
B. Jacobs, Nijmegen (NL)
E. Moggi, Genova (IT)
C. Palamidessi, University Park, Penn. (US)
J. Pardo, Uppsala (SE)
B. Pierce, Philadelphia (US)
A. Rabinovich, Tel Aviv (IL)
D. Sangielski, Edinburgh (UK), chair
W. Thomas, Aachen (DE)
I. Walukiewicz, Bordeaux (FR)

The 31st International Colloquium on Automata, Languages and Programming sponsored by the European Association of Theoretical Computer Science will take place in Turku, on July 12–16, 2004. It is organized at Turku University by the Department of Mathematics and Turku Centre for Computer Science. ICALP04 is colocated with the 19th International Conference on Logic in Computer Science (LICS04). In addition, satellite workshops are planned.

Papers presenting original research on all aspects of theoretical computer science are sought. Typical but not exclusive topics of interest are:

Track A:
- Algorithmic aspects of parallel and distributed computing
- Algorithms and data structures
- Algorithms and models for large networks
- Algorithms for computationally hard problems
- Automata theory and formal languages
- Bioinformatics
- Computational complexity
- Combinatorics and structures in Computer Science
- Cryptography
- Machine learning
- Molecular computing, neural and evolutionary algorithms
- Proof complexity
- Quantum computing

Track B:
- Algebraic and categorical models
- Applications of automata in logic
- Concurrency, mobility and distributed systems
- Databases, semi-structured data and finite model theory
- Logics and their applications
- Principles of programming languages
- Program logics, formal methods and model checking
- Security analysis and verification
- Semantics of programming languages
- Specification, refinement and verification
- Type systems and typed calculi

Submissions:
Authors are invited to submit a paper of no more than 12 pages in LNCS-style, presenting original research on the theory of computer science, see http://www.math.utu.fi/ICALP04/ for submission details. Submissions should indicate which track the paper is submitted to. No simultaneous submission to other conferences with published proceedings is allowed. Accepted papers will be published in the Lecture Notes in Computer Science by Springer.

Contact address:
For further information see: http://www.math.utu.fi/ICALP04/ or contact: ICALP04@cs.utu.fi, karhumak@cs.utu.fi

Important Dates:
Workshop proposals: November 13, 2003
Submissions: January 26, 2004
Notification: March 31, 2004
Final version: April 27, 2004
AGENDA OF EVENTS

July 2003

SEKE 2003

AMoNet 2003

DLT 2003
Seventh International Conference on Developments in Language Theory, July 7 - 11, 2003, Szeged, Hungary. deadline: February 24, 2003, email: dlt03@inf.u-szeged.hu, www: http://www.inf.u-szeged.hu/informatika/dlt03/

DMTCS'03

EEF GCSS

PLS'03

CAV 2003

PSI'03
Andrei Ershov Fifth International Conference on Perspectives of System Informatics, July 9 - 12, 2003, Novosibirsk, Akademgorodok, Russia. deadline: February 2, 2003, email: psi03@iis.nsk.su, www: http://www.iis.nsk.su/PSI03

GECCO 2003
RV 2003

AAMAS 2003

PDMC 2003

SoftMC 2003

DALT 2003

CIAA 2003

WMC-Tarragona 2003

ECOOP 2003

FTfIP 2003

WADS 2003

August 2003

FCT 2003
CRYPTO 2003

ESSLL'I 2003
15th European Summer School in Logic, Language and Information, August 18 - 29, 2003, Vienna, Austria. www: http://www.science.uva.nl/~bcate/esslli03

ISMP 2003

CSL'03 & KGK

ICFP 2003

MFCS 2003

PPDP 2003

CCA 2003

September 2003

EXPRESS 2003

FOCLASA 2003

CONCUR 2003
14th International Conference on Concurrency Theory, September 3 - 5, 2003, Marseille, France. deadline: April 4, 2003, www: http://concur03.univ-mrs.fr/
KES 2003

CMCIM 2003
Workshop on Categorical Methods for Concurrency, Interaction and Mobility, September 6 - 6, 2003, Marseille, France. Satellite event to CONCUR 2003. deadline: June 1, 2003. www: http://www.mcs.le.ac.uk/events/cmcim03/

FORMATS 2003

GETCO 2003

SPV 2003

Eurocomb 2003

TPHOLs 2003

WORDS'03

ESA 2003

EDOC 2003

GD 2003
**LPAR 2003**

**SAGA 2003**

**SFM-03-SA**

**October 2003**

**ISC 2003**

**EMSOFT 2003**

**ESORICS 2003**

**NORDSEC 2003**

**APLAS 2003**

**November 2003**

**FMCO 2003**

**FMOODS 2003**

**SEEFM 2003**
December 2003

ICLP 2003

FSTTCS 2003

HiPC 2003

January 2004

SOFSEM 2004

February 2004

FoIKS 2004

June 2004

SIGMODPODS 2004

July 2004

ICALP 2004
The aim of this section is to help us to become acquainted with the contents of Ph.D. work done worldwide in the field of theoretical computer science. Each abstract submitted to this section must conform to the following (rather strict) editorial scheme. One must explicitly provide the following items of information (in this order and preceded by these explicit headlines): Author, Title, Language of presentation, Promotor, Date of defence, Institution granting degree, Abstract (between two and three pages), Table of contents and Author's correspondence address.

The Editor

Author  Jason Crampton  
Title  Authorization and Antichains  
Language of presentation  English  
Supervisor  Professor George Loizou  
Date of defence  15 April 2002  
Institute granting degree  London University, United Kingdom

Abstract

Access control has been an important issue in military systems for many years and is becoming increasingly important in commercial systems. There are three important access control paradigms: the Bell-LaPadula model, the protection matrix model and the role-based access control model. Each of these models has its advantages and disadvantages. Partial orders play a significant part in the role-based access control model and are also important in defining the security lattice in the Bell-LaPadula model. The main goal of this thesis is to improve the understanding and specification of access control models through a rigorous mathematical approach.

We examine the mathematical foundations of the role-based access control model and conclude that antichains are a fundamental concept in the model. The analytical approach we adopt enables us to identify where improvements in the administration of role-based access control could be made. We then develop a new administrative model for role-based access control based on a novel, mathematical interpretation of encapsulated ranges. We show that this model supports discretionary access control features which have hitherto been difficult to incorporate into role-based access control frameworks.

Separation of duty is an important feature of role-based access control models that has usually been expressed in first-order logic. We present an alternative formalism for separation of duty policies based on antichains in a powerset (Sperner families), and show that it is no less expressive than existing approaches. The simplicity of the formalism enables us to analyze the complexity of implementing separation of duty policies. In the course of this analysis we establish new results about Sperner families.
We also define two orderings on the set of antichains in a partially ordered set and prove that in both cases the resulting structure is a distributive lattice. This lattice provides the formal framework for a family of secure access control models which incorporate the advantages of existing paradigms without introducing many of their respective disadvantages. We present two members of this family: a new model for role-based access control, for which we give an operational semantics and prove a security theorem similar to the Basic Security Theorem for the Bell-LaPadula model; and the secure hierarchical protection matrix model which combines the strong security properties of the Bell-LaPadula model with the flexibility of the protection matrix model.

Table of Contents

Abstract 2
Contents 5
List of Figures 7
List of Tables 9
List of Commands 10
Acknowledgements 12

1 Introduction 13
1.1 Access control 14
1.2 Partial orders 17
1.3 Outline of the thesis 17

2 Preliminaries 21
2.1 Partial orders 21
2.2 Combinatorial considerations 26
2.3 Access control models 31
2.4 Complexity theory 38
2.5 Turing machines 40

3 Role-based Access Control 41
3.1 RBAC1 44
3.2 Notation for role-based access control 47
3.3 RBAC2 49
3.4 ARBAC97 52
3.5 Other role-based access control models 62
3.6 Computational complexity of role-based access control 66
3.7 Conclusion 68

4 Extensions to Role-based Access Control 70
4.1 Motivation 70
4.2 The RHA family of administrative models 73
4.3 SARBAC 86
4.4 Using SARBAC for discretionary access control 92
4.5 The generalized role-based access control model 94
4.6 Summary and discussion 97

5 The Safety Problem 99
5.1 The protection matrix model 99
Abstract

Traversal sequences were defined by Aleliunas et al. (1979) as a tool for the study of undirected s-t-connectivity. In the first part of this thesis we study traversal sequences. We improve on previous results and we present a simple construction of polynomial length universal traversal sequences for 2-regular graphs. These universal traversal sequences are log-space (even $NC^1$) constructible and are of length $O(n^{4.03})$.

Further, we introduce a new notion of traversal sequences that we call exploration sequences. Exploration sequences share many properties with the original traversal sequences but they also exhibit some new properties. For instance, they have an ability to backtrack, and their random properties are robust under choice of the probability distribution on labels. We present simple constructions of polynomial length universal exploration sequences for several classes of graphs, e.g., 2-regular graphs, cliques, expanders, trees. These constructions do not obey previously known lower bounds on the length of universal traversal sequences thus, they highlight another difference between exploration and traversal sequences. We also show that universal traversal sequences can be efficiently converted into universal exploration sequences. Further, we show certain self-correcting properties of traversal and exploration sequences and we propose a candidate universal exploration sequence.

In the second part of this thesis we study Kolmogorov complexity, which is a measure of randomness of finite strings, and its resource-bounded variants. We show that sets consisting of strings of high resource-bounded Kolmogorov complexity provide examples of sets that are complete for complexity classes in which they naturally lie under probabilistic and non-uniform reductions whereas they are provably not complete under the usual many-one reductions.

Further, our results imply that under certain intractability assumptions, resource-bounded Kolmogorov complexity of a string as well as circuit size, formula size and branching program size of a Boolean function can not be efficiently approximated.

We also study unbounded Kolmogorov complexity. We show completeness results in that setting and point out certain inconsistencies caused by the usual definition of Kolmogorov complexity. We also consider the problem of determin-
istically generating a Kolmogorov random string when the set of Kolmogorov random strings is given to us as an oracle.

Table of Contents:

1 Introduction 1

I Traversal Sequences 3

2 Introduction 4

3 The s-t-connectivity Problem 7
   3.1 Definitions and Basic Properties 7
   3.1.1 Three Variants of s-t-connectivity 10
   3.2 Complexity of Undirected s-t-connectivity 12
      3.2.1 Derandomization 16
      3.2.2 Other Upper Bounds on s-t-connectivity 21
      3.2.3 Other Computational Models 22

4 Traversal Sequences 24
   4.1 Universal Traversal Sequence for Cycles 29
      4.1.1 Preliminaries 30
   4.2 An $n^{O(\log n)}$ Universal Traversal Sequence 32
      4.2.1 A Parity Contraction 32
      4.2.2 A Pair Contraction 34
      4.2.3 Putting Them Together 35
   4.3 An $O(n^c)$ Universal Traversal Sequence 36
      4.3.1 1-run and 0-run Breaking 37
      4.3.2 01-run Breaking 49
      4.3.3 The Construction 57
   4.4 The Analysis 59
      4.4.1 The Modified Algorithm 59
      4.4.2 Further Improvements 64

5 Exploration Sequences 65
   5.1 Definitions, Basic Properties, and Explicit Constructions 66
   5.2 Unbalanced Probability Distributions on Labels 72
   5.3 Traversal versus Exploration 78
   5.4 Self-correcting Properties 84
   5.5 Conclusions 88
II  Kolmogorov complexity 90

6  Introduction 91
  6.1 Focus of Our Study .......................... 92
  6.2 Techniques .................................. 95

7  Resource-bounded Kolmogorov Complexity 97
  7.1 Definitions .................................. 97
  7.2 Reductions .................................... 99
  7.3 Variants of Kolmogorov Complexity ............. 100

8  Complexity of $R_{Kt}$ and $R_{KS}$ 107
  8.1 Tools in Derandomization ..................... 108
  8.2 Hardness Results ............................. 110
  8.3 Derandomization ................................ 114

9  Hardness of $R_K$ 119
  9.1 Inside $P^{R_K}$ .............................. 121

10 Hardness of Approximation 131

11 Conclusions 136

Author’s Correspondence Address:

Michal Koucký
Novodvorská 1118/155
142 00 Prague
Czech Republic

email: nikoucky@acm.org
www: http://paul.rutgers.edu/~mkoucky
58 The complementary of square-words

Let $A = \{a, b\}$. It is easy to show that $SQUARE = \{uv|u \in A^*\}$ is not context-free (intersection with the rational language $a^*b^*a^*b^*$). What about the complementary of $SQUARE$?

59 $H$ like Hanoi and Hamilton

You all know about the Hanoi game with 3 staks and $n$ disks. Let us keep the rules and change the aim: There are 3* legal configurations. Show there is a sequence of legal moves to go through each legal configuration and to come back to the initial configuration, that is, there is a Hamiltonian loop in the graph whose vertices are configurations and edges are legal moves. Show that moreover, there is only one such look (of course, up to shifting the of initial point, and up to backward symmetry).

Solutions to previous puzzles

55 Context-free intersections with $\{a^n b^n | n \geq 0\}$

If $L$ and $L'$ are context-free languages, it might happen that $L \cap L'$ is not context-free (usual counter-example: $L = \{a^n b^n c^n|n \geq 0\}$ and $L' = \{a^m b^m c^m|p, m \geq 0\}$). What if $L'$ is not any context-free language, but is $\{a^n b^n|n \geq 0\}$?

Solution: If $L$ is context-free, then $L \cap \{a^n b^n|n \geq 0\}$ is also context-free.

Lemma 1: Let $M$ be a context-free language over an alphabet with only one letter $a$, then there are an integer $N$ and integers $t_1, \ldots, t_k, j_1, \ldots, j_l$ such that $M = a^{t_1} + \ldots + a^{t_k} + (a^j_1 + \ldots + a^{j_l})(a^N)^*$. Proof left to the reader (hint: use the iteration lemma), or to be found in books.

Lemma 2: Let $(p_1, q_1), (p_2, q_2)$ and $(p_3, q_3)$ be in $\mathbb{N}^2$, then there are indices $i, j, k = \{1, 2, 3\}$, a positive integer $n$, non-negative integers $r$ and $s$ such that $n(p_i, q_i) = r(p_j, q_j) + s(p_k, q_k)$. Proof left to the reader (hint: $(p_1, q_1), (p_2, q_2)$ and $(p_3, q_3)$ are linearly dependent in the Q-field $\mathbb{Q}^2$).

Definition 3: Let $H$ be the monoid $(\mathbb{N}^2, +)$. We will use the standard notations $X + Y = X \cup Y, X.Y = \{(x + x', y + y')|(x, y) \in X, (x', y') \in Y\}$ and $X^* = \cup_i X^i$ while working with the monoid $(P(H), +)$.

Lemma 4. Rational languages over $H$ are the finite unions of sets $K_{p_0,p_1,p_2,q_0,q_1,q_2} = \{(p_0 + p_1 k + p_2 k', q_0 + q_1 k + q_2 k' + q_0 k''|k, k', k'' \in \mathbb{N}\}$ where $p$'s and $q$'s are non-negative integers. Note that singletons and $\{(p_0 + p_1 k, q_0 + q_1 k|k \in \mathbb{N}\}$ are such sets since $p$'s and $q$'s may be 0.

Finite unions of $K$'s are rational: proof left to the reader (obvious, it will not be useful to us anyway).

Rational languages over $H$ are finite unions of $K$'s: We must prove that finite unions of $K$'s contain singletons (obvious) and are stable by union (obvious), product and star. Due to the distributivity of product over sum, and due to fact that $H$ is commutative, so that $(X + Y)^* = X^* Y^*$, all what is needed to be proved is that the product of two $K$'s, and the star of a $K$, are finite unions of $K$'s.

$K_{p_0,p_1,p_2,q_0,q_1,q_2} = \{(0, 0)\} + \{(p_0 + p_1 k + p_2 k', q_0 + q_1 k + q_2 k' + q_0 k''|k, k', k'' \in \mathbb{N}\}$.

Let us use lemma 2, which says for example that $n(p_3, q_3) = r(p_3, q_3) + s(p_3, q_3)$. Then $K_{p_0,p_1,p_2,q_0,q_1,q_2} = \{(0, 0)\} + \bigcup_{i=0}^{\infty} \{(p_0 + i \ast p_1 + p_2 k' + q_0 k''|a(q_1 + i \ast q_1) + q_2 k'' + q_0 k''|k, k'' \in \mathbb{N}\}$ which proves the result.

The technique is similar for the product.

Lemma 5. $\Gamma = \{n|\langle n, n \rangle \in K_{p_0,p_1,p_2,q_0,q_1,q_2}\}$ is either finite or eventually periodic.

Assume indeed it is not empty. Let $n_\in \in \Gamma$ and $k, k'$ be such that $n x = p_0 + p_1 k x + p_2 k' x = q_0 + q_1 k x + q_2 k' x$. Let $a$ and $b$ such that $a b = 1$ and $a p_1 = q_1 + b(p_2 - q_2) = 0$. Note: $p_1 = q_1$ and $p_2 = q_2$ is a special case that must be treated separately. Left to the reader. Useful lemma: Let $x = gcd(p_1, p_2)$, then for large enough $y$, there

Readers are invited to send comments, and to send exercises, even if they don't know the answer. Write to Laurent.Rosaz@lri.fr.
are non-negative integers \(r, s\) such that \(r \cdot p_1 + s \cdot p_2 = y \cdot z\). We get that \(p_1 + p_2 k + p_2 k' = q_0 + q_1 k + q_2 k'\) iff \((k, k') = (k_X, k_Y) + \delta(a, b)\) for some integer \(\delta \in \mathbb{Z}\). So that \(\Gamma = \{n + \delta(a \cdot p_1 + b \cdot p_2) | k_X + \delta \cdot a \geq 0 \land k_Y + \delta \cdot b \geq 0\}\) which induces the result.

Proof of the solution:
We may assume that \(L \subseteq a^*b^*\). Indeed, if not, one can replace \(L\) with \(L \cap a^*b^*\), which will be also context-free and which will have the same intersection with \(\{a^nb^m | n \geq 0\}\) than \(L\).

\(L\) is context-free, so there is a grammar \(G\) to generate it.

Change \(G\) to \(G'\) so that every rule \(r\) can be written \(X \rightarrow a^rYb^s\) or \(X \rightarrow a^r\).

To do so, clean the grammar \(G\) to get rid of useless symbols, then observe that each rule \(r\) of \(G\) can be written \(X \rightarrow p.Ys\) or \(X \rightarrow p.s\) where \(p\) and \(s\) are words over terminals and non terminals such that every word generated by \(p\) is in \(a^*\) and every word generated by \(s\) is in \(b^*\). If \(p\) is not already some \(a^r\), use lemma 1 with \(M\) the language generated by \(p\) to replace rule \(r\) (say \(X \rightarrow pYs\)) by rules \(X \rightarrow a^nYs\), rule \(X \rightarrow Z\), rule \(Z \rightarrow a^NZ\) and rules \(Z \rightarrow a^hYs\), where \(Z\) is a new non-terminal. Proceed similarly to lemma 2 with \(s\).

From \(G'\), we build the automaton \(A\) over the monoid \(H\), whose states are the set of non-terminals of \(G\) plus a special state \(end\). The initial state is the starting symbol of the grammar, \(end\) is the only final state. There is an arrow labeled by \((p, q)\) from \(X\) to \(Y\) iff \(X\) and \(Y\) are non terminals and \(X \rightarrow a^rYb^s\) is a rule of \(G'\), or \(X\) is a non terminal, \(Y = end\) and \(X \rightarrow p.Ys\) is a rule of \(G'\).

Observe that \(a^r b^s\) is generated by the grammar iff \((\alpha, \beta)\) is recognized by the automaton. From the automaton, using Kleene's technique, we can get a rational expression over \(H\) to describe \(\{(\alpha, \beta) | a^r b^s \in L\}\).

Thanks to lemma 4, that latter set is a finite union of \(K\)'s, which induces the result by use of lemma 5.

### 56 Linear detection of perfect binary trees

A complete binary tree is a tree where every node has either two sons (interior nodes) or none (leafs). The tree is perfect if the depth of every leaf is the same. One wants a linear algorithm to find if a complete tree is perfect. Find several ones. (Two of them are similar and are natural in terms of algorithmic. There are yet at least another three.)

Solution:
Classical solutions are the following ones:

1. Compute the depth of the leftmost leaf (the depth of the tree will do as well), then check every leaf is at that depth

1'. Make a recursive search of the tree, remind the depth of the first encountered leaf, check the other ones are at that depth.

2. Make a function which return -1 if not all the leaves are at the same depth, the depth of the leaves otherwise.

Less classical are the following solutions:

3. Compute the depth, that is the depth of the deepest leaf, then the anti-depth, that is the depth of the least deep leaf (same code, just min instead of max) and check there are equal.

4. Compute the depth, the number of leafs and check there are \(2^{\text{depth}}\) leaves (To do so without trouble, I suggest check(d,n) = if d=0 then return(n=1) else return(even(n) and check(d-1,n div 2)).

5. For trees which are not leaves, check for the left tree, then check the two subtrees are the same (Use the fact that the complexity of \(\text{AreTheSame(Tree1,Tree2)} = O(\text{size(Tree2)})\) to justify the algorithm is linear)

### 57 Paving an hexagon with diamonds

A regular hexagon with sides of length \(n\) is obviously paved by equilateral triangles with sides of length 1 (using \(6n^2\) of them). The hexagon is paved by diamonds, each diamond being the gluing of two adjacent such triangles. There are three possible diamond orientations. Show that the pavement uses as many diamonds of each orientation.
The hexagon \((n = 3)\)

Solution: The key idea is that such a pavement is the perspective sight of a heap of small cubes inside a box of size \(n\) with only the floor and two walls. The point is that you will see the same surface if you look from above, from the left, or from the right.

Of course, we must take care that we are not sure yet every pavement is such a drawing, but the idea leads the proof.

Note that the replacement in a pavement of \(\square\) by \(\square\) keeps the number of diamonds of each direction.

So do that replacement until it is not possible any more (the replacements have to stop because the sum of the height of the "top" diamonds is decreasing) Then check that the only pavement without the "cube drawing" draws an "empty box". Conclude.
European Association for Theoretical Computer Science
COUNCIL OF EATCS

BOARD

PRESIDENT: M. NIELSEN, Denmark
VICE PRESIDENTS: J. VAN LEEUWEN, The Netherlands
P. SPIRAKIS, Greece
TREASURER: D. JANSENS, Belgium
SECRETARY: B. ROVAN, Slovakia
BULLETIN EDITOR: G. ROZENBERG, The Netherlands
TCS EDITORS: G. AUSIELLO, Italy
D. SANNELE, UK
M.W. MISLOWE, U.S.A.
MONOGRAPHS EDITORS: W. BRAUER, Germany
G. ROZENBERG, The Netherlands
A. SALOMAA, Finland
WEB COORDINATOR: M. NIELSEN, Denmark

FURTHER COUNCIL MEMBERS

S. ARNBOG, Sweden
R. BAEZA-YATES, Chile
C. CALUDE, New Zealand
P. DEGANO, Italy
M. DEZANI-CIANCAGLINI, Italy
H. EHRI, Germany
J. ESPARZA, UK
Z. FULOF, Hungary
H. GABOW, U.S.A.
A. GIBBONS, UK
R. GORRERI, Italy
M. HENNESSY, UK
K. IWAMA, Japan
J.-P. JOUANNAUD, France
J. KARHUMAKI, Finland
K. MEHLHORN, Germany
M. NIVAT, France
G. PAUN, Romania
J. ROLIM, Switzerland
J. SGALL, Czech Republic
P.S. THIAGARAJAN, India
W. THOMAS, Germany
D. WAGNER, Germany
O. WATANABE, Japan
E. WIEZEL, Switzerland
J. WIEDERMANN, Czech Republic
G. WOEGERING, Austria
S. YU, Canada
EATCS

HISTORY AND ORGANIZATION

EATCS is an international organization founded in 1972. Its aim is to facilitate the exchange of ideas and results among theoretical computer scientists as well as to stimulate cooperation between the theoretical and the practical community in computer science.

Its activities are coordinated by the Council of EATCS, out of which a President, Vice Presidents, a Treasurer and a Secretary are elected. Policy guidelines are determined by the Council and the General Assembly of EATCS. This assembly is scheduled to take place during the annual International Colloquium on Automata, Languages and Programming (ICALP), the conference of EATCS.

MAJOR ACTIVITIES OF EATCS

- Organization of ICALP's
- Publication of the "Bulletin of the EATCS"
- Publication of the "EATCS Monographs" and "EATCS Texts"
- Publication of the journal "Theoretical Computer Science"
- Other activities of EATCS include the sponsorship or the cooperation in the organization of various more specialized meetings in theoretical computer science. Among such meetings are:
  CAAP (Colloquium on Trees in Algebra and Programming), TAPSOFT (Conference on Theory and Practice of Software Development), STACS (Symposium on Theoretical Aspects of Computer Science), MFCS (Mathematical Foundations of Computer Science), LICS (Logic in Computer Science), ESA (European Symposium on Algorithms), Conference on Structure in Complexity Theory, SPAA (Symposium on Parallel Algorithms and Architectures), Workshop on Graph Theoretic Concepts in Computer Science, International Conference on Application and Theory of Petri Nets, International Conference on Database Theory, Workshop on Graph Grammars and their Applications in Computer Science.

BENEFITS

Benefits offered by EATCS include:
- Receiving the "Bulletin of the EATCS" (over 1000 pages per year)
- Reduced registration fees at various conferences
- Reciprocity agreements with other organizations
- 25% discount in purchasing ICALP proceedings
- 25% discount in purchasing books from "EATCS Monographs" and "EATCS Texts"
- Discount (about 70 %) per individual annual subscription to "Theoretical Computer Science".
- Discount (about 70 %) per individual annual subscription to "Fundamenta Informaticae".

(1) THE ICALP CONFERENCE

ICALP is an international conference covering all aspects of theoretical computer science and now customarily taking place during the second or third week of July.

Typical topics discussed during recent ICALP conferences are: computability, automata theory, formal language theory, analysis of algorithms, computational complexity, mathematical aspects of programming language definition, logic and semantics of programming languages, foundations of logic programming, theorem proving, software specification, computational geometry, data types and data structures, theory of data bases and knowledge based systems, cryptography, VLSI structures, parallel and distributed computing, models of concurrency and robotics.
Sites of ICALP meetings:
- Paris, France 1972
- Saarbrücken, Germany 1974
- Edinburgh, Great Britain 1976
- Turku, Finland 1977
- Udine, Italy 1978
- Graz, Austria 1979
- Noordwijkerhout, The Netherlands 1980
- Haifa, Israel 1981
- Aarhus, Denmark 1982
- Barcelona, Spain 1983
- Antwerp, Belgium 1984
- Nafplion, Greece 1985
- Rennes, France 1986
- Karlsruhe, Germany 1987
- Tampere, Finland 1988
- Stresa, Italy 1989
- Warwick, Great Britain 1990
- Madrid, Spain 1991
- Wien, Austria 1992
- Lund, Sweden 1993
- Jerusalem, Israel 1994
- Szeged, Hungary 1995
- Paderborn, Germany 1996
- Bologna, Italy 1997
- Aalborg, Denmark 1998
- Prague, Czech Republic 1999
- Genève, Switzerland 2000
- Heraklion, Greece 2001
- Malaga, Spain 2002
- Eindhoven, The Netherlands 2003
- Turku, Finland 2004

(2) THE BULLETIN OF THE EATCS

Three issues of the Bulletin are published annually appearing in February, June and October respectively.

The Bulletin is a medium for rapid publication and wide distribution of material such as:
- EATCS matters
- Information about the current ICALP
- Technical contributions
- Columns
- Surveys and tutorials
- Reports on conferences
- Calendar of events
- Reports on computer science departments and institutes
- Listings of technical reports and publications
- Book reviews
- Open problems and solutions
- Abstracts of Ph.D. Theses
- Information on visitors at various institutions
- Entertaining contributions and pictures related to computer science.

Contributions to any of the above areas are solicited. All written contributions should be sent to the Bulletin Editor:
Prof. Dr. G. Rosenberger
LIACS, University of Leiden
P.O. Box 9512, 2300 RA Leiden, The Netherlands
Email: rosenber@liacs.nl
Fax: +31-71-5276985

Deadlines for submissions to reach the Bulletin Editor are: January 15, May 15 and September 15 for the February, June and October issue respectively.

All pictures (preferably black and white) including text of what they are showing should be sent to the Picture Editor:
Prof. Dr. M. Kudlek
University of Hamburg
Department of Computer Science
Vogt-Köln Strasse 30
D-22527 Hamburg, Germany

Deadlines are 2 weeks before those for written contributions, indicated above.

(3) EATCS MONOGRAPHS AND TEXTS

This is a series of monographs published by Springer-Verlag and launched during ICALP 1984; more than 40 volumes appeared. The series includes monographs in all areas of theoretical computer science, such as the areas considered for ICALPs. Books published in this series present original research or material of interest to the research community and graduate students. Each volume is normally a uniform monograph rather than a compendium of articles. The series also contains high-
level presentations of special topics. Nevertheless, as research and teaching usually go hand in hand, these volumes may still be useful as textbooks, too. Text published in this series are intended mostly for the graduate level. Typically, an undergraduate background in computer science is assumed. However, the background required may vary from topic to topic, and some books may be self-contained. The texts cover both modern and classical areas with an innovative approach that may give them additional value as monographs. Most books in this series will have examples and exercises.

Texts in Theoretical Computer Science. An EATCS Series

The editors of the series are W. Brauer (Munich), G. Rozenberg (Leiden), and A. Salomaa (Turku). Potential authors should contact one of the editors. The advisory board consists of G. Ausiello (Rome), S. Even (Haifa), J. Hartmanis (Ithaca), N. Jones (Copenhagen), M. Nivat (Paris), C. Papadimitriou (Athens and San Diego), and D. Scott (Pittsburgh).

EATCS Monographs and Texts is a very important EATCS activity and its success depends largely on our members. If you are a potential author or know one please contact one of the editors.

An EATCS member can purchase books from the series with 25% discount. Your order should be sent to:
Prof. Dr. G. Rozenberg
LIACS, University of Leiden
P.O. Box 9512, 2300 RA Leiden, The Netherlands
who after acknowledging your EATCS membership forwards the order for further processing to Springer Verlag – they handle the whole matter further on.

Updated information about the series can be obtained from the publisher, Springer-Verlag.

(4) THEORETICAL COMPUTER SCIENCE

The journal *Theoretical Computer Science*, founded in 1975, is published by Elsevier Science Publishers, Amsterdam, currently in 20 volumes (40 issues) a year. Its contents are mathematical and abstract in spirit, but it derives its motivation from practical and everyday computation. Its aim is to understand the nature of computation and, as a consequence of this understanding, provide more efficient methodologies.

All kinds of papers, introducing or studying mathematical, logical and formal concepts and methods are welcome, provided that their motivation is clearly drawn from the field of computing.

Papers published in TCS are grouped in two sections according to their nature. One section, "Algorithms, automata, complexity and games", is devoted to the study of algorithms and their complexity using analytical, combinatorial or probabilistic methods. It includes the whole fields of abstract complexity (i.e., all the results about the hierarchies that can be defined using Turing machines), of automata and language theory (including automata on infinite words and infinitary languages), of geometrical (graphic) applications and of system performance using statistical models.

A subsection is the Mathematical Games Section, which is devoted to the mathematical and computational analysis of games.

The other section, "Logic, semantics and theory of programming", is devoted to formal methods to check properties of programs or implement formally described languages; it contains all papers dealing with semantics of sequential and parallel programming languages. All formal methods treating these problems are published in this section, including rewriting techniques, abstract data types, automatic theorem proving, calculi such as SCP or CCS, Petri nets, new logic calculi and developments in categorical methods.

The Editors-in-Chief of "Theoretical Computer Science" are:
G. Ausiello, Univ. di Roma "La Sapienza", Dip. Inform. e Sistemi, via Salaria 113, 00198 Roma, Italy
D. Sannella, Univ. of Edinburgh, Lab. for Foundations of Computer Science, Division of Informatics, King's Building, Mayfield Road, Edinburgh, EH9 3JZ, UK
M.W. Mislove, Tulane Univ., Dept. of Mathematics, New Orleans, LA 70118, USA

ADDITIONAL INFORMATION

Please contact the Secretary of EATCS:
Prof. Dr. Branislav Rovan
Department of Computer Science, Comenius University,
SK-84248 Bratislava, Slovakia
Email: secretary@eatcs.org or see http://www.eatcs.org
DUES
The dues are 25 zł for a period of one year. If the initial membership payment is received in the period December 21 – April 20, April 21 – August 20 or August 21 – December 20, then the first membership year will start on June 1, October 1 or February 1, respectively. Every contribution payment continues the membership for the same time period. In order to encourage double registration, we are offering a discount: SIGACT members can join EATCS for 20 zł per year.
Additional 13 zł fee is required for ensuring the air mail delivery of the EATCS Bulletin outside Europe.

HOW TO JOIN EATCS
To join, send the annual dues, or a multiple thereof (to cover a number of years), to the Treasurer of EATCS:
Prof. Dr. D. Janssens
University of Antwerp (UIA), Dept. of Mathematics and Computer Science
Universiteitsplein, 1
B-2610 Wilrijk, Belgium
Email: treasurer@eatcs.org
Fax: (+32) 3 820 2421

The dues can be paid (in order of preference) by VISA or EUROCARD / MASTERCARD credit card, by cheques in zł, cheques in other currency or convertible currency cash. When submitting your payment, please make sure to indicate your complete name and address. For this purpose you may use the form below.

Transfers of larger amounts may be made via the following bank account, on the condition that an additional 5 zł is paid for each transfer to cover the bank charges. If you use this way of payment, please send the necessary information (reason for the payment, name and address) to the treasurer.

Our account is:
Generale Bank Antwerpen
Antwerp (Wilrijk), Belgium
Account number: 220-0596350-30-01130
IBAN code: BE 15 2200 5963 5030

I would like to join EATCS / renew my EATCS membership and pay

.......................... as membership fee for .... years (and .... for air mail delivery)

☐ Please charge my credit card: Total amount: .........................

Card Number (16 digits): .............................................................

☐ EUROCARD/MASTERCARD ☐ VISA Exp. Date: ......................

Date/Signature Address (if different from EATCS mailing address below)
☐ Check Enclosed ☐ Transfered to EATCS account ☐ Cash Enclosed

Name: First name:
Address:

Date: Signature: