ISSN 0252–9742

Bulletin

of the

European Association for Theoretical Computer Science

EATCS

Number 90 October 2006

Council of the European Association for Theoretical Computer Science

Past Presidents:

EATCS Council Members

EMAIL ADDRESSES

Luca Aceto	luca@ru.is
Giorgio Ausiello	ausiello@dis.uniroma1.it
Wilfried Brauer	brauer@informatik.tu-muenchen.de
Josep Díaz	diaz@lsi.upc.es
Zoltán Ésik	ze@inf.u-szeged.hu
Giuseppe F. Italiano	italiano@disp.uniroma2.it
Dirk Janssens	Dirk.Janssens@ua.ac.be
Jean-Pierre Jouannaud	jouannaud@lix.polytechnique.fr
Juhani Karhumäki	karhumak@cs.utu.fi
Richard E. Ladner	ladner@cs.washington.edu
Jan van Leeuwen	jan@cs.uu.nl
Eugenio Moggi	moggi@disi.unige.it
Michael Mislove	mwm@math.tulane.edu
Mogens Nielsen	mn@brics.dk
Catuscia Palamidessi	catuscia@lix.polytechnique.fr
David Peleg	peleg@wisdom.weizmann.ac.il
Jiří Sgall	sgall@math.cas.cz
Branislav Rovan	rovan@fmph.uniba.sk
Grzegorz Rozenberg	rozenber@liacs.nl
Arto Salomaa	asalomaa@utu.fi
Don Sannella	dts@dcs.ed.ac.uk
Vladimiro Sassone	vs@ecs.soton.ac.uk
Paul Spirakis	spirakis@cti.gr
Andrzej Tarlecki	tarlecki@mimuw.edu.pl
Wolfgang Thomas	thomas@informatik.rwth-aachen.de
Ingo Wegener	ingo.wegener@uni-dortmund.de
Emo Welzl	emo@inf.ethz.ch
Gerhard Wöeginger	g.j.woeginger@math.utwente.nl
Uri Zwick	zwick@post.tau.ac.il

Bulletin Editor: Vladimiro Sassone, Southampton, United Kingdom
Cartoons: DADARA, Amsterdam, The Netherlands

The bulletin is entirely typeset by pdfTeX and ConTeXt in TXfonts. The Editor is grateful to Uffe H. Engberg, Hans Hagen, Marloes van der Nat, and Grzegorz Rozenberg for their support.

All contributions are to be sent electronically to

`bulletin@eatcs.org`

and must be prepared in LaTeX 2_{ε} using the class `beatcs.cls` (a version of the standard LaTeX 2_{ε} `article` class). All sources, including figures, and a reference PDF version must be bundled in a ZIP file.

Pictures are accepted in EPS, JPG, PNG, TIFF, MOV or, preferably, in PDF. Photographic reports from conferences must be arranged in ZIP files layed out according to the format described at the Bulletin's web site. Please, consult `http://www.eatcs.org/bulletin/howToSubmit.html`.

We regret we are unfortunately not able to accept submissions in other formats, or indeed submission not *strictly* adhering to the page and font layout set out in `beatcs.cls`. We shall also not be able to include contributions not typeset at camera-ready quality.

The details can be found at `http://www.eatcs.org/bulletin`, including class files, their documentation, and guidelines to deal with things such as pictures and overfull boxes. When in doubt, email `bulletin@eatcs.org`.

Deadlines for submissions of reports are January, May and September 15th, respectively for the February, June and October issues. Editorial decisions about submitted technical contributions will normally be made in 6/8 weeks. Accepted papers will appear in print as soon as possible thereafter.

The Editor welcomes proposals for surveys, tutorials, and thematic issues of the Bulletin dedicated to currently hot topics, as well as suggestions for new regular sections.

The EATCS home page is `http://www.eatcs.org`

Table of Contents

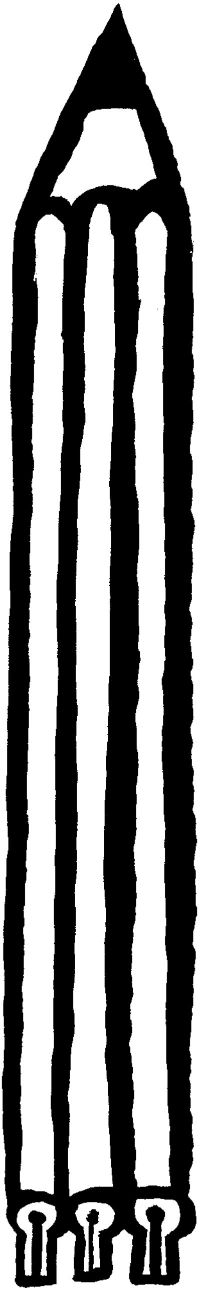
EATCS MATTERS

EATCS NEWS

THE EATCS COLUMNS

EATCS Matters

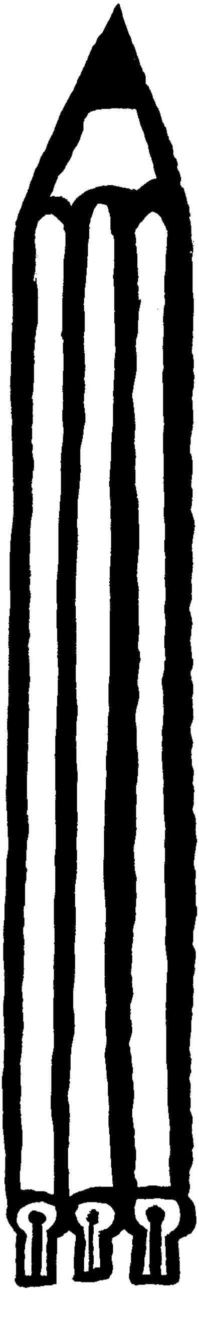
Letter from the President



Dear EATCS members,

Last July, in Venice, the Council has elected me as new EATCS President for the next two years term. Although a little frightened at the beginning, I confess that I am now very pleased and honoured to have the chance to serve our Community and, more generally, to devote my experience to strengthen and expand the role of our Association for the benefit of theoretical computer science. To the former President Mogens Nielsen, who has dedicated so much effort to EATCS in the past four years and who has contributed so successfully to the growth of the Association, a warm thank from all of us.

This year, ICALP has been, as usual, a very successfull event. Our flagship conference was accompanied by nine very interesting workshops and by three well-established conferences: PPDP, LOPSTR, CSFW, spanning from declarative programming to program synthesis, to formal aspects of security. More than four hundred participants attended the various scientific events and enjoyed the charming athmosphere and the colours of the Laguna. We wish to thank once more Michele Bugliesi and his team for the perfect organization and the program Chairs of the three Tracks: Ingo Wegener, Vladimiro Sassone and Bart Preneel, for having set up such an excellent scientific program. During the conference Mike Paterson has received the EATCS Distinguished Achievements Award in recognition of his outstanding scientific contributions to theoretical computer science.

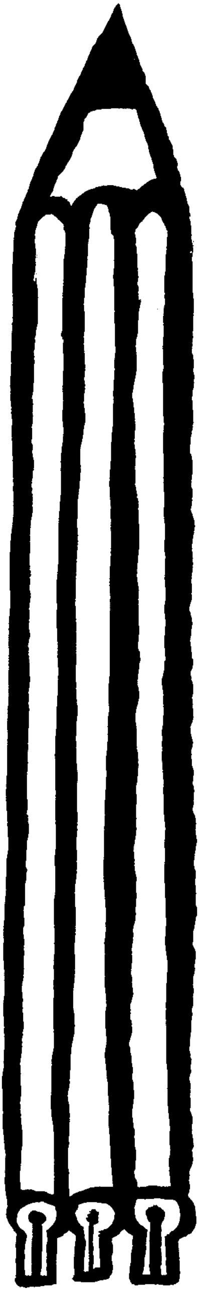


The organization of the next ICALP in Wroclaw is proceeding well. Again in 2007 ICALP will be organized in three Tracks, as in Lisbon and Venice. Besides two important conferences will be co-located with ICALP: LICS and Logic Coloquium. If you wish to contribute with the organization of Satellite Workshops you should get in touch with Tomek Jurdzinski. For more information please consult the site http://icalp07.ii.uni.wroc.pl.

Finally, let me announce you that in Venice it has also been decided that ICALP 2008 will be held in ReykjavŠk, Iceland. The organization is already making the first steps.

In conclusion let me greet the new readers of our Bulletin that, starting with this issue, is freely accessible in the net. As it is explained in the Letter from the Bulletin Editor we are happy to deliver such a qualified scientific service to the theoretical computer science community worldwide and we hope to promote, in this way, the activities of our Association further. I wish to thank Vladimiro Sassone for the extra effort that the larger visibility of the Bulletin will require.

Giorgio Ausiello, Rome
September 2006



Letter from the past President

Dear EATCS members,

As you will see reported in this issue of the Bulletin from our EATCS meetings during ICALP 2006 in Venice, some of the EATCS Council members had expressed wishes to step down from their offices, including Jan van Leeuwen as Vice-President and myself as President.

On behalf of Jan and myself I would like to thank everybody who has contributed to the development of EATCS over the past few years. It has been an exciting and challenging period, in which EATCS has continued its strategy towards playing an increasing role in a rapidly changing global research political environment.

We were happy to see two very recent steps in this direction. First of all, the overwhelming approval by all our members in the recent voting on the proposed new statutes for EATCS, aimed precisely at modernizing our organization. Secondly, the decision by the EATCS Council to experiment with open access to the Bulletin for a one year period.

We are confident that EATCS will continue to grow and to strengthen its role also in the future, in particular with Giorgio Ausiello, with his vast experience, devotion, and visions, taking over as President. EATCS couldn't have wished for a better President, and Jan and I are both looking forward to contributing also in the future, although now from different offices in the Council.

Mogens Nielsen, Aarhus
October 2006

Letter from the Bulletin Editor

Dear Reader,

Rejoice!, as the Bulletin of the EATCS is going Open Access! Yes, starting from the October 2006 issue, the Bulletin will be freely available on the EATCS web site hhtp://www.eatcs.org for a trial period of unspecified length; retrospectively, the past issues from no 81 (October 2003) will also be available electronically. EATCS members will be able to opt for a printed copy in addition to the default PDF one, by logging on to our MemberZone at www.eatcs.org.

The Council of the EATCS, recognising the high quality reached by this publication during its many years of activity, convened that the Bulletin must take up the challenge of becoming more widely available beyond the circle of EATCS members, if it is to keep improving. This is expected to enlarge our readership and, therefore, provide our authors and editors with a well-deserved, higher return for their excellent work and contribute to further raise quality standars. With its decision, the Council turns the Bulletin from 'just' a "members' benefit" to a high-visibility item, an icon to speak up for the entire Association and promote its activities. In this sense, this is a "promotion" for the BEATCS, and indeed a source of satisfaction for me. Of course, going OA is a momentous choice from the Council: the Bulletin has been among the chief Association's members' benefits for over 30 years, and before committing to it for good we need to collect feedback from our members and from the community at large, and assess the return. This is the reason to start with a trial period.

Returning to the specifics of this issue's contents, we offer the usual rich variety of contributions whose details I leave to you to discover. Touching on a sad note, unfortunately

two distinguished members of our community passed away recently, Joseph Goguen and Zdzisław Pawlak: I would like to draw your attention to the two obituaries that pay them tribute, as well as to Grzegorz Rozenberg's column, authored this time by Salomon Marcus, which focuses on work by Pawlak.

I conclude by apologising for the lack of the traditional pictures from ICALP 2006 and associated workshops: for technical reasons it has not been possible to include them; they will appear in a future issue.

Enjoy

Vladimiro Sassone, Southampton
October 2006

ICALP 2005

Report on the EATCS General Assembly 2006

The 2006 General Assembly of EATCS took place on Tuesday, July 11th, 2006, on San Servolo in Venice, the site of the ICALP. President Mogens Nielsen opened the General Assembly (GA) at 18:30. The agenda consisted of the following items.

Report of the EATCS President. Mogens Nielsen reported briefly on the EATCS activities between ICALP 2005 and ICALP 2006. He referred to the more detailed report posted a couple of weeks before the GA on the EATCS web page at `www.eatcs.org`. Mogens Nielsen explicitly mentioned and emphasized several items.

First of all, a status on the composition of the EATCS Council was given. In the 2005 election, the following ten members of the Council were elected:

Luca Aceto
Giorgio Ausiello
Giuseppe Italiano
Eugenio Moggi
Catuscia Palamidessi
Don Sanella
Jiri Sgall
Wolfgang Thomas
Ingo Wegener
Emo Welzl

Mogens Nielsen also reported that he himself, Jan van Leeuwen, and Branislav Rovan had expressed wishes to step down from their offices in the Council as President, Vice-President, and General Secretary respectively. Following this, Giorgio Ausiello had been elected as new EATCS President, Mogens Nielsen and Paul Spirakis appointed as Vice-Presidents, and Jan van Leeuwen (continuing as chairman of the Publications Committee) and Dirk Janssens (continuing as EATCS Treasurer) appointed as members of the Council. The Council had furthermore decided to propose to abandon the notion of Secretary General from the EATCS Statutes (see below).

The EATCS Council had decided to form a small number of Committees responsible for various activities, including

- EATCS Publications, chaired by Jan van-Leeuwen
- EATCS Awards and Prizes, chaired by Vladimiro Sassone
- EATCS Chapters, chaired by Eugenio Moggi
- EATCS Conferences, chaired by Giuseppe Italiano

The number of EATCS members had decreased slightly, following the increases from the past few years. Mogens Nielsen encouraged all members to update their membership information regularly (from `www.eatcs.org`).

The financial situation of EATCS showed a small surplus, mainly due to efforts of the editor of the Bulletin of the EATCS, Vladimiro Sassone, resulting in low production costs. Mogens Nielsen concluded that the financial situation of EATCS in general allows for new EATCS initiatives. Some such initiatives are currently under discussion in the EATCS Council, and he encouraged all members to contribute to this discussion by contacting Council members.

The president reported on the composition of the award committees. At the time of the General Assembly, the new members of the Gödel Prize Committee 2007 had not yet been appointed, but subsequently EATCS has appointed Colin Stirling (supplementing P. Vitanyi, and V. Diekert), and ACM-SIGACT has appointed Shafi Goldwasser (supplementing C. Papadimitriou, and J. Reif, who will be chairing the 2007 Committee). For the EATCS Award 2007 committee EATCS has appointed of Catuscia Palamidessi as a new member, supplementing , D. Peleg, and W. Thomas, who will be chairing the 2007 committee.

Mogens Nielsen also reported on a Council decision to keep also for 2007 the successful structure of ICALP with the three tracks A (*Algorithms, Automata, Complexity and Games*), B (*Logic, Semantics and Theory of Programming*), and C (*Security and Cryptography Foundations*).

In the reporting period a total of 16 events were under the auspices of EATCS, and EATCS sponsored a number of prizes for the best papers or best student papers at conferences (ICALP, ETAPS, ESA, and ICGT), Furthermore, Mogens Nielsen acknowledged the activities of the EATCS chapters. More details in the report on the web.

Mogens Nielsen also included brief reports from the EATCS associated publications, again referring to the annual report for details. In the EATCS Texts and Monographs series, a total of five Texts and one Monograph had been published in the reporting period.

Proposal of Revised EATCS Statutes. For technical reasons, the proposal for new EATCS Statutes presented at the EATCS General Assembly in 2006, had not been sent for approval by EATCS members as expected. Mogens Nielsen apologized for this, and asked the General Assembly to approve again (a slightly modified version of) the new Statues to be sent for a voting amongst all members. The purpose of the revision was still to modernize the formation of the Council (by removing references to explicit publications, and by removing the notion of a Board and the notion of a Secretary General), to clarify some ambiguities (e.g., the formulation of the nationality constraint in the formation of the Council), to remove some unfortunate restrictions (e.g., the inflexibility of timing constraint on Council elections, which fall in the holiday season), and to correct some small inconsistencies.

The proposal was approved by the GA.

Report on the Bulletin of the EATCS. The Bulletin editor, Vladimiro Sassone, gave a brief account on the Bulletin. In the reporting period, three volumes of the Bulletin of the EATCS had been published, A number of recent Bulletin issues are now available electronically for EATCS members. The editor thanked the Column editors, News editors, and everybody else contributing to the success of the Bulletin.

Importantly, the editor reported a recent decision by the Council to experiment with open access to the Bulletin for a one year period. As a consequence of this, it was furthermore decided that members of the EATCS in the future must actively ask for printed versions of the Bulletin to be posted (as opposed to now, where members can actively ask NOT to have the Bulletin sent). Members will, of course, be informed in due time about this new policy.

A special thanks and appreciation was given to the editor, V. Sassone, for his efforts in continuously improving the quality of the Bulletin.

Report ICALP 2006. Michele Bugliesi gave a report on the local arrangements for ICALP 2006, on behalf of himself and the rest of the organizing committee.

ICALP 2006 was co-located with the 8th ACM-SIGPLAN International Conference on Principles and Practice of Declarative Programming (PPDP 2006), the International Symposium on Logic-based Program Synthesis and Transformation (LOPSTR 2006), and the 19th IEEE Computer Security Foundations Workshop (CSFW 2006). On top of this, ICALP 2006 had a total of 9 pre/post-conference workshops.

The GA expressed its appreciation for a superb organization of ICALP 2006.

ICALP 2006 continued the format introduced in 2005 with three tracks with separate program committees. Besides the traditional tracks A (Algorithms, Automata, Complexity and Games) and B (Logic, Semantics and Theory of Programming), an additional track C on Security and Cryptography Foundations.

The three PC chairs Ingo Wegener (track A), Vladimiro Sassone (track B), and Bart Preneel (track C) gave separate reports. There was a very high number of 403 submissions for ICALP (230 for track A, 92 for track B, 81 for track C), out of which 109 were accepted for the conference. The three chairs provided many more statistical details of their work, some of which will appear in the usual

ICALP report contributed to this volume by Manfred Kudlek. Again, the GA expressed its appreciation for their excellent work.

The President kept the tradition presenting the ICALP organizers and the PC chairs with small gifts, thanking all of them for their efforts.

Report ICALP 2007. On behalf of the organizers, Tomasz Jurdzinski reported on the organisation of ICALP 2007 to be held in Wroclaw, Poland, on July 9–13, 2007. ICALP 2007 will follow the successful format of the three tracks A (on Algorithms, Automata, Complexity and Games, chaired by Lars Arge, University of Aarhus, Denmark), B (on Logic, Semantics and Theory of Programming, chaired by Andrzej Tarlecki, University of Warsaw, Poland), and C (on Security and Cryptography Foundations, chaired by Christian Cachin, IBM Zurich Research Laboratory, Switzerland).

The conference will co-locate in 2007 with the 22nd Annual IEEE Symposium on Logic in Computer Science (LICS 2007) and the ASL European Summer Meeting (Logic Colloquium '07).

The GA thanked Jurdzinski and the whole group from Wroclaw for their organizational efforts.

Venue for ICALP 2008. Mogens Nielsen announced that he was only aware of one contender for hosting ICALP in 2008, the Icelandic Center of Excellence in Theoretical Computer Science, ICE-TCS, in Reykjavik, Iceland. When nobody from those present brought up another proposal, Magnus Halldorsson presented (on behalf of himself and his co-organizers Anna Ingolfsdottir and Luca Aceto) the proposal of organizing ICALP 2008, including basic information about the ICE-TCS, the Universities in Reykjavik, the city of Reykjavik, accommodation facilities, etc.

The GA approved unanimously Reykjavik as the site for ICALP 2007.

EU matters. The EATCS Vice-President Paul Spirakis gave a brief account of recent developments concerning on the Seventh Framework Programme (2007-2013) entitled: News From Brussels and Some Thoughts for the Future. Paul Spirakis focused on issues like new funding schemes and new procedures, and emphasized particularly the role of basic science. The presentation was very well received by the GA, indicated by a subsequent lively discussion.

Specials. At this point, around 20:00, the President thanked all present and concluded the 2006 General Assembly of the EATCS by introducing Manfred Kudlek, presenting the statistics of the authors who published repeatedly at ICALP, and presenting the special EATCS badges to those having reached 5 or more full papers at ICALP. By tradition Manfred also presented the EATCS badges to the editors of the ICALP 2006 proceedings.

Giorgio Ausiello and Mogens Nielsen

EATCS AWARD 2007

Call for Nominations

EATCS annually honors a respected scientist from our community with the prestigious EATCS Distinguished Achievements Award. The award is given to acknowledge extensive and widely recognised contributions to theoretical computer science over a life long scientific career.

For the EATCS Award 2007, candidates may be nominated to the Awards Committee. Nominations must include supporting justification and will be kept strictly confidential. The deadline for nominations is: **December 15, 2006**.

Nominations and supporting data should be sent to the chairman of the EATCS Awards Committee:

Prof. Dr. Wolfgang Thomas
Lehrstuhl Informatik 7
RWTH Aachen
Ahornstr. 55, 52074 Aachen (Germany)

Email: `thomas@informatik.rwth-aachen.de`

Previous recipients of the EATCS Award are

R.M. Karp	(2000)	C. Böhm	(2001)
M. Nivat	(2002)	G. Rozenberg	(2003)
A. Salomaa	(2004)	R. Milner	(2005)
M. Paterson	(2006)		

The next award is to be presented during ICALP'2007 in Wroclaw.

Institutional Sponsors

BRICS, Basic Research in Computer Science,
Aarhus, Denmark

Elsevier Science
Amsterdam, The Netherlands

IPA, Institute for Programming Research and Algorithms,
Eindhoven, The Netherlands

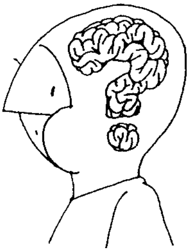
Microsoft Research,
Cambridge, United Kingdom

PWS, Publishing Company,
Boston, USA

TUCS, Turku Center for Computer Science,
Turku, Finland

UNU/IIST, UN University, Int. Inst. for Software Technology,
Macau, China

EATCS News



Report from the Japanese Chapter

K. Makino (Tokyo Univ.)

EATCS-JP/LA Workshop on TCS

The *sixth EATCS/LA Workshop on Theoretical Computer Science* will be held at Research Institute of Mathematical Sciences, Kyoto Univ., January 29 ~ 31, 2007. The workshop will be jointly organized with *LA*, Japanese association of theoretical computer scientists. Its purpose is to give a place for discussing topics on all aspects of theoretical computer science.

A formal call for papers will be announced at our web page early November, and a program will be announce early January, where we are also planning to announce a program in the next issue of the Bulletin. Please check our web page around from time to time. If you happen to stay in Japan around that period, it is worth attending. No registration is necessary for just listening to the talks; you can freely come into the conference room. (Contact us by the end of November if you are considering to present a paper.) Please visit Kyoto in its most beautiful time of the year !

5th EATCS-JP/LA Presentation Award

The fifth EATCS/LA Workshop on Theoretical Computer Science was held at Research Institute of Mathematical Sciences, Kyoto Univ., January 30th ~ February 1st, 2006. **Mr. Ryotaro Hayashi** (Tokyo Inst. of Tech.) who presented the following paper, was selected as the 4th EATCS/LA Presentation Award.

Anonymizable public-key encryption
by R. Hayashi, K. Tanaka (Tokyo Inst. of Tech.)

The award was given to him at the Summer LA Symposium held in August 2006. *Congratulations!* Please check our web page for the detail information and the list of presented papers.

On TCS Related Activities in Japan:

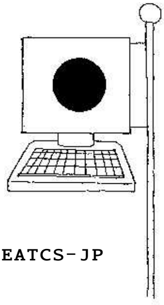
TGCOMP Meetings, January ~ June, 2006

The *IEICE*, Institute for Electronics, Information and Communication Engineers of Japan, has a technical committee called *TGCOMP*, Technical Group on foundation of COMPuting. During January ~ June of 2006, *TGCOMP* organized 4 meetings and about 37 papers (including one tutorial) were presented there. Topics presented are, very roughly, classified as follows.

Algorithm: On Graphs (11)
Algorithm: On Strings (3)
Algorithm: On Other Objects (5)
Combinatorics / Probabilistic Analysis (3)
Computational Complexity (5)
Cryptography (2)
Distributed Computing (2)
Formal Languages and Automata (2)
Quantum Computing (2)
DNA Computing (2)

See our web page for the list of presented papers (title, authors, key words, email).

The Japanese Chapter



Chair: Kazuo Iwama
V.Chair: Osamu Watanabe
Secretary: Kazuhisa Makino
email: eatcs-jp@is.titech.ac.jp
URL: http://www.is.titech.ac.jp/~watanabe/eatcs-jp

News from India

by

Madhavan Mukund
Chennai Mathematical Institute
Chennai, India
madhavan@cmi.ac.in

We begin with a quick summary of some recent events.

Summer School on Algorithms, Complexity and Cryptology A summer school on *Algorithms, Complexity and Cryptology* was organized in Bangalore from May 22 to June 9, 2006 by Microsoft Research India and the IISc Mathematics Initiative, Indian Institute of Science, Bangalore. The list of speakers included Dan Boneh (Stanford, USA), Kamal Jain (Microsoft Research, USA), David Jao (Microsoft Research, USA), Ravi Kannan (Yale University, USA), Kivanc Mihcak (Microsoft Research, USA), A. Shamir (Weizmann Institute, Israel), and Eran Tromer (Weizmann Institute, Israel). The school was attended by senior undergraduate students, graduate students, research scholars and faculty members and was well received.

Formal Methods Update Meeting During the past few years, the Indian Association for Research in Computing Science (IARCS) has organized regular "update" meetings in the area of formal methods. The meetings are intended as a forum for Indian researchers and students in theoretical computer science to update themselves on current trends and to explore new research areas.

This year's meeting was held at IIT Guwahati from 3–6, July 2006. Bharat Adsul and Madhavan Mukund from Chennai Mathematical Institute surveyed various issues related to parity games. K Narayan Kumar gave an introduction to the expressive completeness of LTL with respect to first-order logic. Kamal Lodaya, Antoine Meyer and R Ramanujam from the Institute of Mathematical Sciences gave a series of talks on infinite-state verification. Paritosh Pandya from the Tata Institute of Fundamental Research spoke on timed logics while Anil Seth from

IIT Kanpur discussed quantitative games. In addition to these survey talks, some participants presented technical talks on their work.

For many participants, this was the first opportunity to visit the new IIT Guwahati campus, on the banks of the Brahmaputra. The organization by Purandar Bhaduri's team was impeccable and the workshop went off very well, both academically and socially.

We now move onto some forthcoming events.

SEFM 2006 SEFM 2006, the 4th IEEE International Conference on Software Engineering and Formal Methods, is being held in Pune, India during the period September 11–15, 2006 even as this article is being written. The aim of the conference is to bring together practitioners and researchers from academia, industry and government to advance the state-of-the-art in formal methods, to scale up their application in software industry and to encourage their integration with practical engineering methods.

The Program Committee for SEFM 2006 is jointly chaired by Paritosh Pandya (TIFR, Mumbai, India) and Dang Van Hung (UNU-IIST, Macao, China). This year's invited speakers are Sriram Rajamani (Microsoft Research India, India), John Rushby (SRI International, USA), Joseph Sifakis (CNRS and VERIMAG, France) and Bertrand Meyer (ETH Zurich, Switzerland).

The website for SEFM 2006 is at `http://www.iist.unu.edu/SEFM06`.

FSTTCS 2006 The 26th edition of FSTTCS will take place from December 13–15, 2006 at the Indian Statistical Institute, Kolkata. Anupam Gupta and Amit Kumar will organize a satellite workshop on Approximation Algorithms on December 16. Another satellite workshop is being planned for December 12. Details will be announced shortly.

The Program Committee is co-chaired by S. Arun-Kumar and Naveen Garg from IIT, Delhi. The list of invited speakers for FSTTCS 2006 includes Gordon Plotkin (Edinburgh, UK), Emo Welzl (ETH Zurich, Switzerland), Gérard Boudol (INRIA, Sophia Antipolis, France), David Shmoys (Cornell, USA), and Eugene Asarin (LIAFA, Paris 7, France).

A total of 34 papers have been accepted from over 150 submissions. The list of accepted papers can be found via the conference website at `http://www.fsttcs.org`.

We look forward to seeing a lot of you at FSTTCS, the main conference of the Indian Association for Research in Computing Science (IARCS).

ISAAC 2006 The 17th International Symposium on Algorithms and Computation (ISAAC 2006) will take place in Kolkata, India. The Program Committee is chaired by Tetsuo Asano (JAIST, Japan). The invited speakers at ISAAC 2006 are Tamal Dey, (Ohio State, USA) and Kazuo Iwama (Kyoto, Japan). The list of accepted papers is available via the conference website, `http://www.isical.ac.in/~isaac06`.

International Conference on Discrete Mathematics ICDM 2006 will be held in Bangalore from December 15 to December 18, 2006. The conference is organized jointly by the Ramanujan Mathematical Society and Indian Institute of Science, Bangalore. The academic programme consists of plenary talks, invited talks, poster paper presentations and mini-symposia on Discrete Mathematics and its applications. For more details, look up the conference webpage at `http://www.ramanujanmathsociety.org/icdm2006.html`.

Workshop on Algorithms for Data Streams A workshop on Algorithms for Data Streams will be held at the Department of Computer Science and Engineering IIT Kanpur from December 18–20, 2006.

The aim of this workshop is to bring together active and world-class researchers to discuss cutting-edge research and ideas in the areas of data stream algorithms, techniques and complexity of data streaming problems. The workshop is being organized by Sumit Ganguly (IIT Kanpur), Sudipto Guha (University of Pennsylvania) and S. Muthukrishnan (Google). The list of confirmed speakers is long and studded with illustrious names. Participation is by invitation only.

For more details, see the workshop page at `http://www.cse.iitk.ac.in/users/sganguly/workshop.html`.

Madhavan Mukund, Chennai Mathematical Institute
Secretary, IARCS (Indian Association for Research in Computing Science)
`http://www.cmi.ac.in/~madhavan`

News from Ireland

BY

Anthony K. Seda

Department of Mathematics, National University of Ireland
Cork, Ireland
a.seda@ucc.ie

The conference Information-MFCSIT'06 took place on the campus of NUI, Cork from 1st August to 5th August, 2006. It was a joint meeting in which the Fourth International Conference on Information (Information'06) and the Fourth Irish Conference on the Mathematical Foundations of Computer Science and Information Technology (MFCSIT'06) were co-located, and hosted by the International Information Institute, Tokyo, and NUI, Cork.

The meeting was well-attended with about 110 participants from various parts of the world, including nearly forty from China, Japan, Korea, and Vietnam, as well as many from Ireland, UK and mainland Europe and some from the USA. We were again fortunate in having nine well-known keynote speakers who delivered excellent and stimulating talks, as follows. Eugene Freuder (NUI, Cork, Ireland): "Constraint Programming Software Can Help You Make Decisions"; Grant Malcolm (University of Liverpool, UK): "Sheaves, Objects, and Distributed Systems"; Michael Mitzenmacher (Harvard University, USA): "Network Applications of Bloom Filters and Related Data Structures"; Tadao Nakamura (Tohoku University, Japan): "Trends in High Performance Computing with Low Power"; John Power (Laboratory for Foundations of Computer Science, Edinburgh, UK): "The Category-Theoretic Analysis of Universal Algebra: Lawvere Theories and Monads"; Peter Puschner (Technische Universitaet Wien, Austria): "Architecture Support for Temporal Predictability and Composability in Real-Time Computing"; Fuji Ren (University of Tokushima, Japan): "Affective Information Processing and Recognizing Human Emotion"; Herbert Wiklicky (Imperial College, London, UK): "Approximation in Program Analysis: The Importance of Being

Close"; Jungong Xue (Fudan University, Shanghai, China): "Geometric Tail for Non-Pre-emptive Priority MAP/PH/1 Queues".

In addition to the keynote lectures and regular contributed papers, a number of special sessions were arranged. Two were held in Information'06: "Cyber-Terrorism and the Information Sword" (organized by Mahmoud Eid, University of Ottawa, Canada); and "The Intellectual Human Support Technologies and its Application" (organized by Tetsuya Tanioka, University of Tokushima, Japan and Rozzano C. Locsin, Florida Atlantic University, USA). Seven special sessions were held in MFCSIT: "Formal Approaches to Security" (organized by Alessandra Di Pierro, University of Pisa, Italy, Michael Huth and Herbert Wiklicky both of Imperial College, London, UK); "Complex Networks and Stochastic Dynamics" (organized by James Gleeson, NUI, Cork); "Logic Semantics in Computer Science" (organized by Vladimir Komendantsky, NUI, Cork, Ireland); "Category Theory in Computer Science" (organized by John Power, University of Edinburgh, UK); "Coding Theory and Cryptography" (organized by Max Sala, NUI, Cork, Ireland); "Modular Analysis of Software: Theory and Applications" (organized by Michel Schellekens, NUI, Cork, Ireland); and "Machine Models and Computation" (organized by Damien Woods, NUI, Cork, Ireland).

It is a pleasure to thank the sponsors of the meeting, and they included the Boole Centre for Research in Informatics, NUI, Cork; Science Foundation Ireland; The Chinese Academy of Science and Engineering in Japan; The College of Science, Engineering and Food Science, NUI, Cork; The Department of Computer Science, NUI, Cork; The Department of Mathematics, NUI, Galway; The School of Mathematics, Applied Mathematics and Statistics, NUI, Cork; and Ballygowan Pure Irish Water. The meeting was co-chaired by Lei Li, Fuji Ren, T. Hurley and A.K. Seda.

As usual, the Proceedings of Information'06 will be published in Information: An International Journal, and the Proceedings of MFCSIT'06 will be published in Elsevier's Electronic Notes in Theoretical Computer Science (ENTCS).

NEWS FROM LATIN AMERICA

BY

ALFREDO VIOLA

Instituto de Computación, Facultad de Ingenierìa
Universidad de la República
Casilla de Correo 16120, Distrito 6, Montevideo, Uruguay
viola@fing.edu.uy

In this issue I present the Workshop on Foundations of Databases and the Web to honor the memory of Alberto Mendelzon, the Operations Research Latin-American Congress, the Second Latin-American Workshop on Cliques in Graphs, and the Workshop on Graph Theory and Applications. At the end I present a list of the main events in Theoretical Computer Science to be held in Latin America in the following months.

Workshop on Foundations of Databases and the Web.

The workshop on Foundations of Databases and the Web is organized to honor the memory of our dear friend and colleague Alberto Mendelzon, who contributed so much to the Database community as well as to South American Computer Science.

Alberto Oscar Mendelzon was one of the pioneers who helped to lay the foundations of relational databases. His early work on database dependencies has been influential in both the theory and practice of data management. He was a professor of computer science at the University of Toronto, was born in Buenos Aires, Argentina. His academic journey began in Argentina and he maintained, throughout his life, close ties to his home country and home continent. He graduated from the University of Buenos Aires in 1973 before studying at Princeton as a Fulbright

Scholar. At Princeton, he received a M.S.E. degree in 1977, a M.A. degree in 1978, and a Ph.D. degree in 1979. He was a post-doctoral fellow at IBM's T.J. Watson Research Center for a year before joining the University of Toronto in 1980.

Alberto was a quiet man who did not seek out honors. He was modest about his role in shaping the foundations of relational databases and his pioneering work in laying the foundations for querying the web. He was elected to the Royal Society of Canada (the Canadian National Academy for Science, Engineering, and the Humanities) which is Canada's top academic accolade.

You can visit Alberto Mendelzon's homepage at the University of Toronto. Also you can read the tribute article from SIGMOD, and this memorial.

The workshop will be held November 6-10 2006 in Chile aboard a ship touring the San Rafael Glacier, one of the most impressive and scenic tourist attractions in the world.

Attendance to the workshop is by invitation only. The workshop will provide a venue for Alberto's colleagues and their collaborators to present and discuss research challenges in foundations of the web and databases.

For more information you may visit `http://grupoweb.upf.es/webdb/`.

XIII CLAIO - Operations Research Latin-American Congress.

The XIII CLAIO, the Operations Research Latin-American Congress, and The 1st ALIO/INFORMS Workshop on OR Education will take place on November 27 to 30, 2006, in Montevideo, the capital of the Republic of Uruguay. The Congress is chaired by Dr. Héctor Cancela and organized by the Operations Research Department of the Computer Science Institute of the University of the Universidad de la República, Uruguay (UDELAR), ALIO (Latin American Operation Research Associations) and IFORS (International Federation of Operations Research Societies). The workshop is jointly organized by ALIO and INFORMS, under the auspices of IFORS, and hosted by the Operations Research Department of UDELAR

The plenary speakers are Martin Gr otschel (IFORS Distinguished Lecturer), James J. Cochran, Carlos A. Coello Coello, Monique Guignard, Michel Gendreau, Pierre L'Ecuyer, Gerardo Rubino and Julián Araoz.

In (`http://www.fing.edu.uy/inco/eventos/claio06`) you will find more information of this event.

Second Latin-American Workshop on Cliques in Graphs.

The Second Latin-American Workshop on Cliques in Graphs will be held in the Facultad de Ciencias Exactas of the Universidad Nacional de La Plata, Argentina, on October 18-20, 2006. The aim of the Workshop is to promote a meeting of researchers in Graph Theory, Algorithms and Combinatorics, particularly those working in Graph Operators, Intersection Graphs and Perfect Graphs.

During the meeting 20 scientific communications will be exposed and there will be 5 plenary conferences: Andreas Brandst adt (Germany), Michel Habib (France), Pavol Hell (Canada), Francisco Larrión with Miguel Angel Pizaña (México) in honor to Victor Neumann-Lara, and Jorge Urrutia (México).

Selected full papers will be published in a special issue of Revista de la Unión Matemática Argentina. Papers published in Revista de la Unión Matemática Argentina are reviewed in Mathematical Reviews and Zentralblatt f ur Mathematik.

The Organizing Committee is co-chaired by Liliana Alcón and Marisa Gutierrez (Argentina), and has the participation of Márcia Rosana Cerioli (Brazil), Min Chih Lin (Argentina), Guillermo Durán (Argentina), Celina Miraglia Herrera de Figueiredo (Brazil), Miguel Angel Pizaña (México), Fábio Protti (Brazil) and Jayme Luiz Szwarcfiter (Brazil)

In `http://www.mate.unlp.edu.ar/~liliana/cw06.html` you will find more information of this event.

Workshop on Graph Theory and Applications.

The Workshop on Graph Theory and Applications will be held in Porto Alegre, Brazil on November 20 - 21, 2006 and is chaired by Vilmar Trevisan. This Workshop will be an international forum for researchers to disseminate ideas, propose techniques, present and discuss approaches to open problems, share experiences and discuss applications of Graph Theory. The target audience are graduate students, researchers and professionals working in mathematics and computer science, interested in graphs and their applications.

The Workshop will consist of mini-courses, invited lectures and session of open talks.

The mini-courses and invited lectures are going to be announced as soon as the final program is ready. The following researchers have agreed to give lectures: Celina M. H. de Figueiredo, Stephen T. Hedetniemi (to be confirmed), David P. Jacobs, Robert E. Jamison (to be confirmed), Luis Gustavo Nonato, and Jayme Szwarcfiter.

The Proceedings of the Workshop will be published in a CD (with ISBN). The Proceedings will be available at the time of the conference. The organizers are negotiating a special issue in an international journal for the extended versions of selected papers presented at the workshop.

In `http://euler.mat.ufrgs.br/workgraph/home.html` you will find more information of this event.

Regional Events

- September 17 - 23, 2006, Natal, RN, Brazil: SMBF 2006 - Brazilian Symposium on Formal Methods (http://www.dimap.ufrn.br/sbmf2006).

- September 17 - 23, 2006, Natal, RN, Brazil: ICGT 2006 - International Conference on Graph Transformation

  (http://www.dimap.ufrn.br/icgt2006).

- September 18 - 22, 2006, San Luis Potosí, México: ENC 2006 - Encuentro Internacional de Ciencias de la Computación (http://enc.smcc.org.mx).

- October 18 - 20, 2006, La Plata, Argentina: Second Latin-American Workshop on Cliques in Graphs

  (http://www.mate.unlp.edu.ar/~liliana/cw06.html).

- October 25 - 27, 2006, Puebla, México: LA WEB 06 - 4th Latin American Web Congress (http://ict.udlap.mx/laweb2006/).

- November 6 - 10, 2006, San Rafael Glacier, Chile: Workshop on Foundations of Databases and the Web (http://grupoweb.upf.es/webdb/).

- November 20 - 21, 2006, Porto Alegre, Brazil: Workshop on Graph Theory and Applications

  (http://euler.mat.ufrgs.br/workgraph/home.html).

- November 27 - 30, 2006, Montevideo, Uruguay: XIII CLAIO - Congreso Latino-Iberoamericano de Investigación Operativa

  (http://www.fing.edu.uy/inco/eventos/claio2006).

- January 10 - 13, 2007, Buenos Aires, Argentina: Conference on Logic Computability and Randomness 2007

  (http://www.dc.uba.ar/people/logic2007).

News from New Zealand

by

C.S. Calude

Department of Computer Science, University of Auckland
Auckland, New Zealand
cristian@cs.auckland.ac.nz

1 Scientific and Community News

0. The number of CS+IT students has decreased sharply in many parts of the worlds, including Australia and NZ, and the impact for academia and research in the field was dramatic. According to *Computerworld* (July 17, 2006) the future seems brighter: “According to the Bureau of Labor Statistics, one out of every four new jobs between now and 2012 will be IT-related," (Mark Hanny, vice president of IBM’s Academic Initiative outreach program) and “We’re seeing a lack of talented IT professionals looking for new positions," (Greg Fittinghoff, vice president of business systems development at Time Inc. in New York). To turn this trend around, several initiatives are under way; perhaps theoretical computer science should be also actively involved in this process.

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

280. L. Staiger. On Maximal Prefix Codes, 05/2006.
281. G. J. Chaitin. Is Incompleteness A Serious Problem, 07/2006.
282. G. J. Chaitin. Speculations on Biology, Information and Complexity, 07/2006.

2 A Dialogue on Mathematics & Physics with Gregory Chaitin

As a visiting professor in the Department of Computer Science of the University of Auckland, Greg Chaitin is a frequent visitor in New Zealand. During his recent visit in July 2006 we had time for a dialogue about mathematics, physics, and philosophy—C.S.C.

Cristian Calude: I suggest we discuss the question, *Is mathematics independent of physics*?

Gregory Chaitin: Okay.

CC: Let's recall David Deutsch's 1982 statement:

> *The reason why we find it possible to construct, say, electronic calculators, and indeed why we can perform mental arithmetic, cannot be found in mathematics or logic.* **The reason is that the laws of physics "happen" to permit the existence of physical models for the operations of arithmetic** *such as addition, subtraction and multiplication.*

Does this apply to mathematics too?

GC: Yeah sure, and if there is real randomness in the world then Monte Carlo algorithms can work, otherwise we are fooling ourselves.

CC: So, if experimental mathematics is accepted as "mathematics," it seems that we have to agree that mathematics depends "to some extent" on the laws of physics.

GC: You mean math conjectures based on extensive computations, which of course depend on the laws of physics since computers are physical devices?

CC: Indeed. The typical example is the four-color theorem, but there are many other examples. The problem is more complicated when the verification is not done by a conventional computer, but, say, a quantum automaton. In the classical scenario the computation is huge, but in principle it can be verified by an army of mathematicians working for a long time. In principle, theoretically, it is feasible to check every small detail of the computation. In the quantum scenario this possibility is gone.

GC: Unless the human mind is itself a quantum computer with quantum parallelism. In that case an exponentially long quantum proof could not be written out, since that would require an exponential amount of "classical" paper, but a quantum mind could directly perceive the proof, as David Deutsch points out in one of his papers.

CC: Doesn't Roger Penrose claim that the mind is actually a quantum computer?

GC: Yes, he thinks quantum gravity is involved, but there are many other possible ways to get entanglement.

CC: How can such a parallel quantum proof be communicated and checked when it exists only in the mind of the mathematician who "saw" it?

GC: Well, I guess it's like the design of a quantum computer. You tell someone the parallel quantum computation to perform to check all the cases of something, and if they have a quantum mind maybe they can just do it. So you could publish the quantum algorithm as a proof, which the readers would do in their heads to verify your claim.

CC: On paper you have only the quantum algorithm; everything else is in the mind! What about disagreements, how can one settle them "keeping in mind" (no pun!) that quantum algorithms are probabilistic? Aren't we in danger of loosing an essential feature of mathematics, the independent checkability of proofs in finite time?

GC: Well, even now you don't publish **all** the steps in a proof, you depend on people to do some of it in their heads. And if one of us has a quantum mind, then probably everyone does, or else that would become a prerequisite, like a high IQ, for doing mathematics!

CC: Theoretical physics suggests that in certain relativistic space-times, the so-called Malament-Hogarth space-times, it may be possible for a computer to receive the answer to a yes/no question from an *infinite computation* in a *finite time*. This may lead to a kind of "realistic scenario" for super-Turing computability.

GC: Well, to get a big speed-up you can just take advantage of relativistic time dilation due either to a very strong gravitational field near the event horizon of a black hole or due to very high-speed travel (near the speed of light). You assign a task to a normal computer, then you slow down your clock so that you can wait for the result of an extremely lengthy computation. To you, it seems like just a short wait, to the computer, aeons have passed...

CC: Physicist Seth Lloyd[1] has found that the "ultimate laptop," a computer with a mass of one kilogram confined to a volume of one litre, operating at the

[1] S. Lloyd, "Ultimate physical limits to computation," *Nature* (2000) **406**, 1047–1054.

fundamental limits of speed and memory capacity determined by the physics of our universe, can perform 10^{51} operations per second on 10^{31} bits. This device sort of looks like a black hole.

GC: And he's just published a book called *Programming the Universe.* The basic idea is that the universe is a computation, it's constantly computing its own time evolution.

CC: What about the Platonic universe of mathematical ideas? Is that "muddied" by physics too? To exist mathematics has to be communicated, eventually in some written form. This depends upon the physical universe!

GC: Yes, proofs have to be written on paper, which is physical. Proofs that are too long to be written down may exist in principle, but they are impossible to read.

CC: Talking about writing things down, logicians have studied logics with infinitely long formulas, with infinite sets of axioms, and with infinitely long proofs.

GC: How infinite? $\aleph_0$, $\aleph_1$, $\aleph_2$?

CC: Could it be that such eccentric proofs correspond to something "real"?

GC: Well, if people had $\aleph_2$ minds, then formulas $\aleph_0$ characters long would be easy to deal with! There's even a set-theoretical science fiction novel by Rudy Rucker called *White Light* in which he tries to describe what this might feel like. I personally like a world which is discrete and $\aleph_0$ infinite, but why should Nature care what I think?

In one of his wilder papers, physicist Max Tegmark suggests that any conceptually possible world, in other words, one that isn't self-contradictory, actually exists. Instead of conventional Feynman path integrals summing over all histories, he suggests some kind of crazy new integral over all possible universes! His reasoning is that the ensemble of all possible universes is **simpler** than having to pick out individual universes!

Leibniz had asked why is there something rather than nothing, because nothing is simpler than something, but as Tegmark points out, so is **everything**. In his approach you don't have to specify the individual laws for this particular universe, it's just one of many possibilities.

CC: What about constructive mathematics?

GC: Of course the mathematical notion of computability depends upon the physical universe you are in. We can imagine worlds in which oracles for the halting problem exist, or worlds in which Hermann Weyl's one second, half second, quarter second, approach actually enables you to calculate an infinite number of steps in exactly two seconds. But I guess computability can handle this, everything relativises, you just add an appropriate oracle. All the proofs go through as before.

CC: —Are you talking about a physical Church-Turing Thesis?

GC: Yes I am.—But I think the notion of a universal Turing machine changes in a more fundamental way if Nature permits us to toss a coin, if there really are independent random events. (Quantum mechanics supplies such events, but you can postulate them separately, without having to buy the entire QM package.) If Nature really lets us toss a coin, then, with extremely high probability, you can actually compute algorithmically irreducible strings of bits, but there's no way to do that in a deterministic world.

CC: Didn't you say that in your 1966 *Journal of the ACM* paper?

GC: Well yes, but the referee asked me to remove it, so I did. Anyway, that was a long time ago.

CC: A spin-off company from the University of Geneva, *id Quantique*, markets a quantum mechanical random number generator called *Quantis*. *Quantis* is available as an OEM component which can be mounted on a plastic circuit board or as a PCI card; it can supply a (theoretically, arbitrarily) long string of quantum random bits sufficiently fast for cryptographic applications. A universal Turing machine working with *Quantis* as an oracle seems different from a normal Turing machine. Are Monte Carlo simulations powered with quantum random bits more accurate than those using pseudo-randomness?

GC: Well yes, because you can be unlucky with a pseudo-random number generator, but never with real random numbers. People have gotten anomalous results from Monte Carlo simulations because the pseudo-random numbers they used were actually in sync with what they were simulating.

Also real randomness enables you, with probability one, to produce an algorithmically irreducible infinite stream of bits. But any infinite stream of pseudo-random bits is extremely redundant and highly compressible, since it's just the output of a finite algorithm.

CC: In a universe in which the halting problem is solvable many important current open problems will be instantly solved: the Riemann hypothesis or the Goldbach Conjecture.

GC: Yes, and you could also look through the tree of all possible proofs in any formal axiomatic theory and see whether something is a theorem or not, which would be mighty handy.

CC: Talking about the Riemann hypothesis, which is about primes, there's the surprising connection with physics noticed by Freeman Dyson that the distribution of the zeros of the Riemann function looks a lot like the Wigner distribution for energy levels in a nucleus.[2]

And in an inspiring paper on "Missed opportunities" written by Dyson in 1972, he observes that relativity could have been discovered 40 years before Einstein if mathematicians and physicists in Göttingen had spoken to each other.

GC: Well in fact, relativity **was** discovered before Einstein by Poincaré—that's why the transformation group for Maxwell's equations is called the Poincaré group—however Einstein's version was easier for most people to understand.

But mathematicians shouldn't think they can replace physicists: There's a beautiful little 1943 book on *Experiment and Theory in Physics* by Max Born where he decries the view that mathematics can enable us to discover how the world works by pure thought, without substantial input from experiment.

CC: What about set theory? Does this have anything to do with physics?

GC: I think so. I think it's reasonable to demand that set theory has to apply to **our** universe. In my opinion it's a fantasy to talk about infinities or Cantorian cardinals that are larger than what you have in your physical universe. And what's **our** universe actually like?

- a finite universe?
- discrete but infinite universe ($\aleph_0$)?
- universe with continuity and real numbers ($\aleph_1$)?
- universe with higher-order cardinals ($\geq \aleph_2$)?

Does it really make sense to postulate higher-order infinities than you have in your physical universe? Does it make sense to believe in real numbers if our world is actually discrete? Does it make sense to believe in the set $\{0, 1, 2, \ldots\}$ of **all** natural numbers if our world is really finite?

CC: Of course, we may never know if our universe is finite or not. And we may never know if at the bottom level the physical universe is discrete or continuous...

GC: Amazingly enough, Cris, there is some evidence that the world may be discrete, and even, in a way, two-dimensional. There's something called the holographic principle, and something else called the Bekenstein bound. These ideas come from trying to understand black holes using thermodynamics. The tentative

[2] Andrew Odlyzko and Michael Berry continued this work. And recently Jon Keating and Nina Snaith, two mathematical physicists, have been able to prove something new about the moments of the Riemann zeta function this way.

conclusion is that any physical system only contains a finite number of bits of information, which in fact grows as the surface area of the physical system, not as the volume of the system as you might expect, whence the term "holographic."

CC: That's in Lee Smolin's book *Three Roads to Quantum Gravity,* right?

GC: Yes. Then there are physical limitations on the human brain. Human beings and computers feel comfortable with different styles of proofs. The human push-down stack is short. Short-term memory is small. But a computer has a big push-down stack, and its short-term memory is large and extremely accurate. Computers don't mind lots of computation, but human beings prefer ideas, or visual diagrams. Computer proofs have a very different style from human proofs. As Turing said, poetry written by computers would probably be of more interest to other computers than to humans!

CC: In a deterministic universe there is no such thing as real randomness. Will that make Monte Carlo simulations fail?

GC: Well, maybe. But one of the interesting ideas in Stephen Wolfram's *A New Kind of Science* is that all the randomness in the physical universe might actually just be pseudo-randomness, and we might not see much of a difference. I think he has deterministic versions of Boltzmann gas theory and fluid turbulence that work even though the models in his book are all deterministic.

CC: What about the axioms of set theory, shouldn't we request arguments for their validity? An extreme, but not unrealistic view discussed by physicist Karl Svozil, is that the only "reasonable" mathematical universe is the physical universe we are living in (or where mathematics is done). Pythagoreans might have subscribed to this belief.

Should we still work with an axiom—say the axiom of choice—if there is evidence against it (or there is not enough evidence favouring it) in this specific universe? In a universe in which the axiom of choice is not true one cannot prove the existence of Lebesgue non-measurable sets of reals (Robert Solovay's theorem).

GC: Yes, I argued in favor of that a while back, but now let me play Devil's advocate. After all, the real world is messy and hard to understand. Math is a kind of fantasy, an ideal world, but maybe in order to be able to prove theorems you have to simplify things, you have to work with a toy model, not with something that's absolutely right. Remember you can only solve the Schrödinger equation exactly for the hydrogen atom! For bigger atoms you have to work with numerical approximations and do lots and lots of calculations. . .

CC: Maybe in the future mathematicians will work closely with computers.

Maybe in the future there will be hybrid mathematicians, maybe we will have a man/machine symbiosis. This is already happening in chess, where Grandmasters use chess programs as sparing partners and to do research on new openings.

GC: Yeah, I think you're right about the future. The machine's contribution will be speed, highly accurate memory, and performing large routine computations without error. The human contribution will be new ideas, new points of view, intuition.

CC: But most mathematicians are not satisfied with the machine proof of the four-color conjecture. Remember, for us humans, *Proof = Understanding.*

GC: Yes, but in order to be able to amplify human intelligence and prove more complicated theorems than we can now, we may be forced to accept incomprehensible or only partially comprehensible proofs. We may be forced to accept the help of machines for mental as well as physical tasks.

CC: We seem to have concluded that mathematics depends on physics, haven't we? But mathematics is the main tool to understand physics. Don't we have some kind of circularity?

GC: Yeah, that sounds very bad! But if math is actually, as Imre Lakatos termed it, quasi-empirical, then that's exactly what you'd expect. And as you know Cris, for years I've been arguing that information-theoretic incompleteness results inevitably push us in the direction of a quasi-empirical view of math, one in which math and physics are different, but maybe not as different as most people think. As Vladimir Arnold provocatively puts it, math and physics are the same, except that in math the experiments are a lot cheaper!

CC: In a sense the relationship between mathematics and physics looks similar to the relationship between meta-mathematics and mathematics. The incompleteness theorem puts a limit on what we can do in axiomatic mathematics, but its proof is built using a substantial amount of mathematics!

GC: What do you mean, Cris?

CC: Because mathematics is incomplete, but incompleteness is proved within mathematics, meta-mathematics is itself incomplete, so we have a kind of unending uncertainty in mathematics. This seems to be replicated in physics as well: Our understanding of physics comes through mathematics, but mathematics is as certain (or uncertain) as physics, because it depends on the physical laws of the universe where mathematics is done, so again we seem to have unending uncertainty. Furthermore, because physics is uncertain, you can derive a new form of uncertainty principle for mathematics itself...

GC: Well, I don't believe in absolute truth, in total certainty. Maybe it exists in the Platonic world of ideas, or in the mind of God—I guess that's why I became a mathematician—but I don't think it exists down here on Earth where we are. Ultimately, I think that that's what incompleteness forces us to do, to accept a spectrum, a continuum, of possible truth values, not just black and white absolute truth.

In other words, I think incompleteness means that we have to also accept heuristic proofs, the kinds of proofs that George Pólya liked, arguments that are rather convincing even if they are not totally rigorous, the kinds of proofs that physicists like. Jonathan Borwein and David Bailey talk a lot about the advantages of that kind of approach in their two-volume work on experimental mathematics. Sometimes the evidence is pretty convincing even if it's not a conventional proof. For example, if two real numbers calculated for thousands of digits look exactly alike. . .

CC: It's true, Greg, that even now, a century after Gödel's birth, incompleteness remains controversial. I just discovered two recent essays by important mathematicians, Paul Cohen and Jack Schwartz.[3] Have you seen these essays?

GC: No.

CC: Listen to what Cohen has to say:

> "I believe that the vast majority of statements about the integers are totally and permanently beyond proof in any reasonable system."

And according to Schwartz,

> "truly comprehensive search for an inconsistency in any set of axioms is impossible."

GC: Well, my current model of mathematics is that it's a living organism that develops and evolves, forever. That's a long way from the traditional Platonic view that mathematical truth is perfect, static and eternal.

CC: What about Einstein's famous statement that

> "Insofar as mathematical theorems refer to reality, they are not sure, and insofar as they are sure, they do not refer to reality."

[3] P. J. Cohen, "Skolem and pessimism about proof in mathematics," *Phil. Trans. R. Soc. A* (2005) **363**, 2407–2418; J. T. Schwartz, "Do the integers exist? The unknowability of arithmetic consistency," *Comm. Pure & Appl. Math.* (2005) **LVIII**, 1280–1286.

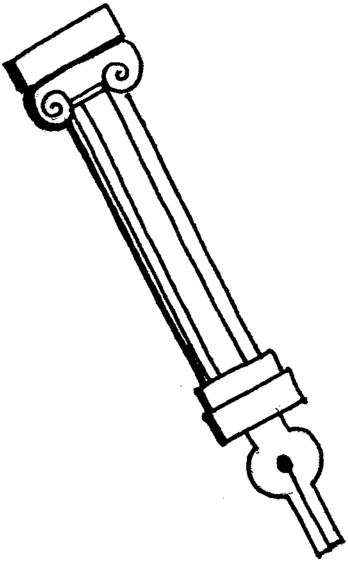
Still valid?

GC: Or, slightly misquoting Pablo Picasso, theories are lies that help us to see the truth!

CC: Perhaps we should adopt Svozil's attitude of "suspended attention" (a term borrowed from psychoanalysis) about the relationship between mathematics and physics. . .

GC: Deep philosophical questions are never resolved, you just get tired of discussing them. Enough for today!

The EATCS Columns



The Algorithmics Column

by

Gerhard J Woeginger

Department of Mathematics and Computer Science
Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
gwoegi@win.tue.nl

Some problems around travelling salesmen, dart boards, and Euro-coins

Vladimir G. Deĭneko [*]   Gerhard J. Woeginger [†]

Abstract

In 1957 Fred Supnick investigated and solved a special case of the Travelling Salesman Problem. Since then, Supnick's results have been rediscovered many times by other researchers. This article discusses Supnick's results and some of the rediscoveries.

1 The Travelling Salesman Problem

The Travelling Salesman Problem (TSP, for short) is probably the most prominent and the most studied problem in combinatorial optimization. An instance of the TSP consists of n cities $1, 2, \ldots, n$ whose distances $d_{i,j}$ are summarized in an $n \times n$

[*] Email: orsvd@wbs.warwick.ac.uk. Warwick Business School, The University of Warwick, Coventry CV4 7AL, United Kingdom.

[†] Email: gwoegi@win.tue.nl. Department of Mathematics and Computer Science, TU Eindhoven, P.O. Box 513, 5600 MB Eindhoven, The Netherlands.

distance matrix $D = (d_{i,j})$. The goal is to find a shortest closed tour through these cities. In other words, a travelling salesman starts from his home city $\phi(1)$, then visits each of the other $n-1$ cities $\phi(2), \phi(3), \ldots, \phi(n)$ exactly once, in the end returns to his home city $\phi(1)$, and he does all this with the smallest possible amount of gas. Mathematically speaking, we wish to find a permutation ϕ of $1, 2, \ldots, n$ that minimizes the value of

$$\left(\sum_{i=1}^{n-1} d_{\phi(i),\phi(i+1)}\right) + d_{\phi(n),\phi(1)} \tag{1}$$

The TSP models tons of situations that arise in robotics, production, scheduling, engineering, and many other areas. For more specific information on the TSP and its applications, we refer the reader to the book by Lawler, Lenstra, Rinnooy Kan & Shmoys (1985).

The TSP in its general formulation is an NP-hard problem (Garey & Johnson, 1979), and hence computationally intractable. In this article, we will concentrate on a special case of the TSP where the underlying distance matrix is a so-called *Supnick matrix*.

2 Supnick matrices

The following inequalities (2) go back to the eighteenth century, to the work of the French mathematician and Naval minister Gaspard Monge (1781). An $n \times n$ matrix $D = (d_{i,j})$ is called Monge matrix, if it satisfies the so-called Monge inequalities

$$d_{i,j} + d_{r,s} \;\leq\; d_{i,s} + d_{r,j} \tag{2}$$

for all i, j, r, s with $1 \leq i < r \leq n$ and $1 \leq j < s \leq n$. In words: In every 2×2 sub-matrix the sum of the two entries on the main diagonal is less or equal to the sum of the two entries on the other diagonal. Burkard, Klinz & Rudolf (1996) survey the role of Monge structures in combinatorial optimization.

A *Supnick matrix $D = (d_{i,j})$* is a symmetric Monge matrix. That is, a Supnick matrix satisfies (2), and it satisfies $d_{i,j} = d_{j,i}$ for all i and j. Here is a small catalogue of Supnick matrices:

- Sum matrices:
  Let $\alpha_1, \ldots, \alpha_n$ be real numbers. Then the sum matrix D with $d_{i,j} = \alpha_i + \alpha_j$ is a Supnick matrix. In fact, a sum matrix satisfies all inequalities (2) even with equality.

- Convex-function matrices:
  Let $f : \mathbb{R} \to \mathbb{R}$ be a function that is symmetric (hence: $f(x) = f(-x)$ for all

x) and convex (hence: $f(x+\delta) - f(x) \leq f(y+\delta) - f(y)$ for all $x \leq y$ and all $\delta \geq 0$). Let $\beta_1 \leq \beta_2 \leq \cdots \leq \beta_n$ be real numbers.

Then matrix D with $d_{i,j} = f(\beta_i - \beta_j)$ is a Supnick matrix: Symmetry of f implies symmetry of D. For i, j, r, s with $1 \leq i < r \leq n$ and $1 \leq j < s \leq n$, we set $x := \beta_i - \beta_s$, $y := \beta_r - \beta_s$, and $\delta = \beta_s - \beta_j$. This yields $x \leq y$ and $\delta \geq 0$. Plugging these values into the convexity condition yields (2).

- LL-UR block matrices:
  Let $1 \leq x < y \leq n$ be integers, and consider the Lower-Left Upper-Right block matrix D with $d_{i,j} = 1$ if $i \leq x$ and $j \geq y$ or if $i \geq y$ and $j \leq x$, and with $d_{i,j} = 0$ in all other cases. It is easily verified that this matrix D is a Supnick matrix. It has a rectangular block of 1-entries in the lower left corner (below the main diagonal), a symmetric block of 1-entries in the upper right corner (above the main diagonal), and it has 0-entries everywhere else. See Figure 1 for an illustration.

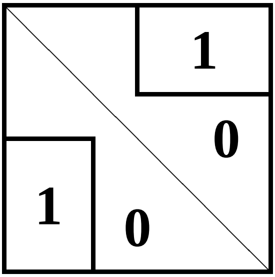


Figure 1: A Lower-Left Upper-Right block matrix.

Note that the inequalities stated in (2) are *linear* inequalities, and that also the symmetry condition is a linear condition. Consequently, if we multiply a Supnick matrix by a non-negative real number, or if we add up two Supnick matrices, then the resulting matrix will again be a Supnick matrix: The Supnick matrices form a cone. Rudolf & Woeginger (1995) took a closer look at the structure of this cone and its extremal rays, and they came up with the following simple characterization of Supnick matrices.

Theorem 1. *A matrix is a Supnick matrix, if and only if it can be written as the sum of a sum matrix S and a non-negative linear combination of LL-UR block matrices.*

3 Fred Supnick's theorem

Now let us return to the travelling salesman problem. Fred Supnick (1957) proved by a (somewhat involved) exchange argument that for the TSP with Supnick distance matrices, the optimal TSP tour is easy to find: The shortest tour is *always* given by the same permutation $\sigma^{\min}$. This permutation $\sigma^{\min}$ first visits the odd cities in increasing order and then the even cities in decreasing order, and it constitutes a universally optimal solution for all instances of the Supnick TSP.

Theorem 2. *Let $D = (d_{i,j})$ be an $n \times n$ Supnick matrix. The shortest TSP tour is given by the permutation*

$$\sigma^{\min} = \langle 1, 3, 5, 7, 9, 11, 13, \ldots 14, 12, 10, 8, 6, 4, 2 \rangle. \tag{3}$$

The longest TSP tour is given by the permutation

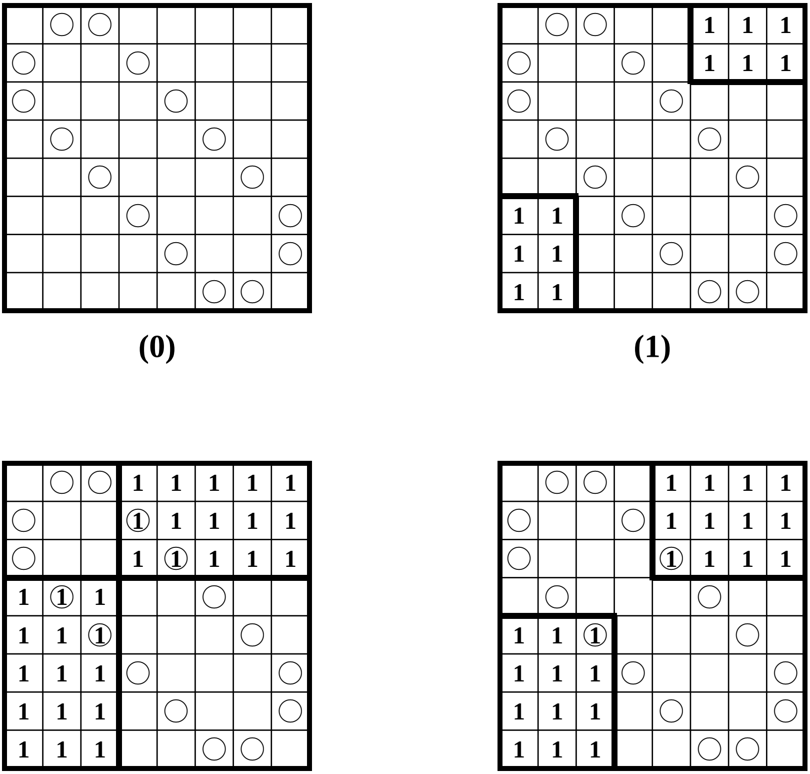
$$\sigma^{\max} = \langle n, 2, n-2, 4, n-4, 6, \ldots, 5, n-3, 3, n-1, 1 \rangle. \tag{4}$$

(We write $\phi = \langle \phi(1), \phi(2), \ldots, \phi(n) \rangle$ to specify a permutation ϕ.) In the following paragraphs we will present a simple and quite straightforward argument for Supnick's result on the shortest tour. The argument is based on the additive characterization of Supnick matrices stated in Theorem 1. A similar argument can be used to prove Supnick's result on the longest tour.

The TSP with a sum distance matrix D with $d_{i,j} = \alpha_i + \alpha_j$ is absolutely uninteresting: Since every city i contributes the value $2\alpha_i$ to the total tour length, every possible tour has the length $2\sum_{i=1}^{n} \alpha_i$. Every permutation ϕ minimizes (and simultaneously maximizes) the value of the expression in (1). In particular, the Supnick permutation $\sigma^{\min}$ yields a shortest tour for the TSP on sum matrices.

The TSP on LL-UR block matrices is slightly more interesting. For technical reasons, we will now *double* every TSP tour and traverse it once in forward and once in backward direction. Since the distances are symmetric, this simply doubles the total tour length. Shortest solutions remain shortest, and non-shortest solutions remain non-shortest. Let us take a closer look at such a doubled tour corresponding to the permutation $\sigma^{\min}$: In the forward direction, the doubled tour runs from city 1 to city 3, from city 3 to city 5, from 5 to 7 and so on. In the backward direction, it runs from city 2 to city 4, from 4 to 6, and so on. Hence, the doubled tour picks the entries $d_{i,i+2}$ and $d_{i+2,i}$ for $i = 1, \ldots, n-2$ together with the four entries $d_{1,2}$, $d_{2,1}$, $d_{n-1,n}$, $d_{n,n-1}$ out of the distance matrix, and it pays their total value. All the picked entries lie in the two diagonals above and in the two diagonals below the main diagonal of D. See Figure 2.0 for an illustration.

Now let us argue that the doubled tour for $\sigma^{\min}$ is the shortest doubled tour for any LL-UR block matrix D. We distinguish three cases that depend on the size and position of the two rectangular blocks of 1-entries. We recall that the lower left corner of the upper right block is the matrix element $d_{x,y}$ with $1 \leq x < y \leq n$.



(0)

(1)

(2)

(3)

Figure 2: Illustrations for the proof of Supnick's result.

Case 1: If $y - x \geq 3$, then the rectangular blocks in matrix D do not touch the doubled tour $\sigma^{\min}$; see Figure 2.1. Then the corresponding cost is 0, which clearly is minimum.

Case 2: If $y - x = 1$, then the rectangular blocks in D cover four of the entries picked by the doubled tour $\sigma^{\min}$, and the corresponding cost equals 4. See Figure 2.2 for an illustration. In this case the cities in $G_1 = \{1, \ldots, x\}$ are pairwise at distance 0, and the cities in $G_2 = \{x + 1, \ldots, n\}$ are pairwise at distance 0. The distance between any city in G_1 and any city in G_2 equals 1. Any TSP tour must go at least once from G_1 into G_2, and at least once back from G_2 into G_1. Hence, in this case any tour has cost at least 2 and any doubled tour has cost at least 4. Again, $\sigma^{\min}$ yields a shortest solution.

Case 3: The last case $y - x = 2$ is illustrated in Figure 2.3. Now the cost of the doubled tour $\sigma^{\min}$ equals 2. Every TSP tour must contain some move from a city with number $\leq x$ to a city with number $\geq y$, or a move from a city $\geq y$ to a city $\leq x$. Therefore any tour has cost at least 1, any doubled tour has cost at least 2, and also in this case $\sigma^{\min}$ yields a shortest solution.

Summarizing, we have shown that permutation $\sigma^{\min}$ yields the shortest TSP tour for every distance matrix that is an LL-UR block matrix and for every distance matrix that is a sum matrix. Then $\sigma^{\min}$ also yields the shortest TSP tour for any non-negative linear combination of such matrices, and by Theorem 1 these combinations are exactly the Supnick matrices. The argument is complete.

4 Rediscoveries of Supnick's theorem

During the Cold War, scientific results were often discovered on one side of the Iron Curtain and rediscovered independently on the other side. This also happened to Theorem 2. Supnick derived his result in 1957, when he worked at the City University of New York. Supnick's result was rediscovered once by Rubinshtein (1971) in Russia and once by Michalski (1987) in Poland. Both rediscoveries state Supnick's result in its full generality and in the language of the TSP.

Other researchers only rediscovered special cases of Theorem 2. For instance, Chao & Liang (1992) derived the special case of Theorem 2 where the underlying matrix is a convex-function matrix. Another rediscovery is problem B-3 of the 57th William Lowell Putnam Mathematical Competition, which reads as follows:

> Given that $\{x_1, x_2, \ldots, x_n\} = \{1, 2, \ldots, n\}$, find the largest possible value of $x_1x_2 + x_2x_3 + \cdots + x_{n-1}x_n + x_nx_1$ as a function of $n \geq 2$.

This Putnam problem asks for the longest TSP tour in the distance matrix $D = (d_{i,j})$ with $d_{i,j} = ij$. Matrix D itself is not a Supnick matrix, but matrix $-D$ is a Supnick matrix. Since the longest tour for D corresponds to the shortest tour for $-D$, the problem is solved by permutation $\sigma^{\min}$. Theorem 2 then yields the answer $\frac{1}{6}(2n^3 + 3n^2 - 11n + 18)$.

Dart boards. Now let us turn to a number of rediscoveries of Supnick's result that are centered around the game of darts. The arrangement of the numbers $1, 2, \ldots, 20$ on a modern dart board was devised in 1896 by Brian Gamlin, a carpenter from Bury in the County of Lancashire. See the left half of Figure 3 for an illustration. Gamlin's arrangement reduces the element of chance and encourages accurate play, since large numbers (good scores) are always placed between

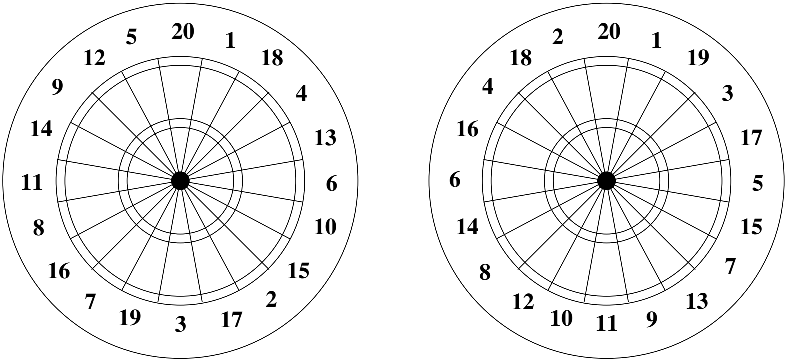


Figure 3: A classical dart board to the left. The most difficult dart board to the right.

small numbers (bad scores). For instance, the large number 20 at the top of the dart board is placed between the small numbers 1 and 5. If you are aiming for the segment 20, then a poor shot is penalized by a low score of 1 or 5.

Why did Gamlin select this particular arrangement? Is there some simple quality criterion that uniquely singles out the Gamlin arrangement from all possible arrangements? Unfortunately, no such simple quality criterion is known. Some Mathematicians are worried about this, and over the years they have proposed and analyzed a considerable number of 'reasonable' quality criteria; see for instance Singmaster (1980) and Lipscombe & Sangalli (2000). The most popular quality criterion is the so-called L_p-criterion: Here the penalty incurred for missing a segment x and hitting the adjacent segment y instead equals $|x-y|^p$. The best (that is, most difficult) dart board with respect to the L_p-criterion is a permutation of the numbers $1, 2, \ldots, 20$ (or generally of the numbers $1, 2, \ldots, n$) that maximizes the total penalty of all segments. The reader will have little difficulty to recognize that the most difficult dart board for the L_p-criterion corresponds to the longest TSP tour in the convex-function matrix with $f(x) = |x|^p$ and $\beta_i = i$ for $i = 1, \ldots, n$. Theorem 2 yields that the best number arrangement is $\sigma^{\max}$. For $n = 20$ this arrangement is

$$\langle 20, 2, 18, 4, 16, 6, 14, 8, 12, 10, 11, 9, 13, 7, 15, 5, 17, 3, 19, 1 \rangle, \tag{5}$$

and the corresponding most difficult dart board is depicted in the right half of Figure 3.

Selkirk (1976) was probably the first to discuss the L_1 and the L_2-criterion for dart boards. He correctly identifies the permutation $\sigma^{\max}$, and he states that for L_2

the highest possible total penalty equals $\frac{1}{3}(n^3 - 4n + 3)$ if n is odd and $\frac{1}{3}(n^3 - 4n + 6)$ if n is even. Selkirk states a number of assertions all of which are correct, but none of which are proved. Eiselt & Laporte (1991) formulate L_1 and L_2 as a maximum cost TSP, and then compute an optimal solution by using a branch-and-bound code. They only consider the case $n = 20$, and their computer program indeed comes up with the arrangement in (5). Everson & Bassom show that permutation $\sigma^{\max}$ maximizes the L_1-criterion for arbitrary values of n. Cohen & Tonkes (2001) prove by an exchange argument that permutation $\sigma^{\max}$ maximizes the total penalty under the L_p-criterion for every integer $p \geq 1$. Curtis (2004) reproves the results of Cohen & Tonkes (2001), but by a different approach that is based on a certain greedy algorithm. Curtis also discusses so-called hoopla boards, and thereby rediscovers Supnick's result for convex-function matrices with $f(x) = |x|^p$ and *arbitrary* values $\beta_1 \leq \beta_2 \leq \cdots \leq \beta_n$.

Problem 10725 in the American Mathematical Monthly (Mihai & Woltermann, 2001) reads as follows:

> Fix a positive integer n. Given a permutation ϕ of $\{1, 2, \ldots, n\}$, let $F(\phi) = \sum_{i=1}^{n}(\phi(i) - \phi(i+1))^2$, where $\phi(n+1) = \phi(1)$. Find the extreme values of $F(\phi)$ as ϕ ranges over all permutations.

Obviously, this problem asks for the easiest and for the most difficult dart board under the L_2-criterion. The minimum of $F(\phi)$ is $4n - 6$, and the maximum is $\frac{1}{3}(n^3 - 4n + 3)$ if n is odd and $\frac{1}{3}(n^3 - 4n + 6)$ if n is even (as also observed by Selkirk, 1976).

Euro-coins. The diameter of a 1-Euro coin is 23.25 mm, and the diameter of a 2-Euro coin is 25.75 mm. There are only two essentially different ways of arranging two 2-Euro coins and three 1-Euro coins in a ring, so that each coin is tangent to two others while all five coins are externally tangent to a disk inside the ring. See Figure 4 for an illustration. For which of the two arrangements is the diameter of the inner disk larger? This puzzle goes back to Joe Konhauser, and it is discussed in problem 43 of the book *"Which Way Did the Bicycle Go?"* by Konhauser, Velleman & Wagon (1996). It turns out that the arrangement with adjacent 2-Euro coins has a slightly larger central disk. Duncan, Velleman & Wagon (1996) discuss a generalization of this puzzle.

> Suppose we are given $n \geq 3$ disks of radii $r_1 \leq r_2 \leq \cdots \leq r_n$. We wish to place them in some order around a central disk so that each given disk touches the central disk and its two immediate neighbors. If the given disks are of widely different sizes (such as 100, 100, 100, 100, 1), we allow a disk to overlap other given disks that are not immediate

> neighbors. In what order should the given disks be arranged so as to maximize the radius of the central disk?

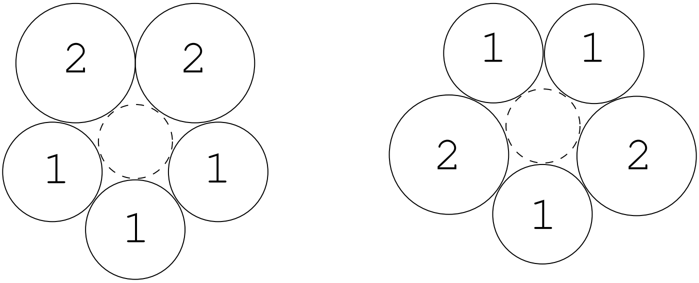


Figure 4: There are two ways of arranging three 1-Euro coins and two 2-Euro coins around a central disk: 2-Euro coins together (left) or 2-Euro coins apart (right). In the right picture the central disk is slightly larger than in the left picture.

Let us first consider a central disk with a fixed radius R. Look at a single tangent configuration made up of the central disk, and two disks of radius x and y. The three centers form a triangle with sides $R + x$, $R + y$, and $x + y$. Applying the law of cosines and simplifying gives that in this triangle, the angle at the center of the disk with radius R is

$$\theta_R(x, y) \;=\; \arccos\left(1 - \frac{2xy}{R^2 + Rx + Ry + xy}\right). \tag{6}$$

It can be checked that the distances $d_{i,j} = -\theta_R(r_i, r_j)$ are symmetric and satisfy the inequalities (2). Therefore, the corresponding distance matrix D is a Supnick matrix, and Theorem 2 can be applied. Because of the minus-sign in the definition of the $d_{i,j}$, the permutation $\sigma^{\min}$ in this case yields the arrangement with largest overall angle $\theta(R)$. If the overall angle $\theta(R)$ is strictly less than 2π, then the radius R was chosen too large and must be decreased. If the overall angle $\theta(R)$ is strictly greater than 2π, then the radius R was chosen too small and must be increased. To summarize, the arrangement $\sigma^{\min}$ maximizes and the arrangement $\sigma^{\max}$ minimizes the radius of the central disk. This has been (re)discovered by Duncan, Velleman & Wagon (1996).

If we go back to the puzzle of the five Euro-coins with radii $r_1 = 23.25$, $r_2 = 23.25$, $r_3 = 23.25$, $r_4 = 25.75$, $r_5 = 25.75$, we see that the maximizing permutation $\sigma^{\min} = \langle 1, 3, 5, 4, 2\rangle$ indeed puts the two larger coins next to each other.

References

[1] R.E. Burkard, B. Klinz, and R. Rudolf (1996). Perspectives of Monge properties in optimization. *Discrete Applied Mathematics 70,* 95–161.

[2] C.C. Chao and W.Q. Liang (1992). Arranging *n* distinct numbers on a line or a circle to reach extreme total variations *European Journal of Combinatorics 13,* 325–334.

[3] G.L. Cohen and E. Tonkes (2001). Dartboard arrangements. *Electronic Journal of Combinatorics 8(2),* R4.

[4] S.A. Curtis (2004). Darts and hoopla board design. *Information Processing Letters 92,* 53–56.

[5] J. Duncan, D. Velleman, and St. Wagon (1996). Solution to Problem 2006. *Crux Mathematicorum 22,* 37–38.

[6] H.A. Eiselt and G. Laporte (1991). A combinatorial optimization problem arising in dartboard design. *The Journal of the Operational Research Society 42,* 113–118.

[7] P.J. Everson and A.P. Bassom (1994/5). Optimal arrangements for a dartboard. *Mathematical Spectrum 27,* 32–34.

[8] M.R. Garey and D.S. Johnson (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco.

[9] J.D.E. Konhauser, D. Velleman, and St. Wagon (1996). Problem 43 in *"Which Way Did the Bicycle Go?"* Dolciani Mathematical Expositions.

[10] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys (eds.) (1985). *The travelling salesman problem*. John Wiley, Chichester.

[11] T. Lipscombe and A. Sangalli (2000). The devil's dartboard. *Crux Mathematicorum 26,* 215–217.

[12] M. Michalski (1987). On a class of polynomially solvable travelling salesman problems. *Zastosowania Matematyki 19,* 531–539.

[13] V. Mihai and M. Woltermann (2001). Problem 10725: The smoothest and roughest permutations. *American Mathematical Monthly 108,* 272–273.

[14] G. Monge (1781). Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Science (Année 1781, avec les Mémoires de Mathématique et de Physique, pour la même Année),* 666–704.

[15] M.I. Rubinshtein (1971). On the symmetric travelling salesman problem. *Automation and Remote Control 32,* 1453–1460.

[16] R. Rudolf and G.J. Woeginger (1995). The cone of Monge matrices: Extremal rays and applications. *Mathematical Methods of Operations Research 42,* 161–168.

[17] K. Selkirk (1976). Re-designing the dartboard. *Mathematical Gazette 60,* 171–178.

[18] D. Singmaster (1980). Arranging a dartboard. *Bulletin of the Institute of Mathematics and its Applications 16,* 93–97.

[19] F. Supnick (1957). Extreme Hamiltonian lines. *Annals of Mathematics 66,* 179–201.

The Computational Complexity Column

by

Jacobo Torán

Dept. Theoretische Informatik, Universität Ulm
Oberer Eselsberg, 89069 Ulm, Germany
jacobo.toran@uni-ulm.de
http://theorie.informatik.uni-ulm.de/Personen/jt.html

Error-correcting codes were initially introduced to cope with the problem of unreliable communication. In the last years however, many applications for these codes in the fields of complexity theory and cryptography have been found, being the PCP Theorem probably the most spectacular application example. Conversely the use of complexity techniques has enabled to improve code constructions and to develop more efficient coding and decoding algorithms. In the present column Venkatesan Guruswami gives a thorough introduction to one of these algorithmic aspects, reporting about the area of iterative algorithms for decoding low-density parity check codes.

Iterative Decoding of Low-Density Parity Check Codes

An Introductory Survey

Venkatesan Guruswami[*]

Abstract

Much progress has been made on decoding algorithms for error-correcting codes in the last decade. In this article, we give an introduction

[*] Department of Computer Science and Engineering, University of Washington, Seattle, WA 98195, U.S.A. Supported in part by NSF CCF-0343672, an Alfred P. Sloan Research Fellowship, and a David and Lucile Packard Foundation Fellowship.

to some fundamental results on iterative, message-passing algorithms for low-density parity check codes. For certain important stochastic channels, this line of work has enabled getting very close to Shannon capacity with algorithms that are extremely efficient (both in theory and practice).

1 Introduction

Over the past decade or so, there has been substantial new progress on algorithmic aspects of coding theory. A (far from exhaustive) list of the themes that have witnessed intense research activity includes:

1. A resurgence of interest in the long forgotten class of low-density parity check (LDPC) codes and on iterative, message-passing decoding algorithms for them, which has resulted in codes with rates extremely close to Shannon capacity together with efficient decoding algorithms.

2. Linear time encodable/decodable error-correcting codes (based on expanders) for worst-case errors.

3. List decoding algorithms which correct many more worst-case errors beyond the "half-the-code-distance" bound, and which can achieve capacity even against adversarial noise.[1]

Of course there are some interrelations between the above directions; in particular, progress on linear-time encodable/decodable codes is based on expander codes, which are LDPC codes with additional properties. Also, list decoding algorithms that run in linear time and correct a fraction ρ of errors for any desired $\rho < 1$ have been developed using expander-based ideas [12].

Of the above lines of work, the last two have a broader following in the theoretical computer science community, due to their focus on the combinatorial, worst-case noise model and the extraneous applications of such codes in contexts besides communication (such as pseudorandomness and average-case complexity). The sister complexity theory column that appears in SIGACT news featured recent surveys on both these topics [9, 32]. A longer survey on very recent developments in list decoding of algebraic codes will appear in [10]. A very brief survey featuring couple of complexity-theoretic uses of list decoding appears in [11]. Applications of coding theory to complexity theory, especially those revolving around sub-linear algorithms, are surveyed in detail in [34].

[1] The capacity-achieving part was recently shown for codes over *large* alphabets, specifically explicit codes of rate close to $1 - p$ that can be list decoded in polynomial time from a fraction p of errors were constructed in [14]. For binary codes, the capacity for decoding a fraction p of errors equals $1 - H(p)$, but we do not know how to achieve this constructively.

We use the opportunity provided by this column to focus on the first line of work on iterative (also called message-passing or belief propagation) algorithms for decoding LDPC codes. This is in itself a vast area with numerous technically sophisticated results. For a comprehensive discussion of this area, we point the reader to the upcoming book by Richardson and Urbanke [25], which is an excellent resource on this topic. The February 2001 issue of Volume 47 of the IEEE Transactions on Information Theory is another valuable resource — this was a special issue dedicated to iterative decoding and in particular contains the series of papers [16, 17, 23, 22]. This sequence of papers is arguably one of the most important post-Gallager developments in the analysis of iterative decoding, and it laid down the foundations for much of the recent progress in this field.

Disclaimer: The literature on the subject of LDPC and related codes and belief propagation algorithms is vast and diverse, and the author, not having worked on the topic himself, is only aware of a small portion of it. Our aim will be to merely provide a peek into some of the basic context, results, and methods of the area. We will focus almost exclusively on LDPC codes, and important related constructions such as LT codes, Raptor codes, Repeat-Accumulate codes, and turbo codes are either skipped or only very briefly mentioned. While the article should (hopefully) be devoid of major technical inaccuracies, we apologize for any inappropriate omissions in credits and citations (and welcome comments from the reader if any such major omissions are spotted).

Organization: We begin with some basic background information concerning LDPC codes, the channel models we will study, and the goal of this line of study in Section 2. In Section 3, we discuss how concatenated codes with an outer code that can correct a small fraction of errors can be used to approach capacity, albeit with a poor dependence on the gap to capacity. We then turn to message passing algorithms for LDPC codes and describe their high level structure in Section 4. With this in place, we develop and analyze some specific message passing algorithms for *regular* LDPC codes in Section 5, establishing theoretical thresholds for the binary erasure and binary symmetric channels. We then turn our focus to *irregular* LDPC codes in Section 6, and discuss, among other things, how one can use them to achieve the capacity of the binary erasure channel. Finally, in Section 7, we discuss how one can achieve linear encoding time for LDPC codes, and also discuss a variant called Irregular Repeat-Accumulate (IRA) codes that are linear-time encodable by design and additionally offer improved complexity-vs-performance trade-offs.

2 Background

2.1 Linear and LDPC codes

We will focus exclusively on binary linear codes. A binary linear code C of *block length* n is a subspace of $\mathbb{F}_2^n$ where $\mathbb{F}_2 = \{0, 1\}$ is the field with two elements. The rate of C, denoted $R(C)$, equals k/n where k is the dimension of C (as a vector space over $\mathbb{F}_2$); such a code is also referred to as an $[n, k]$ code. Being a linear subspace of dimension k, the code C can be described as the kernel of a matrix $H \in \mathbb{F}_2^{(n-k)\times n}$, so that $C = \{c \in \mathbb{F}_2^n \mid Hc = 0\}$ (we treat codewords c as column vectors for this description). The matrix H is called the *parity check matrix* of the code C. In general, any choice of H whose rows form a basis of the dual space $C^\perp = \{x \in \mathbb{F}_2^n \mid x^t c = 0 \forall c \in C\}$ describes the same code. Of special interest to us here are codes that admit a *sparse* parity check matrix. In particular, we will study *low-density parity check* (LDPC) codes, which were introduced and studied in Gallager's amazing work [8] that was way ahead of its time. LDPC codes are described by a parity check matrix all of whose rows and columns have at most a fixed constant number of 1's (the constant is independent of n).[2]

A convenient way to describe an LDPC code is in terms of its *factor graph*.[3] This is a natural bipartite graph defined as follows. On the left side are n vertices, called *variable* nodes, one for each codeword position. On the right are $m = n - k$ vertices, called *check* nodes, one for each parity check (row of the parity check matrix). A check node is adjacent to all variable nodes whose corresponding codeword symbols appear in this parity check. In other words, the parity check matrix of the code is precisely the bipartite adjacency matrix of the factor graph.

A special class of LDPC codes are regular LDPC codes where the factor graph is both left-regular and right-regular. Regular LDPC codes were in fact the variant originally studied by Gallager [8], as well as in the works of Mackay and Neal [18, 19] and Sipser and Spielman [29, 30] that sparked the resurgence of interest in LDPC codes after over 30 years since Gallager's work.[4] LDPC codes based on non-regular graphs, called irregular LDPC codes, rose to prominence beginning in the work of Luby *et al* [16, 17] (studying codes based on irregular graphs was

[2] We will throughout be interested in a family of codes of increasing block length n with rate k/n held a fixed constant. For convenience, we don't spell this out explicitly, but this asymptotic focus should always be kept in mind.

[3] This graphical representation applies for any linear code. But the resulting graph will be sparse, and hence amenable to linear time algorithms, only for LDPC codes.

[4] In the long interim period, LDPC codes went into oblivion, with the exception of two (known to us) works. Zyablov and Pinsker [35] proved that for random LDPC codes, with high probability over the choice of the code, Gallager's algorithm corrected a constant fraction of *worst-case* errors. Tanner [33] presented an important generalization of Gallager's construction and his decoding algorithms, which was later important in the work on linear time decodable expander codes [29].

one of the big conceptual leaps made in these works). We will return to this aspect later in the survey. A popular choice of regular LDPC codes (with a rate of $1/2$) are $(3, 6)$-regular LDPC codes where variable nodes have degree 3 and check nodes have degree 6.

2.2 Channel models and their capacity

Design of good LDPC codes, together with progress in analyzing natural message-passing algorithms for decoding them, has led to rapid progress towards approaching the capacity of important stochastic channels. We now review the main noise models that we will be interested in.

Throughout, we deal with binary codes only. We will find it convenient to use $\{+1, -1\}$ (instead of $\{0, 1\}$) for the binary alphabet, where $+1$ corresponds to the bit 0 and -1 to the bit 1. Note the XOR operation becomes multiplication in the ± 1 notation.

We will assume the channel's operation to be *memoryless*, so that each symbol of the codeword is distorted independently according to the same channel law. So to specify the noise model, it suffices to specify how the noise distorts a single input symbol. For us the input symbol will always be either ± 1, and so the channels have as input alphabet $\mathcal{X} = \{1, -1\}$. Their output alphabet will be denoted by $\mathcal{Y}$ and will be different for the different channels. Upon transmission of a codeword $c \in \mathcal{X}^n$, the word y observed by the receiver belongs to $\mathcal{Y}^n$. The receiver must then decode y and hopefully compute the original transmitted codeword c. The challenge is to achieve a vanishingly small error probability (i.e., the probability of either a decoding failure or an incorrect decoding), while at the same time operating at a good rate, hopefully close to the capacity of the channel.

We begin with the simplest noise model, the *Binary Erasure Channel* (BEC). This is parameterized by a real number α, $0 \leq \alpha < 1$. The output alphabet is $\mathcal{Y} = \{1, -1, ?\}$, with ? signifying an *erasure*. Upon input $x \in \mathcal{X}$, the channel outputs x with probability $1 - \alpha$, and outputs ? with probability α. The value α is called the erasure probability, and we denote by BEC_α the BEC with erasure probability α. For large n, the received word consists of about $(1 - \alpha)n$ unerased symbols with high probability, so the maximum rate at which reliable communication is possible is at most $(1 - \alpha)$ (this holds even if the transmitter and receiver knew in advance which bits will be erased). It turns out this upper bound can be achieved, and Elias [5], who first introduced the BEC, also proved that its capacity equals $(1 - \alpha)$.

The *Binary Symmetric Channel* (BSC) is parameterized by a real number p, $0 \leq p < 1/2$, and has output alphabet $\mathcal{Y} = \{1, -1\}$. On input $x \in \mathcal{X}$, the channel outputs bx where $b = -1$ with probability p and $b = 1$ with probability $1 - p$. The value p is called the *crossover probability*. The BSC with crossover probability p

is denoted by BSC_p. The capacity of BSC_p is well known to be $1 - H(p)$, where $H(p) = -p \lg p - (1 - p) \lg(1 - p)$ is the binary entropy function.

Finally, we mention a channel with continuous output alphabet $\mathcal{Y}$ called *Binary Input Additive White Gaussian Noise* (BIAWGN). Here $\mathcal{Y}$ equals the set of real numbers, and the channel operation is modeled as $y = x + z$ where $x \in \{\pm 1\}$ is the input and z is a normal variable with mean 0 and variance σ^2 (i.e., has probability density function $p(z) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{z^2}{2\sigma^2}}$). We denote by BIAWGN_σ the BIAWGN with variance σ^2; its capacity is a function of $1/\sigma^2$ alone, though there is no elementary form expression known for the capacity (but it can be expressed as an integral that can be estimated numerically). For rate 1/2, the largest σ (Shannon limit) for which reliable communication on the BIAWGN channel is possible is (up to the precision given) $\sigma_{\text{opt}} = 0.9787$.

More generally, if we allow scaling of inputs, the capacity is a function of the "signal-to-noise" ratio E_N/σ^2 where E_N is the energy expended per channel use. If the inputs to the channel are not constrained to be ±1, but instead can take arbitrary real values, then it is well known that the capacity of the AWGN channel equals $\frac{1}{2} \log_2 \left(1 + E_N/\sigma^2\right)$ bits per channel use. In particular, in order to achieve reliable communication at a rate of 1/2 over the real-input AWGN channel, a signal-to-noise ratio of 1, or 0 dB, is required.[5] For the BIAWGN channel, this ratio increases to $1/\sigma^2_{\text{opt}} = 1.044$ or 0.187 dB. Accordingly, the yardstick to measure the quality of a decoding algorithm for an LDPC code of rate 1/2 is how close to this limit it can lead to correct decoding with probability tending to 1 (over the realization of the BIAWGN channel noise).

The continuous output of a BIAWGN channel can be quantized to yield a discrete approximation to the original value, which can then be used in decoding. (Of course, this leads to loss in information, but is often done for considerations of decoding complexity.) A particularly simple quantization is to decode a signal x into 1 if $x \geq 0$ and into -1 if $x < 0$. This effectively converts an AWGN channel with variance σ^2 into a BSC with crossover probability $Q(1/\sigma) = \frac{1}{\sqrt{2\pi}} \int_{1/\sigma}^{\infty} e^{-x^2/2} dx$. It should not come as a surprise that the capacity of the resulting BSC falls well short of the capacity of the BIAWGN.

All the above channels have the following *output-symmetry* property: For each possible channel output q, $p(y = q|x = 1) = p(y = -q|x = -1)$. (Here $p(y|x)$ denotes the conditional probability that the channel output equals y given the channel input is x.)

We will focus a good deal of attention on the BEC. Being a very simple channel, it serves as a good warm-up to develop the central ideas, and at the same time achieving capacity on the BEC with iterative decoding of LDPC codes is technically non-trivial. The ideas which were originally developed for erasure codes in

[5] In decibel notation, $\lambda > 0$ is equivalent to $10 \log_{10} \lambda$ dB.

[16] have been generalized for more general channels, including the BSC and BI-AWGN, with great success [17, 23, 22]. Yet, to date the BEC is the only channel known for which one can provably get arbitrarily close to capacity via iterative decoding of (an ensemble of) LDPC codes. So naturally, given our focus on the theoretical aspects, the BEC is of particular interest.

2.3 Spirit of the results

The central goal of research in channel coding is the following: given a particular channel, find a family of codes which have fast (ideally linear-time) encoding algorithms and which can be reliably decoded in linear time at rates arbitrarily close to channel capacity. This is, of course, also the goal of the line of work on LDPC codes.

In "practice" one of the things that seems to get people excited are plots of the signal-to-noise ratio (SNR) vs bit error probability (BER) for finite-length codes found by non-trivial optimization based on theoretical insights, followed by simulation on, say, the BIAWGN channel. Inspired by the remarkable success on the BEC [16], this approach was pioneered for LDPC codes in the presence of errors in [31, 17], culminating in the demonstration of codes for the BIAWGN channel in [22] that beat turbo codes and get very close to the Shannon limit.

Since this article is intended for a theory audience, our focus will be on the "worst" channel parameter (which we call threshold) for which one can prove that the decoding will be successful with probability approaching 1 in the asymptotic limit as the block length grows to infinity. The relevant channel parameters for the BEC, BSC, and BIAWGN are, respectively, the erasure probability, crossover probability, and the variance of the Gaussian noise. The threshold is like the random capacity for a *given* code (or ensemble of codes) and a *particular* decoder. Normally for studying capacity we fix the channel and ask what is the largest rate under which reliable communication is possible, whereas here we fix the rate and ask for the worst channel under which probability of miscommunication tends to zero. Of course, the goal is to attain as a large a threshold as possible, ideally approaching the Shannon limit (for example, $1 - \alpha$ for BEC_α and $1 - H(p)$ for BSC_p).

3 Simple concatenated schemes to achieve capacity on BEC and BSC

We could consider the channel coding problem solved (at least in theory) on a given channel if we have explicit codes, with efficient algorithms for encoding and reliable decoding at rates within any desired ε of capacity. Ideally, the run time of the algorithms should be linear in the block length n, and also depend polynomially on $1/\varepsilon$. (But as we will see later, for certain channels like the BEC, we can have a runtime of $O(n \log(1/\varepsilon))$, or even better cn with c independent of ε, if we allow randomization in the construction.) In this section, we discuss some "simple" attacks on this problem for the BEC and BSC, why they are not satisfactory, and the basic challenges this raises (some of which are addressed by the line of work on LDPC codes).

For the BEC, once we have the description of the generator matrix of a linear code that achieves capacity, we can decode in $O(n^3)$ time by solving a linear system (the decoding succeeds if the system has a unique solution). Since a random linear code achieves capacity with high probability [5], we can sample a random generator matrix, thus getting a code that works with high probability (together with a cubic time algorithm). However, we do not know any method to *certify* that the chosen code indeed achieves capacity. The drawbacks with this solution are the cubic time and randomized nature of the construction.

A construction using *concatenated codes* gets around both these shortcomings. The idea originates in Forney's work [7] that was the first to present codes approaching capacity with polynomial time encoding and decoding algorithms.

Let α be the erasure probability of the BEC and say our goal is to construct a code of rate $(1 - \alpha - \varepsilon)$ that enables reliable communication on BEC_α. Let C_1 be a linear time encodable/decodable binary code of rate $(1 - \varepsilon/2)$ that can correct a small constant fraction $\gamma = \gamma(\varepsilon) > 0$ of *worst-case* erasures. Such codes were constructed in [30, 1]. For the concatenated coding, we do the following. For some parameter b, we block the codeword of C_1 into blocks of size b, and then encode each of these blocks by a suitable *inner* binary linear code C_2 of dimension b and rate $(1 - \alpha - \varepsilon/2)$. The inner code will be picked so that it achieves the capacity of the BEC_α, and specifically recovers the correct message with success probability at least $1 - \gamma/2$. For $b = b(\varepsilon, \gamma) = \Omega\left(\frac{\log(1/\gamma)}{\varepsilon^2}\right)$, a random code meets this goal with high probability, so we can find one by brute-force search (that takes constant time depending only on ε).

The decoding proceeds as one would expect: first each of the inner blocks is decoded, by solving a linear system, returning either decoding failure or the correct value of the block. (There are no errors, so when successful, the decoder knows it is correct.) Since the inner blocks are chosen to be large enough, each inner decoding fails with probability at most $\gamma/2$. Since the noise on different blocks are independent, by a Chernoff bound, except with exponentially small probability, we have at most a fraction γ of erasures in the outer codeword. These are then handled by the linear-time erasure decoder for C_1.

We conclude that, for the BEC_α, we can construct codes of rate $1 - \alpha - \varepsilon$, i.e., within ε of capacity, that can be encoded and decoded in $n/\varepsilon^{O(1)}$ time. While this is pretty good, the brute-force search for the inner code is unsatisfying, and the BEC is simple enough that better runtimes (such as $O(n \log(1/\varepsilon))$) are achieved by certain irregular LDPC codes.

A similar approach can be used for the BSC_p. The outer code C_1 must be picked so that it can correct a small fraction of worst-case *errors* — again, such codes of rate close to 1 with linear time encoding and decoding are known [30, 13]. Everything works as above, except that the decoding of the inner codes, where we find the codeword of C_2 closest to the received block, requires a brute-force search and this takes $2^b = 2^{\Omega(1/\varepsilon^2)}$ time. This can be improved to polynomial in $1/\varepsilon$ by building a look-up table, but then the size of the look-up table, and hence the space complexity and time for precomputing the table, is exponential in $1/\varepsilon$.

In summary, for the BSC_p, we can construct codes of rate $1 - H(p) - \varepsilon$, i.e., within ε of capacity, that can be encoded in $n/\varepsilon^{O(1)}$ time and which can be reliably decoded in $n2^{1/\varepsilon^{O(1)}}$ time. It remains an important open question to obtain such a result with decoding complexity $n/\varepsilon^{O(1)}$, or even $\mathrm{poly}(n/\varepsilon)$.[6]

We also want to point out that recently an alternate method using LP decoding has been used to obtain polynomial time decoding at rates arbitrarily close to capacity [6]. But this also suffers from a similar poor dependence on the gap ε to capacity.

4 Message-passing iterative decoding: An abstract view

4.1 Basic Structure

We now discuss the general structure of natural message-passing iterative decoding algorithms, as discussed, for example, in [23]. In these algorithms, messages are exchanged between the variable and check nodes in discrete time steps. Initially, each variable node v_j, $1 \leq j \leq n$, has an associated received value r_j, which is a random variable taking values in the channel output alphabet $\mathcal{Y}$. Based on this, each variable sends a message belong to some message alphabet $\mathcal{M}$. A common choice for this initial message is simply the received value r_j, or perhaps some quantized version of r_j for continuous output channels such as BIAWGN. Now, each check node c processes the messages it receives from its neighbors, and sends back a suitable message in $\mathcal{M}$ to each of its neighboring variable nodes. Upon receipt of the messages from the check nodes, each variable node v_j uses

[6] We remark that asymptotically, with ε fixed and $n \to \infty$, the exponential dependence on $1/\varepsilon$ can be absorbed into an additional factor with a slowly growing dependence on n. However, since in practice one is interested in moderate block length codes, say $n \leq 10^6$, a target runtime such as $O(n/\varepsilon)$ seems like a clean way to pose the underlying theoretical question.

these together with its own received value r_j to produce new messages that are sent to its neighboring check nodes. This process continues for many time steps, till a certain cap on the number of iterations is reached. In the analysis, we are interested in the probability of incorrect decoding, such as the bit-error probability. For every time step i, $i \in \mathbb{N}$, the i'th iteration consists of a round check-to-variable node messages, followed by the variable nodes responding with their messages to the check nodes. The 0'th iteration consists of dummy messages from the check nodes, followed by the variable nodes sending their received values to the check nodes.

A very important condition in the determination of the next message based on the messages received from the neighbors is that message sent by u along an edge *e does not depend on the message just received along edge e*. This is important so that only "extrinsic" information is passed along from a node to its neighbor in each step. It is exactly this restriction that leads to the independence condition that makes analysis of the decoding possible.

In light of the above restriction, the iterative decoding can be described in terms of the following message maps: $\Psi_v^{(\ell)} : \mathcal{Y} \times \mathcal{M}^{d_v - 1} \to \mathcal{M}$ for variable node v with degree d_v for the ℓ'th iteration, $\ell \geq 1$, and $\Psi_c^{(\ell)} : \mathcal{M}^{d_v - 1} \to \mathcal{M}$ for check node c with degree d_c. Note the message maps can be different for different iterations, though several powerful choices exist where they remain the same for all iterations (and we will mostly discuss such decoders). Also, while the message maps can be different for different variable (and check) nodes, we will use the same map (except for the obvious dependence on the degree, in case of irregular graphs).

The intuitive interpretation of messages is the following. A message is supposed to be an estimate or guess of a particular codeword bit. For messages that take ±1 values, the guess on the bit is simply the message itself. We can also add a third value, say 0, that would signify an erasure or abstention from guessing the value of the bit. More generally, messages can take values in a larger discrete domain, or even take continuous values. In these cases the sign of the message is the estimated value of the codeword bit, and its absolute value is a measure of the reliability or confidence in the estimated bit value.

4.2 Symmetry Assumptions

We have already discussed the output-symmetry condition of the channels we will be interested in, i.e., $p(y = q|x = 1) = p(y = -q|x = -1)$. We now mention two reasonable symmetry assumptions on the message maps, which will be satisfied by the message maps underlying the decoders we discuss:

- **Check node symmetry:** Signs factor out of check node message maps, i.e., for all $(b_1, \ldots, b_{d_c-1}) \in \{1, -1\}^{d_c-1}$

$$\Psi_c^{(\ell)}(b_1 m_1, \cdots, b_{d_c-1} m_{d_c-1}) = \left(\prod_{i=1}^{d_c-1} b_i \right) \Psi_c^{(\ell)}(m_1, \cdots, m_{d_c-1}) \,.$$

- **Variable node symmetry:** If the signs of all messages into a variable node are flipped, then the sign of its output gets flipped:

$$\Psi_v^{(\ell)}(-m_0, -m_1, \cdots, -m_{d_v-1}) = -\Psi_v^{(\ell)}(m_0, m_1, \cdots, m_{d_c-1}) \,.$$

When the above symmetry assumptions are fulfilled and the channel is output-symmetric, the decoding error probability is independent of the actual codeword transmitted. Indeed, it is not hard (see, for instance [23, Lemma 1]) to show that when a codeword $(x_1, \ldots, x_n)$ is transmitted and $(y_1, \ldots, y_n)$ is received where $y_i = x_i z_i$, the messages to and from the variable node v_i are equal to x_i times the corresponding message when the all-ones codeword is transmitted and $(z_1, \ldots, z_n)$ is received. Therefore, the entire behavior of the decoder can be predicted from its behavior assuming transmission of the all-ones codeword (recall that we are using $\{1, -1\}$ notation for the binary alphabet). So, for the analysis, we will assume that the all-ones codeword was transmitted.

5 Regular LDPC codes and simple iterative decoders

We will begin with regular LDPC codes and a theoretical analysis of simple message-passing algorithms for decoding them.

5.1 Gallager's program

The story of LDPC codes and iterative decoding begins in Gallager's remarkable Ph.D. thesis completed in 1960, and later published in 1963 [8]. Gallager analyzed the behavior of a code picked randomly from the ensemble of (d_v, d_c)-regular LDPC codes of a large block length. He proved that with high probability, as d_v and d_c increase, the rate vs. minimum distance trade-off of the code approaches the Gilbert-Varshamov bound. Gallager also analyzed the error probability of maximum likelihood (ML) decoding of random (d_c, d_c)-regular LDPC codes, and showed that LDPC codes are at least as good on the BSC as the optimum code a somewhat higher rate (refer to [8] for formal details concerning this statement). This demonstrated the promise of LDPC codes independently of their decoding algorithms (since ML decoding is the optimal decoding algorithm in terms of minimizing error probability).

To complement this statement, Gallager also proved a "negative" result showing that for each finite d_c, there is a finite gap to capacity on the BSC when using regular LDPC codes with check node degrees d_c More precisely, he proved that the largest rate that can be achieved for BSC_p with error probability going to zero is at most $1 - \frac{H(p)}{H(p_{d_c})}$ where $p_{d_c} = \frac{1+(1-2p)^{dc}}{2}$. This claim holds even for irregular LDPC codes with d_c interpreted as the maximum check node degree. This shows that the maximum check node degree needs to grow with the gap ε between the rate of the code and capacity of the BSC.

Since only exponential time solutions to the ML decoding problem are known, Gallager also developed simple, iterative decoding algorithms for LDPC codes. These form the precursor to the modern day message-passing algorithms. More generally, he laid down the foundations of the following program for determining the threshold channel parameter below which a suitable LDPC code can be used in conjunction with a given iterative decoder for reliable information transmission.

Code construction: Construct a family of (d_v, d_c)-regular factor graphs with n variable nodes (for increasing n) with girth greater than $4\ell(n) = \Omega(\log n)$. An explicit construction of such graphs was also given by Gallager [8, Appendix C].

Analysis of Decoder: Determine the average fraction of incorrect[7] messages passed at the i'th iteration of decoding for $i \leq \ell = \ell(n)$ (assuming there are no cycles of length at most 4ℓ). This fraction is usually expressed by a system of recursive equations that depend on d_v, d_c and the channel parameter (such as crossover probability, in case of the BSC).

Threshold computation: Using the above equations, compute (analytically or numerically) the threshold channel parameter below which the expected fraction of incorrect messages approaches zero as the number of iterations increases. Conclude that the chosen decoder when applied to this family of codes with $\ell(n)$ decoding rounds leads to bit-error probability approaching zero as long as the channel parameter is below the threshold.

The recent research on (irregular) LDPC codes shares the same essential features of the above program. The key difference is that the requirement of an explicit code description in Step 1 is relaxed. This is because for irregular graphs with specific requirements on degree distribution, explicit constructions of large

[7] A message is incorrect if the bit value it estimates is wrong. For transmission of the all-ones codeword, this means the message has a non-positive value.

girth graphs seem very hard. Instead, a factor graph chosen randomly from a suitable ensemble is used. This raises issues such as the concentration of the performance of a random code around the average behavior of the ensemble. It also calls for justification of the large girth assumption in the decoding. We will return to these aspects when we begin our discussion of irregular LDPC codes in Section 6.

We should point out that Gallager himself used random regular LDPC codes for his experiments with iterative decoders for various channels such as the BSC, the BIAWGN, and the Rayleigh fading channel. However, if we so desire, for the analytic results, even explicit constructions are possible. In the rest of this section, we assume an explicit large girth factor graph is used, and focus on the analysis of some simple and natural iterative decoders. Thus the only randomness involved is the one realizing the channel noise.

5.2 Decoding on the binary erasure channel

Although Gallager did not explicitly study the BEC, his methods certainly apply to it, and we begin by studying the BEC. For the BEC, there is essentially a unique choice for a non-trivial message-passing decoding algorithm. In a variable-to-check message round, a variable whose bit value is known (either from the channel output or from a check node in a previous round) passes along its value to the neighboring check nodes, and a variable whose bit value is not yet determined passes a symbol (say 0) signifying erasure. In the check-to-variable message round, a check node c passes to a neighbor v an erasure if it receives an erasure from at least one neighbor besides v, and otherwise passes the bit value b to v where b is the parity of the bits received from neighbors other than v. Formally, the message maps are given as follows:

$$\Psi_v^{(\ell)}(r, m_1, \ldots, m_{d_v-1}) = \begin{cases} b & \text{if at least one of } r, m_1, \ldots, m_{d_v-1} \text{ equals } b \in \{1, -1\} \\ 0 & \text{if } r = m_1 = \cdots = m_{d_v-1} = 0 \end{cases}$$

(Note that the map is well-defined since the inputs to a variable node will never give conflicting ±1 votes on its value.)

$$\Psi_c^{(\ell)}(m_1, \ldots, m_{d_c-1}) = \prod_{i=1}^{d_c-1} m_i$$

We note that an implementation of the decoder is possible that uses each edge of the factor for message passing exactly once. Indeed, once a variable node's value is known, the bit value is communicated to its neighboring check nodes, and this node (and edges incident on it) are removed from the graph. Each check node maintains the parity of the values received from its neighboring variables so far, and updates this after each round of variable messages (note that it receives each variable node's value exactly once). When a check node has degree exactly one (i.e., values of all but one of its variable node neighbors are now known), it communicates the parity value it has stored to its remaining neighbor, and both the check node and the remaining edge incident on it are deleted. This version of the iterative decoder has been dubbed the *Peeling Decoder*. The running time of the Peeling Decoder is essentially the number of edges in the factor graph, and hence it performs about d_v operations per codeword bit.

Let us analyze this decoding algorithm for ℓ iterations, where ℓ is a constant (chosen large enough to achieve the desired bit-error probability). We will assume that the factor graph does not have any cycle of length at most 4ℓ (which is certainly true if it has $\Omega(\log n)$ girth).

The following is crucial to our analysis.

Lemma 1. *For each node, the random variables corresponding to the messages received by it in the i'th iteration are all independent, for $i \leq \ell$.*

Let us justify why the above is the case. For this, we crucially use the fact that the message sent along an edge, say from v to c, does not depend on the message that v receives from c. Therefore, the information received at a check node c (the situation for variable nodes is identical) from its neighbors in the i'th iteration is determined by by a computation graph rooted at c, with its d_c variable node neighbors as its children, the $d_v - 1$ neighbors besides c of each these variable nodes as their children, the $d_c - 1$ other neighbors of these check nodes as their children, and so on. Since the girth of the graph is greater than 4ℓ, the computation graph is in fact a tree. Therefore, the information received by c from its neighbors in the i'th iteration are all independent.

Take an arbitrary edge (v, c) between variable node v and check node c. Let us compute the probability p_i that the message from v to c in the i'th iteration is an erasure (using induction and the argument below, one can justify the claim that this probability, which is taken over the channel noise, will be independent of the edge and only depend on the iteration number, as long as $i \leq \ell$). For $i = 0$, $p_0 = \alpha$, the probability that the bit value for v was erased by the BEC_α. In the $(i + 1)$'st iteration, v passes an erasure to c iff it was originally erased by the channel, and it received an erasure from each of its $d_v - 1$ neighbors other than c. Each of these neighboring check nodes c' in turn sends an erasure to v iff at least one neighbor of c' other than v sent an erasure to c' during iteration i — due to the independence of the involved messages, this event occurs for node c' with probability $(1 - (1 - p_i)^{d_c-1})$. Again, because the messages from various check nodes to v in the $(i + 1)$'st round are independent, we have

$$p_{i+1} = \alpha \cdot (1 - (1 - p_i)^{d_c-1})^{d_v-1} . \tag{1}$$

By linearity of expectation, p_i is the expected fraction of variable-to-check messages sent in the i'th iteration that are erasures. We would like to show that $\lim_{\ell\to\infty} p_\ell = 0$, so that the bit-error probability of the decoding vanishes as the number of iterations grows. The largest erasure probability α for which this happens is given by the following lemma.

Lemma 2. *The threshold erasure probability $\alpha^{\mathsf{MP}}(d_v, d_c)$ for the BEC below which the message-passing algorithm results in vanishing bit-erasure probability is given by*

$$\alpha^{\mathsf{MP}}(d_v, d_c) = \min_{x\in[0,1]} \frac{x}{(1-(1-x)^{d_c-1})^{d_v-1}} . \tag{2}$$

Proof. By definition, $\alpha^{\mathsf{MP}}(d_v, d_c) = \sup\{\alpha \in [0,1] : \lim_{i\to\infty} p_i = 0\}$ where p_i is as defined recursively in (1). Define the functions $g(x) = \frac{x}{(1-(1-x)^{d_c-1})^{d_v-1}}$, and $f(\alpha, x) = \alpha(1-(1-x)^{d_c-1})^{d_v-1}$. Also let $\alpha^* = \min_{x\in[0,1]} g(x)$. We wish to prove that $\alpha^{\mathsf{MP}}(d_v, d_c) = \alpha^*$.

If $\alpha < \alpha^*$, then for every $x \in [0,1]$, $f(\alpha, x) = \frac{\alpha x}{g(x)} \le \frac{\alpha^* x}{g(x)} \le x$, and in fact $f(\alpha, x) < x$ for $x \in (0,1]$. Hence it follows that $p_{i+1} = f(\alpha, p_i) \le p_i$ and since $0 \le f(\alpha, x) \le \alpha$ for all $x \in [0,1]$, the probability converges to a value $p_\infty \in [0,\alpha]$. Since f is continuous, we have $p_\infty = f(\alpha, p_\infty)$, which implies $p_\infty = 0$ (since $f(\alpha, x) < x$ for $x > 0$). This shows that $\alpha^{\mathsf{MP}}(d_v, d_c) \ge \alpha^*$.

Conversely, if $\alpha > \alpha^*$, then let $x_0 \in [0,1]$ be such that $\alpha > g(x_0)$. Then $\alpha \ge f(\alpha, x_0) = \frac{\alpha x_0}{g(x_0)} > x_0$, and of course $f(\alpha,\alpha) \le \alpha$. Since $f(\alpha, x)$ is a continuous function of x, we must have $f(\alpha, x^*) = x^*$ for some $x^* \in (x_0, \alpha]$. For the recursion (1) with a fixed value of α, it is easy to see by induction that if $p_0 \ge p'_0$, then $p_i \ge p'_i$ for all $i \ge 1$. If $p'_0 = x^*$, then we have $p'_i = x^*$ for all i. Therefore, when $p_0 = \alpha \ge x^*$, we have $p_i \ge x^*$ for all i as well. In other words, the error probability stays bounded below by x^* irrespective of the number of iterations. This proves that $\alpha^{\mathsf{MP}}(d_v, d_c) \le \alpha^*$.

Together, we have exactly determined the threshold to be $\alpha^* = \min_{x\in[0,1]} g(x)$. ■

Remark 3. *Using standard calculus, we can determine $\alpha^{\mathsf{MP}}(d_v, d_c)$ to be $\frac{1-\gamma}{(1-\gamma^{d_c-1})^{d_v-1}}$ where γ is the unique positive root of the polynomial $p(x) = ((d_v-1)(d_c-1)-1)x^{d_c-2} - \sum_{i=0}^{d_c-3} x^i$. Note that when $d_v = 2$, $p(1) = 0$, so the threshold equals 0. Thus we must pick $d_v \ge 3$, and hence $d_c \ge 4$ (to have positive rate). For the choice $d_v = 3$ and $d_c = 4$, $p(x)$ is a quadratic and we can analytically compute $\alpha^{\mathsf{MP}}(3,4) \approx 0.6474$; note that capacity for this rate equals $3/4 = 0.75$. (The best threshold one can hope for equals d_v/d_c since the rate is at least $1 - d_v/d_c$.) Closed form analytic expressions for some other small values of (d_v, d_c) are given in [2]: for example, $\alpha^{\mathsf{MP}}(3,5) \approx 0.5406$ (compare to capacity of 0.6) and $\alpha^{\mathsf{MP}}(3,6) \approx 0.4294$ (compare to capacity of 0.5).*

Theorem 4. *For integers $3 \leq d_v < d_c$, there exists an explicit family of binary linear codes of rate at least $1 - \frac{d_v}{d_c}$ that can be reliably decoded in linear time on BEC_α provided $\alpha < \alpha^{\mathsf{MP}}(d_v, d_c)$.*[8]

5.3 Decoding on the BSC

The relatively clean analysis of regular LDPC codes on the BEC is surely encouraging. As mentioned earlier, Gallager in fact did not consider the BEC in his work. We now discuss one of his decoding algorithms for the BSC, that has been dubbed Gallager's Algorithm A, and some simple extensions of it.

5.3.1 Gallager's Algorithm A

The message alphabet of Algorithm A will equal $\{1, -1\}$, so the nodes simply pass guesses on codeword bits. The message maps are time invariant and do not depend on the iteration number, so we will omit the superscript indicating the iteration number in describing the message maps. The check nodes send a message to a variable node indicating the parity of the *other* neighboring variables, or formally:

$$\Psi_c(m_1, \ldots, m_{d_c-1}) = \prod_{i=1}^{d_c-1} m_i \ .$$

The variable nodes send to a neighboring check node their original received value unless the incoming messages from the *other* check nodes unanimously indicate otherwise, in which case it sends the negative of the received value. Formally,

$$\Psi_v(r, m_1, \ldots, m_{d_v-1}) = \begin{cases} -r & \text{if } m_1 = \cdots = m_{d_v-1} = -r \\ r & \text{otherwise .} \end{cases}$$

As in the case of BEC, we will track the expected fraction of variable-to-check node messages that are erroneous in the i'th iteration. Since we assume the all-ones codeword was transmitted, this is simply the expected fraction of messages that equal -1. Let p_i be the probability (over the channel noise) that a particular variable-to-check node message in iteration i equals -1 (as in the case of the BEC, this is independent of the actual edge for $i \leq \ell$). Note that we have $p_0 = p$, the crossover probability of the BSC.

[8] Our analysis showed that the bit-error probability can be made below any desired $\varepsilon > 0$ by picking the number of iterations to be a large enough constant. A more careful analysis using $\ell(n) = \Omega(\log n)$ iterations shows that bit-error probability is at most $\exp(-n^\beta)$ for some constant $\beta = \beta(d_v, d_c)$. By a union bound, the entire codeword is thus correctly recovered with high probability.

It is a routine calculation using the independence of the incoming messages to prove the following recursive equation [8, Sec. 4.3], [23, Sec III]:

$$p_{i+1} = p_0 - p_0 \left(\frac{1 + (1 - 2p_i)^{d_c - 1}}{2} \right)^{d_v - 1} + (1 - p_0) \left(\frac{1 - (1 - 2p_i)^{d_c - 1}}{2} \right)^{d_v - 1} \tag{3}$$

For a fixed value of p_0, p_{i+1} is a increasing function of p_i, and for a fixed value of p_i, p_{i+1} is an increasing function of p_0. Therefore, by induction p_i is an increasing function of p_0. Define the threshold value of this algorithm "A" as $p^A(d_v, d_c) = \sup\{p_0 \in [0, 1] : \lim_{\ell \to \infty} p_\ell = 0\}$. By the above argument, if the crossover probability $p < p^A(d_v, d_c)$, then the expected fraction of erroneous messages in the ℓ'th iteration approaches 0 as $\ell \to \infty$.

Regardless of the exact quantitative value, we want to point out that when $d_v \geq 3$, the threshold is positive. Indeed, for $d_v > 2$, for small enough $p_0 > 0$, one can see that $p_{i+1} < p_i$ for $0 < p_i \leq p_0$ and $p_{i+1} = p_i$ for $p_i = 0$, which means that $\lim_{i \to \infty} p_i = 0$.

Exact analytic expressions for the threshold have been computed for some special cases [2]. This is based on the characterization of $p^A(d_v, d_c)$ as the supremum of all $p_0 > 0$ for which

$$x = p_0 - p_0 \left(\frac{1 + (1 - 2x)^{d_c - 1}}{2} \right)^{d_v - 1} + (1 - p_0) \left(\frac{1 - (1 - 2x)^{d_c - 1}}{2} \right)^{d_v - 1}$$

does not have a strictly positive solution x with $x \leq p_0$. Below are some example values of the threshold (up to the stated precision). Note that the rate of the code is $1 - d_v/d_c$ and the Shannon limit is $H^{-1}(d_v/d_c)$ (where $H^{-1}(y)$ for $0 \leq y \leq 1$ is defined as the unique value of $x \in [0, 1/2]$ such that $H(x) = y$).

d_v	d_c	$p^A(d_v, d_c)$	Capacity
3	6	0.0395	0.11
4	8	1/21	0.11
5	10	1/36	0.11
4	6	1/15	0.174
3	4	0.106	0.215
3	5	0.0612	0.146

5.3.2 Gallager's Algorithm B

Gallager proposed an extension to the above algorithm, which is now called Gallager's Algorithm B, in which a variable node decides to flip its value in an outgoing message when at least b of the incoming messages suggest that it ought to flip its value. In Algorithm A, we have $b = d_v - 1$. The threshold b can also depend on the iteration number, and we will denote by b_i this value during the i'th iteration. Formally, the variable message map in the i'th iteration is given by

$$\Psi_v^{(i)}(r, m_1, \ldots, m_{d_v-1}) = \begin{cases} -r & \text{if } |\{j : m_j = -r\}| \geq b_i \\ r & \text{otherwise .} \end{cases}$$

The check node message maps remain the same. The threshold should be greater than $(d_v - 1)/2$ since intuitively one should flip only when more check nodes suggest a flip than those that suggest the received value. So when $d_v = 3$, the above algorithm reduces to Algorithm A.

Defining the probability of an incorrect variable-to-check node message in the i'th iteration to be $\tilde{p}_i$, one can show the recurrence [8, Sec. 4.3]:

$$\begin{aligned} \tilde{p}_{i+1} = \tilde{p}_0 - \tilde{p}_0 \sum_{j=b_{i+1}}^{d_v-1} \binom{d_v-1}{j} \left(\frac{1 + (1-2\tilde{p}_i)^{d_c-1}}{2}\right)^j \left(\frac{1 - (1-2\tilde{p}_i)^{d_c-1}}{2}\right)^{d_v-1-j} \\ + (1-\tilde{p}_0) \sum_{j=b_{i+1}}^{d_v-1} \binom{d_v-1}{j} \left(\frac{1 + (1-2\tilde{p}_i)^{d_c-1}}{2}\right)^{d_v-1-j} \left(\frac{1 - (1-2\tilde{p}_i)^{d_c-1}}{2}\right)^j \end{aligned}$$

The cut-off value b_{i+1} can then be chosen to minimize this value. The solution to this minimization is the smallest integer b_{i+1} for which

$$\frac{1-\tilde{p}_0}{\tilde{p}_0} \leq \left(\frac{1 + (1-2\tilde{p}_i)^{d_c-1}}{1 - (1-2\tilde{p}_i)^{d_c-1}}\right)^{2b_{i+1}-d_v+1} .$$

By the above expression, we see that as $\tilde{p}_i$ decreases, b_{i+1} never increases. And, as $\tilde{p}_i$ is sufficiently small, b_{i+1} takes the value $d_v/2$ for even d_v and $(d_v + 1)/2$ for odd d_v. Therefore, a variable node flips its value when a majority of the $d_v - 1$ incoming messages suggest that the received value was an error. We note that this majority criterion for flipping a variable node's bit value was also used in decoding of expander codes [29].

Similar to the analysis of Algorithm A, using the above recurrence, one can show that when $d_v \geq 3$, for sufficiently small $p_0 > 0$, we have $p_{i+1} < p_i$ when $0 < p_i \leq p_0$, and of course when $p_i = 0$, we have $p_{i+1} = 0$. Therefore, when $d_v \geq 3$, for small enough $p_0 > 0$, we have $\lim_{i\to\infty} p_i = 0$ and thus a positive threshold.

The values of the threshold of this algorithm for small pairs (d_v, d_c) appear in [23]. For the pairs $(4, 8)$, $(4, 6)$ and $(5, 10)$ the thresholds are about 0.051, 0.074, and 0.041 respectively. For comparison, for these pairs Algorithm A achieved a threshold of about 0.047, 0.066, and 0.027 respectively.

5.3.3 Using Erasures in the Decoder

In both the above algorithms, each message made up its mind on whether to guess 1 or -1 for a bit. But it may be judicious to sometimes abstain from guessing, i.e., to send an "erasure" message (with value 0), if there is no good reason to guess one way or the other. For example, this may be the appropriate course of action if a variable node receives one-half 1's and one-half -1's in the incoming check node messages. This motivates an algorithm with message alphabet $\{1, 0, -1\}$ and the following message maps (in iteration ℓ):

$$\Psi_v^{(\ell)}(r, m_1, m_2, \ldots, m_{d_v-1}) = \mathsf{sgn}\left(w^{(\ell)} r + \sum_{j=1}^{d_v-1} m_j \right)$$

and

$$\Psi_c^{(\ell)}(m_1, m_2, \ldots, m_{d_c-1}) = \prod_{j=1}^{d_c-1} m_j \ .$$

The weight $w^{(\ell)}$ dictates the relative importance given to the received value compared to the suggestions by the check nodes in the ℓ'th iteration. These weights add another dimension of design choices that one can optimize.

Exact expressions for the probabilities $p_i^{(-1)}$ and $p_i^{(0)}$) that a variable-to-check message is an error (equals -1) and an erasure (equals 0) respectively in the i'th iteration can be written down [23]. These can be used to pick appropriate weights $w^{(i)}$. For the $(3, 6)$-regular code, $w^{(1)} = 2$ and $w^{(i)} = 1$ for $i \geq 2$ is reported as the optimum choice in [23], and using this choice the resulting algorithm has a threshold of about 0.07, which is a good improvement over the 0.04 achieved by Algorithm A. More impressively, this is close to the threshold of 0.084 achieves by the "optimal" belief propagation decoder. A heuristic to pick the weights $w^{(i)}$ is suggested in [23] and the threshold of the resulting algorithm is computed for small values of (d_v, d_c).

5.4 Decoding on BIAWGN

We now briefly turn to the BIAWGN channel. We discussed the most obvious quantization of the channel output which converts the channel to a BSC with crossover probability $Q(1/\sigma)$. There is a natural way to incorporate erasures into the quantization. We pick a threshold τ around zero, and quantize the AWGN channel output r into -1, 0 (which corresponds to erasure), or 1 depending on whether $r \leq -\tau$, $-\tau < r < \tau$, or $r \geq \tau$, respectively. We can then run exactly the above message-passing algorithm (the one using erasures). More generally, we can pick a separate threshold τ_i for each iteration i — the choice of τ_i and

$w^{(i)}$ can be optimized using some heuristic criteria. Using this approach, a threshold of $\sigma^* = 0.743$ is reported for communication using a $(3,6)$-regular LDPC code on the BIAWGN channel. This corresponds to a raw bit-error probability of $Q(1/\sigma^*) = 0.089$, which is almost 2% greater than the threshold crossover probability of about 0.07 achieved on the BSC. So even with a ternary message alphabet, providing soft information (instead of quantized hard bit decisions) at the input to the decoder can be lead to a good performance gain. The belief propagation algorithm we discuss next uses a much large message alphabet and yields further substantial improvements for the BIAWGN.

5.5 The belief propagation decoder

So far we have discussed decoders with quantized, discrete messages taking on very few values. Naturally, we can expect more powerful decoders if more detailed information, such as real values quantifying the likelihood of a bit being ± 1, are passed in each iteration. We now describe the "belief propagation" (BP) decoder which is an instance of such a decoder (using a continuous message alphabet). We follow the description in [23, Sec. III-B]. In belief propagation, the messages sent along an edge e represent the posterior conditional distribution on the bit associated with the variable node incident on e. This distribution corresponds to a pair of nonnegative reals p_1, p_{-1} satisfying $p_1 + p_{-1} = 1$. This pair can be encoded as a single real number (including $\pm\infty$) using the log-likelihood ratio $\log \frac{p_1}{p_{-1}}$, and the messages used by the BP decoder will follow this representation.

Each node acts under the assumption that each message communicated to it in a given round is a conditional distribution on the associated bit, and further each such message is conditionally independent of the others. Upon receiving the messages, a node transmits to each neighbor the conditional distribution of the bit conditioned on all information *except* the information from that neighbor (i.e., only extrinsic information is used in computing a message). If the graph has large enough girth compared to the number of iterations, this assumption is indeed met, and the messages at each iteration reflect the true log-likelihood ratio given the observed values in the tree neighborhood of appropriate depth.

If $l_1, l_2, \ldots, l_k$ are the likelihood ratios of the conditional distribution of a bit conditioned on independent random variables, then the likelihood ratio of the bit value conditioned on all of the random variables equals $\prod_{i=1}^{k} l_i$. Therefore, log-likelihoods of independent messages add up, and this leads to the variable message map (which is independent of the iteration number):

$$\Psi_v(m_0, m_1, \ldots, m_{d_v-1}) = \sum_{i=0}^{d_v-1} m_i$$

where m_0 is the log-likelihood ratio of the bit based on the received value (eg., for the BSC_p, $m_0 = r \log \frac{1-p}{p}$ where $r \in \{1, -1\}$ is the received value).

The performance of the decoder is analyzed by tracking the evolution of the probability density of the log-likelihood ratios (hence the name "density evolution" for this style of analysis). By the above, given densities $P_0, P_1, \ldots, P_{d_v-1}$ on the real quantities $m_0, m_1, \ldots, m_{d_v-1}$, the density of $\Psi_v(m_0, m_1, \ldots, m_{d_v-1})$ is the convolution $P_0 \otimes P_1 \otimes \cdots \otimes P_{d_v-1}$ over the reals of those densities. In the computation, one has $P_1 = P_2 = \cdots = P_{d_v-1}$ and the densities will be quantized, and the convolution can be efficiently computed using the FFT.

Let us now turn to the situation for check nodes. Given bits b_i, $1 \le i \le k$, with independent probability distributions (p_1^i, p_{-1}^i), what is the distribution (p_1, p_{-1}) of the bit $b = \prod_{i=1}^k b_i$? We have the expectation

$$E[b] = E[\prod_i b_i] = \prod_i E[b_i] = \prod_i (p_1^i - p_{-1}^i) .$$

Therefore we have $p_1 - p_{-1} = \prod_{i=1}^k (p_1^i - p_{-1}^i)$. Now if m is the log-likelihood ratio $\log \frac{p_1}{p_{-1}}$, then $p_1 - p_{-1} = \frac{e^m - 1}{e^m + 1} = \tanh(m/2)$. Conversely, if $p_1 - p_{-1} = q$, then $\log \frac{p_1}{p_{-1}} = \log \frac{1+q}{1-q}$. These calculations lead to the following check node map for the log-likelihood ratio:

$$\Psi_c(m_1, m_2, \ldots, m_{d_c-1}) = \log \left(\frac{1 + \prod_{i=1}^{d_c-1} \tanh(m_i/2)}{1 - \prod_{i=1}^{d_c-1} \tanh(m_i/2)} \right) .$$

It seems complicated to track the density of $\Psi_c(m_1, m_2, \ldots, m_{d_c-1})$ based on those of the m_i's. However, as shown in [23], this can be also be realized via a Fourier transform, albeit with a slight change in representation of the conditional probabilities (p_1, p_{-1}). We skip the details and instead point the reader to [23, Sec. III-B].

Using these ideas, we have an effective algorithm to recursively compute, to any desired degree of accuracy, the probability density $P^{(\ell)}$ of the log-likelihood ratio of the variable-to-check node messages in the ℓ-th iteration, starting with an explicit description of the initial density $P^{(0)}$. The initial density is simply the density of the log-likelihood ratio of the received value, assuming transmission of the all-ones codeword; for example, for BSC_p, the initial density $P^{(0)}$ is given by

$$P^{(0)}(x) = p\delta\left(x - \log \frac{p}{1-p}\right) + (1-p)\delta\left(x - \log \frac{1-p}{p}\right) ,$$

where $\delta(x)$ is the Dirac delta function.

The threshold crossover probability for the BSC and the threshold variance for the BIAWGN under belief propagation decoding for various small values of

(d_v, d_c) are computed by this method and reported in [23]. For the $(3, 6)$ LDPC code, these thresholds are respectively $p^* = 0.084$ (compare with Shannon limit of 0.11) and $\sigma^* = 0.88$ (compare with Shannon limit of 0.9787).

The above numerical procedure for tracking the evolution of densities for belief propagation and computing the associated threshold to any desired degree of accuracy has since been applied with great success. In [22], the authors apply this method to irregular LDPC codes with optimized structure and achieve a threshold of $\sigma^* = 0.9718$ with rate $1/2$ for the BIAWGN, which is a mere 0.06 dB way from the Shannon capacity limit.[9]

6 Irregular LDPC codes

Interest in LDPC codes surged following the seminal paper [16] that initiated the study of irregular LDPC codes, and proved their potential by achieving the capacity on the BEC. Soon, it was realized that the benefits of irregular LDPC codes extend to more powerful channels, and this led to a flurry of activity. In this section, we describe some of the key elements of the analytic approach used to to study message-passing decoding algorithms for irregular LDPC codes.

6.1 Intuitive benefits of irregularity

We begin with some intuition on why one might expect improved performance by using irregular graphs. In terms of iterative decoding, from the variable node perspective, it seems better to have high degree, since the more information it gets from check nodes, the more accurately it can guess its correct value. On the other hand, from the check node perspective, the lower its degree, the more valuable the information it can transmit back to its neighbors. (The XOR of several mildly unpredictable bits has a much larger unpredictability.) But in order to have good rate, there should be far fewer check nodes than variable nodes, and therefore meeting the above competing requirements is challenging. Irregular graphs provide significantly more flexibility in balancing the above incompatible degree requirements. It seems reasonable to believe that a wide spread of degrees for variable nodes could be useful. This is because one might expect that variable nodes with high degree will converge to their correct value quickly. They can then provide good information to the neighboring check nodes, which in turn provide better information to lower degree variable nodes, and so on leading to a cascaded wave effect.

[9] The threshold signal-to-noise ratio $1/(\sigma^*)^2 = 0.2487$ dB, and the Shannon limit for rate $1/2$ is 0.187 dB.

The big challenge is to leap from this intuition to the design of appropriate irregular graphs where this phenomenon provably occurs, and to provide analytic bounds on the performance of natural iterative decoders on such irregular graphs.

Compared to the regular case, there are additional technical issues revolving around how irregular graphs are parameterized, how they are constructed (sampled), and how one deals with the lack of explicit large-girth constructions. We discuss these issues in the next two subsections.

6.2 The underlying ensembles

We now describe how irregular LDPC codes can be parameterized and constructed (or rather sampled). Assume we have an LDPC code with n variable nodes with Λ_i variable nodes of degree i and P_i check nodes of degree i. We have $\sum_i \Lambda_i = n$, and $\sum_i i\Lambda_i = \sum_i iP_i$ as both these equal the number of edges in the graph. Also $\sum_i P_i = n(1-r)$ where r is the designed rate of the code. It is convenient to capture this information in the compact polynomial notation:

$$\Lambda(x) = \sum_{i=2}^{d_v^{\max}} \Lambda_i x^i \ , \qquad P(x) = \sum_{i=1}^{d_c^{\max}} P_i x^i \ .$$

We call the polynomials Λ and P the variable and check degree distributions from a node perspective. Note that $\Lambda(1)$ is the number of variable nodes, $P(1)$ the number of check nodes, and $\Lambda'(1) = P'(1)$ the number of edges.

Given such a degree distribution pair (Λ, P), let $\mathsf{LDPC}(\Lambda, P)$ denote the "standard" ensemble of bipartite (multi)graphs with $\Lambda(1)$ variable nodes and $P(1)$ check nodes, with Λ_i variable nodes and P_i check nodes of degree i. This ensemble is defined by taking $\Lambda'(1) = P'(1)$ "sockets" on each side, allocating i sockets to a node of degree i in some arbitrary manner, and then picking a random matching between the sockets.

To each member of $\mathsf{LDPC}(\Lambda, P)$, we associate the code of which it is the factor graph. A slight technicality: since we are dealing with multigraphs, in the parity check matrix, we place a non-zero entry at row i and column j iff the ith check node is connected to the jth variable node an *odd* number of times. Therefore, we can think of the above as an ensemble of codes, and by abuse of notation also refer to it as $\mathsf{LDPC}(\Lambda, P)$. (Note that the graphs have a uniform probability distribution, but the induced codes need not.) In the sequel, our LDPC codes will be obtained by drawing a random element from the ensemble $\mathsf{LDPC}(\Lambda, P)$.

To construct a family of codes, one can imagine using a normalized degree distribution giving the *fraction* of nodes of a certain degree, and then considering an increasing number of nodes. For purposes of analysis, it ends up being convenient to use normalized degree distributions from the *edge* perspective. Let λ_i and

ρ_i denote the fraction of *edges* incident to variable nodes and check nodes of degree i respectively. That is, λ_i (resp. ρ_i) is the probability that a randomly chosen edge is connected to a variable (resp. check) node of degree i. These distributions can be compactly written in terms of the power series defined below:

$$\lambda(x) = \sum_i \lambda_i x^{i-1} , \qquad \rho(x) = \sum_i \rho_i x^{i-1} .$$

It is easily seen that $\lambda(x) = \frac{\Lambda'(x)}{\Lambda'(1)}$ and $\rho(x) = \frac{P'(x)}{P'(1)}$. If M is the total number of edges, then the number of variable nodes of degree i equals $M\lambda_i/i$, and thus the total number of variable nodes is $M\sum_i \lambda_i/i$. It follows that that the average variable node degree equals $\frac{1}{\sum_i \lambda_i/i} = \frac{1}{\int_0^1 \lambda(z)dz}$. Likewise, the average check node degree equals $\frac{1}{\int_0^1 \rho(z)dz}$. It follows that the designed rate can be expressed in terms of λ, ρ as

$$r = r(\lambda,\rho) = 1 - \frac{\int_0^1 \rho(z)dz}{\int_0^1 \lambda(z)dz} . \tag{4}$$

We also have the inverse relationships

$$\frac{\Lambda(x)}{n} = \frac{\int_0^x \lambda(z)dz}{\int_0^1 \lambda(z)dz} , \quad \frac{P(x)}{n(1-r)} = \frac{\int_0^x \rho(z)dz}{\int_0^1 \rho(z)dz} . \tag{5}$$

Therefore, (Λ, P) and (n, λ, ρ) carry the same information (in the sense we can obtain each from the other). For the asymptotic analysis we use (n, λ, ρ) to refer to the LDPC code ensemble. There is a slight technicality that for some n, the (Λ, P) corresponding to (n, λ, ρ) may not be integral. In this case, rounding the individual node distributions to the closest integer has negligible effect on the asymptotic performance of decoder or the rate, and so this annoyance may be safely ignored.

The degree distributions λ, ρ play a prominent role in the line of work, and the performance of the decoder is analyzed and quantified in terms of these.

6.3 Concentration around average performance

Given a degree distribution pair (λ, ρ) and a block length n, the goal is to mimic Gallager's program (outlined in Section 5.1), using a factor graph with degree distribution (λ, ρ) in place of a (d_v, d_c)-regular factor graph. However, the task of constructing explicit large girth graphs obeying precise irregular degree distributions seems extremely difficult. Therefore, a key difference is to give up on explicitness, and rather sample an element from the ensemble $\mathsf{LDPC}(n, \lambda, \rho)$, which can be done easily as mentioned above.

It is not very difficult to show that a random code drawn from the ensemble will have the needed girth (and thus be tree-like in a local neighborhood of every edge/vertex) with high probability; see for instance [23, Appendix A]. A more delicate issue is the following: For the irregular case the neighborhood trees out of different nodes have a variety of different possible structures, and thus analyzing the behavior of the decoder on a specific factor graph (after it has been sampled, even conditioning on it having large girth) seems hopeless. What *is* feasible, however, is to analyze the *average* behavior of the decoder (such as the expected fraction, say $P_n^{(\lambda,\rho)}(\ell)$, of erroneous variable-to-check messages in the ℓ'th iteration) taken over all instances of the code drawn from the ensemble $\mathsf{LDPC}(n, \lambda, \rho)$ and the realization of the channel noise. It can be shown that, as $n \to \infty$, $P_n^{(\lambda,\rho)}(\ell)$ converges to a certain quantity $P_{\mathcal{T}}^{(\lambda,\rho)}(\ell)$, which is defined as the probability (taken over both choice of the graph and the noise) that an incorrect message is sent in the ℓ'th iteration along an edge (v, c) assuming that the depth 2ℓ neighborhood out of v is a tree.

In order to define the probability $P_{\mathcal{T}}^{(\lambda,\rho)}(\ell)$ more precisely, one uses a "tree ensemble" $\mathcal{T}_\ell(\lambda,\rho)$ defined inductively as follows. $\mathcal{T}_0(\lambda,\rho)$ consists of the trivial tree consisting of just a root variable node. For $\ell \geq 1$, to sample from $\mathcal{T}_\ell(\lambda,\rho)$, first sample an element from $\mathcal{T}_{\ell-1}(\lambda,\rho)$. Next for each variable leaf node (independently), with probability λ_{i+1} attach i check node children. Finally, for each of the new check leaf nodes, independently attach i variable node children with probability ρ_{i+1}. The quantity $P_{\mathcal{T}}^{(\lambda,\rho)}(\ell)$ is then formally defined as the probability that the outgoing message from the root node of a sample T from $\mathcal{T}_\ell(\lambda,\rho)$ is incorrect, assuming the variable nodes are initially labeled with 1 and then the channel noise acts on them independently (the probability is thus both over the channel noise and the choice of the sample T from $\mathcal{T}_\ell(\lambda,\rho)$).

The convergence of $P_n^{(\lambda,\rho)}(\ell)$ to $P_{\mathcal{T}}^{(\lambda,\rho)}(\ell)$ is a simple consequence of the fact that, for a random choice of the factor graph from $\mathsf{LDPC}(n, \lambda, \rho)$, the depth 2ℓ neighborhood of an edge is tree-like with probability tending to 1 as n gets larger (for more details, see [23, Thm. 2]).

The quantity $P_{\mathcal{T}}^{(\lambda,\rho)}(\ell)$ for the case of trees is easily computed, similar to the case of regular graphs, by a recursive procedure. One can then determine the threshold channel parameter for which $P_{\mathcal{T}}^{(\lambda,\rho)}(\ell) \to 0$ as $\ell \to \infty$.

However, this only analyzed the *average* behavior of the ensemble of codes. What we would like is for a random code drawn from the ensemble $\mathsf{LDPC}(n, \lambda, \rho)$ to concentrate around the average behavior with high probability. This would mean that almost all codes behave alike and thus the individual behavior of almost all codes is characterized by the average behavior of the ensemble (which can be computed as outlined above). A major success of this theory is that such a concentration phenomenon indeed holds, as shown in [17] and later extended to a large class of channels in [23]. The proof uses martingale arguments where the edges of the factor graph and then the inputs to the decoder are revealed one by one. We refrain from presenting the details here and point the reader to [17, Thm. 1] and [23, Thm. 2] (the result is proved for regular ensembles in these works but extends to irregular ensembles as long as the degrees in the graph are bounded).

In summary, it suffices to analyze and bound $P_{\mathcal{T}}^{(\lambda,\rho)}(\ell)$, and if this tends to 0 as $\ell \to \infty$, then in the limit of a large number of decoding iterations, for almost all codes in the ensemble, the actual bit error probability of the decoder tends to zero for large enough block lengths.

Order of limits: A remark on the order of the limits might be in order. The proposed style of analysis aims to determine the threshold channel parameter for which $\lim_{\ell\to\infty} \lim_{n\to\infty} E[P_n^{(\lambda,\rho)}(\ell)] = 0$. That is, we first fix the number of iterations and determine the limiting performance of an ensemble as the block length tends to infinity, and then let the number of iterations tend to infinity. Exchanging the order of limits gives us the quantity $\lim_{\ell\to\infty} \lim_{n\to\infty} E[P_n^{(\lambda,\rho)}(\ell)]$. It is this limit that corresponds to the more typical scenario in practice where for each fixed block length, we let the iterative decoder run until no further progress is achieved. We are then interested in the limiting performance as the block length tends to infinity. For the BEC, it has been shown that for both the orders of taking limits, we get the same threshold [25, Sec. 2.9.8]. Based on empirical observations, the same has been conjectured for channels such as the BSC, but a proof of this seems to be out of sight.

6.4 Analysis of average performance for the BEC

We now turn to analyzing the average behavior of the ensemble $\mathsf{LDPC}(n, \lambda, \rho)$ under message-passing decoding on the BEC. (The algorithm for regular codes from Section 5.2 extends to irregular codes in the obvious fashion — the message maps are the same except the maps at different nodes will have different number of arguments.)

Lemma 5 (Performance of tree ensemble channel on BEC). *Consider a degree distribution pair (λ, ρ) and a real number $0 < \alpha < 1$. Define $x_0 = \alpha$ and for $\ell \geq 1$,*

$$x_\ell = \alpha\lambda(1 - \rho(1 - x_{\ell-1})) \ . \tag{6}$$

Then, for the BEC with erasure probability α, for every $\ell \geq 1$, we have $P_{\mathcal{T}}^{(\lambda,\rho)}(\ell) = x_\ell$.

Proof. The proof follows along the lines of the recursion (1) that we established for the regular case. The case $\ell = 0$ is clear since the initial variable-to-check message equals the received value which equals an erasure with probability α. Assume that for $0 \leq i < \ell$, $P_{\mathcal{T}}^{(\lambda,\rho)}(i) = x_i$. In the ℓ'th iteration, a check-to-variable node message sent by a degree i check node is the erasure message if any of the $(i-1)$ incoming messages is an erasure, an event that occurs with probability $1-(1-x_{\ell-1})^{i-1}$ (since the incoming messages are independent and each is an erasure with probability $x_{\ell-1}$ by induction). Since the edge has probability ρ_i to be connected to a check node of degree i, the erasure probability of a check-to-variable message in the ℓ'th iteration for a randomly chosen edge is equal to $\sum_i \rho_i(1-(1-x_{\ell-1})^{i-1}) = 1-\rho(1-x_{\ell-1})$. Now consider a variable-to-check message in the ℓ'th iteration sent by a variable node of degree i. This is an erasure iff the node was originally erased and each of the $(i-1)$ incoming messages are erasures. Thus it is an erasure with probability $\alpha(1-\rho(1-x_{\ell-1}))^{i-1}$. Averaging over the edge degree distribution $\lambda(\cdot)$, we have $P_{\mathcal{T}}^{(\lambda,\rho)}(\ell) = \alpha\lambda(1-\rho(1-x_{\ell-1})) = x_\ell$. ■

The following lemma yields the threshold erasure probability for a given degree distribution pair (λ,ρ). The proof is identical to Lemma 2 — we simply use the recursion (6) in place of (1). Note that Lemma 2 is a special case when $\lambda(z) = z^{d_v-1}$ and $\rho(z) = z^{d_c-1}$.

Lemma 6. *For the BEC, the threshold erasure probability $\alpha^{\mathsf{MP}}(\lambda,\rho)$ below which the above iterative message passing algorithm leads to vanishing bit-erasure probability as the number of iterations grows is given by*

$$\alpha^{\mathsf{MP}}(\lambda,\rho) = \min_{x\in[0,1]} \frac{x}{\lambda(1-\rho(1-x))} . \tag{7}$$

6.5 Capacity achieving distributions for the BEC

Having analyzed the performance possible on the BEC for a given degree distribution pair (λ,ρ), we now turn to the question of what pairs (λ,ρ), if any, have a threshold approaching capacity. Recalling the designed rate from (4), the goal is to find (λ,ρ) for which $\alpha^{\mathsf{MP}}(\lambda,\rho) \approx \frac{\int_0^1 \rho(z)dz}{\int_0^1 \lambda(z)dz}$.

We now discuss a recipe for constructing such degree distributions, as discussed in [20] and [25, Sec. 2.9.11] (we follow the latter description closely). In the following we use parameters $\theta > 0$ and a positive integer N that will be fixed later. Let $\mathcal{D}$ be the space of non-zero functions $h : [0,1) \to \mathbb{R}^+$ which are analytic around zero with a Taylor series expansion comprising of non-negative coefficients. Pick functions $\hat{\lambda}_\theta(x) \in \mathcal{D}$ and $\rho_\theta(x) \in \mathcal{D}$ that satisfy $\rho_\theta(1) = 1$ and

$$\hat{\lambda}_\theta(1-\rho_\theta(1-x)) = x , \quad \forall x \in [0,1) . \tag{8}$$

Here are two example choices of such functions:

1. Heavy-Tail Poisson Distribution [16], dubbed "Tornado sequence" in the literature. Here we take

$$\begin{aligned}\hat{\lambda}_\theta(x) &= \frac{-\ln(1-x)}{\theta} = \frac{1}{\theta}\sum_{i=1}^{\infty}\frac{x^i}{i}\ , \text{ and}\\ \rho_\theta(x) &= e^{\theta(x-1)} = e^{-\theta}\sum_{i=0}^{\infty}\frac{\theta^i x^i}{i!}\ .\end{aligned}$$

2. Check-concentrated degree distribution [28]. Here for $\theta \in (0,1)$ so that $1/\theta$ is an integer, we take

$$\begin{aligned}\hat{\lambda}_\theta(x) &= 1-(1-x)^\theta = \sum_{i=1}^{\infty}\binom{\theta}{i}(-1)^{i-1}x^i\ , \text{ and}\\ \rho_\theta(x) &= x^{1/\theta}\ .\end{aligned}$$

Let $\hat{\lambda}_\theta^{(N)}(x)$ be the function consisting of the first N terms (up to the x^{N-1} term) of the Taylor series expansion of $\hat{\lambda}_\theta(x)$ around zero, and define the normalized function $\lambda_\theta^{(N)}(x) = \frac{\hat{\lambda}_\theta^{(N)}(x)}{\hat{\lambda}_\theta^{(N)}(1)}$ (for large enough N, $\hat{\lambda}_\theta^{(N)}(1) > 0$, and so this polynomial has positive coefficients). For suitable parameters N, θ, the pair $(\lambda_\theta^{(N)}, \rho_\theta)$ will be our candidate degree distribution pair.[10] The non-negativity of the Taylor series coefficients of $\hat{\lambda}_\theta(x)$ implies that for $x \in [0,1]$, $\hat{\lambda}_\theta(x) \geq \lambda_\theta^{(N)}(x)$, which together with (8) gives

$$x = \hat{\lambda}_\theta(1-\rho_\theta(1-x)) \geq \hat{\lambda}_\theta^{(N)}(1-\rho_\theta(1-x)) = \hat{\lambda}_\theta^{(N)}(1)\lambda_\theta^{(N)}(1-\rho_\theta(1-x))\ .$$

By the characterization of the threshold in Lemma 6, it follows that $\alpha^{\mathsf{MP}}(\lambda_\theta^{(N)}, \rho_\theta) \geq \hat{\lambda}_\theta^{(N)}(1)$. Note that the designed rate equals

$$r = r(\lambda_\theta^{(N)}, \rho_\theta) = 1 - \frac{\int_0^1 \rho_\theta(z)dz}{\int_0^1 \lambda_\theta^{(N)}(z)dz} = 1 - \hat{\lambda}_\theta^{(N)}(1)\frac{\int_0^1 \rho_\theta(z)dz}{\int_0^1 \hat{\lambda}_\theta^{(N)}(z)dz}\ .$$

Therefore, given a target erasure probability α, to communicate at rates close to capacity $1-\alpha$, the functions $\hat{\lambda}_\theta^{(N)}$ and ρ_θ must satisfy

$$\hat{\lambda}_\theta^{(N)}(1) \approx \alpha \quad \text{ and } \frac{\int_0^1 \rho_\theta(z)dz}{\int_0^1 \hat{\lambda}_\theta^{(N)}(z)dz} \to 1 \text{ as } N \to \infty\ . \tag{9}$$

[10] If the power series expansion of $\rho_\theta(x)$ is infinite, one can truncate it at a sufficiently high term and claimed bound on threshold still applies. Of course for the check-concentrated distribution, this is not an issue!

For example, for the Tornado sequence, $\hat{\lambda}_{\theta}^{(N)}(1) = \frac{1}{\theta}\sum_{i=1}^{N-1}\frac{1}{i} = \frac{\mathsf{H}(N-1)}{\theta}$ where $\mathsf{H}(m)$ is the Harmonic function. Hence, picking $\theta = \frac{\mathsf{H}(N-1)}{\alpha}$ ensures that the threshold is at least α. We have $\int_0^1 \hat{\lambda}_{\theta}^{(N)}(z)dz = \frac{1}{\theta}\sum_{i=1}^{N-1}\frac{1}{i(i+1)} = \frac{N-1}{\theta N}$, and $\int_0^1 \rho_\theta(z)dz = \frac{1-e^{-\theta}}{\theta}$. Therefore, $\frac{\int_0^1 \rho_\theta(z)dz}{\int_0^1 \hat{\lambda}_{\theta}^{(N)}(z)dz} = (1-e^{-\mathsf{H}(N-1)/\alpha})(1-1/N) \to 1$ as $N \to \infty$, as desired. Thus the degree distribution pair is explicitly given by

$$\lambda^{(N)}(x) = \frac{1}{\mathsf{H}(N-1)}\sum_{i=1}^{N-1}\frac{x^i}{i}, \quad \rho^{(N)}(x) = e^{\frac{\mathsf{H}(N-1)}{\alpha}(x-1)}.$$

Note that picking $N \approx 1/\varepsilon$ yields a rate $(1-\varepsilon)\alpha$ for reliable communication on BEC_α. The average variable node degree equals $\frac{1}{\int_0^1 \lambda^{(N)}(z)dz} \approx \mathsf{H}(N-1) \approx \ln N$. Therefore, we conclude that we achieve a rate within a multiplicative factor $(1-\varepsilon)$ of capacity with decoding complexity $O(n\log(1/\varepsilon))$.

For the check-concentrated distribution, if we want to achieve $\alpha^{\mathsf{MP}}(\lambda_{\theta}^{(N)}, \rho_\theta) \geq \alpha$ and a rate $r \geq (1-\varepsilon)\alpha$, then it turns out that the choice $N \approx 1/\varepsilon$ and $1/\theta = \lceil\frac{\ln N}{-\ln(1-\alpha)}\rceil$ works. In particular, this means that the factor graph has at most $O(n\log(1/\varepsilon))$ edges, and hence the "Peeling decoder" will again run in $O(n\log(1/\varepsilon))$ time.

One might wonder that among the various capacity achieving degree distributions that might exist for the BEC, which one is the "best" choice? It turns out that in order to achieve a fraction $(1-\varepsilon)$ of capacity, the average degree of the factor graph has to be $\Omega(\ln(1/\varepsilon))$. This is shown in [26] using a variant of Gallager's argument for lower bounding the gap to capacity of LDPC codes. In fact, rather precise lower bounds on the sparsity of the factor graph are known, and the check-concentrated distribution is optimal in the sense that it matches these bounds very closely; see [26] for the detailed calculations.

In light of the above, it might seem that check-concentrated distributions are the final word in terms of the performance-complexity trade-off. While this is true in this framework of decoding LDPC codes, it turns out by using more complicated graph based codes, called Irregular Repeat-Accumulate Codes, even better trade-offs are possible [21]. We will briefly return to this aspect in Section 7.

6.6 Extensions to channels with errors

Spurred by the remarkable success of [16] in achieving capacity of the BEC, Luby *et al* [17] investigated the performance of irregular LDPC codes for the BSC.

In particular, they considered the natural extension of Gallager's Algorithm B to irregular graphs, where in iteration i, a variable node of degree j uses a threshold $b_{i,j}$ for flipping its value. Applying essentially the same arguments as in

Section 5.3.2, but accounting for the degree distributions, one gets the following recurrence for the expected fraction p_ℓ of incorrect variable-to-check messages in the ℓ'th iteration:

$$\begin{aligned} p_{i+1} = p_0 - p_0 \sum_{j=1}^{d_v^{\max}} \sum_{t=b_{i+1,j}}^{j} \binom{j-1}{t} \left(\frac{1+\rho(1-2p_i)}{2}\right)^{t} \left(\frac{1-\rho(1-2p_i)}{2}\right)^{j-1-t} \\ + (1-p_0) \sum_{j=1}^{d_v^{\max}} \sum_{t=b_{i+1,j}}^{j} \binom{j-1}{t} \left(\frac{1+\rho(1-2p_i)}{2}\right)^{j-1-t} \left(\frac{1-\rho(1-2p_i)}{2}\right)^{t} \end{aligned}$$

As with the regular case, the cut-off value $b_{i+1,j}$ can then be chosen to minimize the value of p_{i+1}, which is given by the smallest integer for which

$$\frac{1-p_0}{p_0} \leq \left(\frac{1+\rho(1-2p_i)}{1-\rho(1-2p_i)}\right)^{2b_{i+1,j}-j+1} .$$

Note that $2b_{i+1,j} - j + 1 = b_{i+1,j} - (j - 1 - b_{i+1,j})$ equals the difference between the number of check nodes that agree in the majority and the number that agree in the minority. Therefore, a variable node's decision in each iteration depends on whether this difference is above a certain threshold, regardless of its degree.

Based on this, the authors of [17] develop a linear programming approach to find a good λ given a distribution ρ, and use this to construct some good degree distributions. Then using the above recurrence they estimate the theoretically achievable threshold crossover probability. Following the development of the density evolution algorithm to track the performance of belief propagation decoding [23], the authors of [22] used optimization techniques to find good irregular degree distributions for belief propagation decoding. The BIAWGN channel was the primary focus in [22], but the authors also list a few examples that demonstrate the promise of the techniques for other channels. In particular, for the BSC with rate 1/2, they report a degree distribution pair with maximum variable node degree 75 and check-node distribution $\rho(x) = 0.25x^9 + 0.75x^{10}$ for which the computed threshold is 0.106, which is quite close to the Shannon capacity limit 0.11. The techniques were further refined and codes with rate 1/2 and a threshold of $\sigma^* \approx 0.9781$ (whose SNR is within 0.0045 dB of capacity) were reported for the BIAWGN in [3] — these codes use only two different check node degrees $j, j+1$ for some integer $j \geq 2$.

7 Linear encoding time and Repeat-Accumulate Codes

The linear decoding complexity of LDPC codes is one of their attractive features. Being linear codes, they generically admit quadratic time encoding. In this section, we briefly discuss how the encoding complexity can be improved, and give pointers to where results in this vein can be found in more detail.

The original Tornado codes paper [16] achieved linear time encoding using a cascade of several low-density generator matrix (LDGM) codes. In LDGM codes, the "factor" graph is actually used to compute actual check bits from the k message bits (instead of specifying parity checks that the codeword bits must obey). Due to the sparse nature of the graph, the check bits can be computed in linear time. These check bits are then used as message bits for the next layer, and so on, till the number of check bits becomes $O(\sqrt{k})$. These final set of check bits are encoded using a quadratic time encodable linear code.

We now mention an alternate approach to achieve linear time encoding for LDPC codes themselves (and not a cascaded variant as in [16]), based on finding a sparse parity check matrix with additional nice properties. Let $H \in \mathbb{F}_2^{m \times n}$ be the parity check matrix of an LDPC code of dimension $n - m$. By means of row and column operations, we can convert H into a form $\tilde{H}$ where the last m columns are linearly independent, and moreover the $m \times m$ submatrix consisting of the last m columns is lower triangular (with 1's on the diagonal). Using $\tilde{H}$, it is a simple matter of "back-substitution" to compute the m parity bits corresponding to the $n - m$ information bits (the encoding is *systematic*). The complexity of this encoding is governed by the number of 1's in $\tilde{H}$. In general, however, when we begin with a sparse H, the resulting matrix $\tilde{H}$ is no longer sparse. In a beautiful paper [24], Richardson and Urbanke propose finding an "approximate" lower triangulation of the parity check matrix that is still sparse. The idea is to make the top right $(m - g) \times (m - g)$ corner of the matrix lower triangular for some small "gap" parameter g. The encoding can be done in $O(n + g^2)$ time, which is linear if $g = O(\sqrt{n})$. Remarkably, for several distribution pairs (λ, ρ), including all the optimized ones listed in [22], it is shown in [24] that, with high probability over the choice of the code from the ensemble $\mathsf{LDPC}(n, \lambda, \rho)$, a gap of $O(\sqrt{n})$ can in fact be achieved, thus leading to linear encoding complexity!

Yet another approach to achieve linear encoding complexity that we would like to focus on (as it has some additional applications), is to use Irregular Repeat-Accumulate (IRA) codes. IRA codes were introduced by Jin, Khandekar and McEliece in [15], by generalizing the notion of Repeat-Accumulate codes from [4] in conjunction with ideas from the study of irregular LDPC codes.

IRA codes are defined as follows. Let (λ, ρ) be a degree distribution pair. Pick a random bipartite graph G with k *information* nodes on left (with a fraction λ_i of the edges being incident on information nodes of degree i), and $n > k$ *check* nodes on the right (with a fraction ρ_i of the edges incident being incident on check nodes of degree i). Actually, it turns out that one can pick the graph to be regular on the check node side and still achieve capacity, so we can even restrict ourselves to check-degree distributions given by $\rho_a = 1$ for some integer a. Using G, the encoding of the IRA code (of dimension k and block length n) proceeds as follows:

- Place the k message bits on the k information nodes.
- For $1 \leq i \leq n$, at the i'th check node, compute the bit $v_i \in \{1, -1\}$ which equals the parity (i.e., product, in ±1 notation) of the message bits placed on its neighbors.
- (Accumulation step) Output the codeword $(w_1, w_2, \ldots, w_n)$ where $w_j = \prod_{i=1}^{j} v_i$. In other words, we accumulate the parities of the prefixes of the bit sequence $(v_1, v_2, \ldots, v_n)$.

Note that the encoding takes $O(n)$ time. Each of the check nodes has constant degree, and thus the v_i's can be computed in linear time. The accumulation step can then be performed using additional $O(n)$ operations.

It is not hard to show that the rate of the IRA code corresponding to a pair (λ, ρ) as defined above equals $\frac{\int_0^1 \lambda(z)dz}{\int_0^1 \rho(z)dz}$.

A natural iterative decoding algorithm for IRA codes is presented and analyzed in [4] (a description also appears in [21]). The iterative algorithm uses a graphical model for message passing that includes the above bipartite graph G connecting information nodes to check nodes, juxtaposed with another bipartite graph connecting the check nodes to n *code* nodes labeled $x_1, x_2, \ldots, x_n$. In this graph, which is intended to reflect the accumulation process, code node x_i for $1 \leq i < n$ is connected to the i'th and $(i+1)$'th check nodes (ones where v_i, v_{i+1} are computed), and node x_n is connected to the check node where v_n is computed.

It is proved (see [21, Sec. 2]) that for the above *non-systematic* IRA codes, the iterative decoding on BEC_α converges to vanishing bit-erasure probability as the block length $n \to \infty$, provided

$$\lambda\left(1 - \left[\frac{1-\alpha}{1-\alpha R(1-x)}\right]^2 \rho(1-x)\right) < x \quad \forall x \in (0,1] \,. \tag{10}$$

In the above $R(x) = \sum_{i=1}^{\infty} R_i x^i$ is the power series whose coefficient R_i equals the fraction of check nodes that are connected to i information nodes in G. Recalling (5), we have $R(x) = \frac{\int_0^x \rho(z)dz}{\int_0^1 \rho(z)dz}$.

Using the above characterization, degree distribution pairs (λ, ρ) for IRA codes that achieve the capacity of the BEC have been found in [4, 27].[11] In particular, we

[11] Actually, these papers work with a *systematic* version of IRA where the codeword includes the message bits in addition to the accumulated check bits $x_1, \ldots, x_n$. Such systematic codes have rate equal to $\left(1 + \frac{\int_0^1 \rho(z)dz}{\int_0^1 \lambda(z)dz}\right)^{-1}$, and the decoding success condition (10) for them is slightly different, with a factor α multiplying the $\lambda(\cdot)$ term on the left hand side.

want to draw attention to the construction in [21] with $\rho(x) = x^2$ that can achieve a rate of $(1-\varepsilon)(1-\alpha)$, i.e., within a $(1-\varepsilon)$ multiplicative factor of the capacity of the BEC, for $\alpha \in [0, 0.95]$.[12] Since $\rho(x) = x^2$, all check nodes are connected to exactly 3 information nodes. Together with the two code nodes they are connected to, each check node has degree 5 in the graphical model used for iterative decoding. The total number of edges in graphical model is thus $5n$, and this means that the complexity of the encoder as well as the "Peeling" implementation of the decoder is at most $5n$. In other words, the complexity per codeword bit of encoding and decoding is bounded by an absolute constant, independent of the gap ε to capacity.

8 Summary

We have seen that LDPC codes together with natural message-passing algorithms constitute a powerful approach for the channel coding problem and to approach the capacity of a variety of channels. For the particularly simple binary erasure channel, irregular LDPC codes with carefully tailored degree distributions can be used to communicate at rates arbitrarily close to Shannon capacity. Despite the impressive strides in the asymptotic analysis of iterative decoding of irregular LDPC codes, for all nontrivial channels except for the BEC, it is still unknown if there exist sequences of degree distributions that can get arbitrarily close to the Shannon limit. By optimizing degree distributions numerically and then computing their threshold (either using explicit recurrences or using the density evolution algorithm), various rather excellent bounds on thresholds are known for the BSC and BIAWGN. These, however, still do not come close to answering the big theoretical open question on whether there are capacity-achieving ensembles of irregular LDPC codes (say for the BSC), nor do they provide much insight into their structure.

For irregular LDPC codes, we have explicit sequences of *ensembles* of codes that achieve the capacity of the BEC (and come pretty close for the BSC and the BIAWGN channel). The codes themselves are not fully explicit, but rather sampled from the ensemble. While the concentration bounds guarantee that almost all codes from the ensemble are likely to be good, it may still be nice to have an explicit family of codes (rather than ensembles) with these properties. Even for achieving capacity of the BEC, the only known "explicit" codes require a brute-force search for a rather large constant sized code, and the dependence of the decoding complexity on the gap ε to capacity is not as good as for irregular LDPC ensembles. For the case of errors, achieving a polynomial dependence on the gap ε to capacity remains an important challenge.

[12] The claim is conjectured to hold also for $\alpha \in (0.95, 1)$.

References

[1] N. Alon and M. Luby. A linear time erasure-resilient code with nearly optimal recovery. *IEEE Transactions on Information Theory*, 42(6):1732–1736, 1996.

[2] L. Bazzi, T. J. Richardson, and R. L. Urbanke. Exact thresholds and optimal codes for the binary-symmetric channel and Gallager's decoding algorithm A. *IEEE Transactions on Information Theory*, 50(9):2010–2021, 2004.

[3] S. Chung, J. G. D. Forney, T. Richardson, and R. Urbanke. On the design of low-density parity-check codes within 0.0045 dB of the shannon limit. *IEEE Communications Letters*, 5:58–60, February 2001.

[4] D. Divsalar, H. Jin, and R. J. McEliece. Coding theorems for 'turbo-like' codes. In *Proc. of the 36th Allerton Conference on Communication, Control, and Computing*, pages 201–210, 1998.

[5] P. Elias. Coding for two noisy channels. *Information Theory, Third London Symposium*, pages 61–76, September 1955.

[6] J. Feldman and C. Stein. LP decoding achieves capacity. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 460–469, 2005.

[7] G. D. Forney. *Concatenated Codes*. MIT Press, Cambridge, MA, 1966.

[8] R. G. Gallager. *Low-Density Parity-Check Codes*. MIT Press, 1963.

[9] V. Guruswami. Error-correcting codes and expander graphs. *SIGACT News*, 35(3):25–41, September 2004.

[10] V. Guruswami. *List Decoding: Achieving Capacity for Worst-Case Errors*. Foundations and Trends in Theoretical Computer Science (FnT-TCS). NOW publishers, 2006.

[11] V. Guruswami. List decoding in pseudorandomness and average-case complexity. In *Proceedings of the IEEE Information Theory Workshop*, pages 32–36, March 2006.

[12] V. Guruswami and P. Indyk. Linear-time encodable and list decodable codes. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC)*, pages 126–135, June 2003.

[13] V. Guruswami and P. Indyk. Linear-time encodable/decodable codes with near-optimal rate. *IEEE Transactions on Information Theory*, 51(10):3393–3400, October 2005.

[14] V. Guruswami and A. Rudra. Explicit capacity-achieving list-decodable codes. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, pages 1–10, May 2006.

[15] H. Jin, A. Khandekar, and R. J. McEliece. Irregular Repeat-Accumulate codes. In *Proceddings of the 2nd International Conference on Turbo Codes and Related Topics*, pages 1–8, September 2000.

[16] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman. Efficient erasure correcting codes. *IEEE Transactions on Information Theory*, 47(2):569–584, 2001.

[17] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman. Improved low-density parity-check codes using irregular graphs. *IEEE Transactions on Information Theory*, 47(2):585–598, 2001.

[18] D. MacKay. Good error correcting codes based on very sparse matrices. *IEEE Transactions on Information Theory*, 45(2):399–431, 1999.

[19] D. MacKay and R. Neal. Near shannon limit performance of low density parity check codes. *Electronic Letters*, 32:1645–1646, 1996.

[20] P. Oswald and A. Shokrollahi. Capacity-achieving sequences for the erasure channel. *IEEE Transactions on Information Theory*, 48(12):3017–3028, 2002.

[21] H. D. Pfister, I. Sason, and R. L. Urbanke. Capacity-achieving ensembles for the binary erasure channel with bounded complexity. *IEEE Transactions on Information Theory*, 51(7):2352–2379, 2005.

[22] T. Richardson, A. Shokrollahi, and R. Urbanke. Design of capacity-approaching irregular low-density parity-check codes. *IEEE Trans. Inform. Theory*, 47:619–637, February 2001.

[23] T. Richardson and R. Urbanke. The capacity of low-density parity check codes under message-passing decoding. *IEEE Trans. Inform. Theory*, 47:599–618, February 2001.

[24] T. Richardson and R. Urbanke. Efficient encoding of low-density parity-check codes. *IEEE Trans. Inform. Theory*, 47:638–656, February 2001.

[25] T. Richardson and R. Urbanke. *Modern Coding Theory*. http://lthcwww.epfl.ch/mct/index.php, 2006.

[26] I. Sason and R. L. Urbanke. Parity-check density versus performance of binary linear block codes over memoryless symmetric channels. *IEEE Transactions on Information Theory*, 49(7):1611–1635, 2003.

[27] I. Sason and R. L. Urbanke. Complexity versus performance of capacity-achieving irregular repeat-accumulate codes on the binary erasure channel. *IEEE Transactions on Information Theory*, 50(6):1247–1256, 2004.

[28] M. A. Shokrollahi. New sequences of linear time erasure codes approaching the channel capacity. In *Proceesings of the 13th International Symposium on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes (AAECC)*, pages 65–76, 1999.

[29] M. Sipser and D. Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, 1996.

[30] D. Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Transactions on Information Theory*, 42(6):1723–1732, 1996.

[31] D. Spielman. Finding good LDPC codes. In *Proceedings of the 36th Annual Allerton Conference on Communication, Control, and Computing*, 1998.

[32] M. Sudan. List decoding: Algorithms and applications. *SIGACT News*, 31:16–27, 2000.

[33] R. M. Tanner. A recursive approach to low complexity codes. *IEEE Transactions on Information Theory*, 27(5):533–547, 1981.

[34] L. Trevisan. Some applications of coding theory in computational complexity. *Quaderni di Matematica*, 13:347–424, 2004.

[35] V. V. Zyablov and M. S. Pinsker. Estimation of the error-correction complexity of gallger low-density codes. *Problems of Information Transmission*, 11(1):18–28, 1976.

The Concurrency Column

by

Luca Aceto

BRICS, Department of Computer Science
Aalborg University, 9220 Aalborg Ø, Denmark

Dept. of Computer Science, School of Science and Engineering
Reykjavik University, 103 Reykjavik, Iceland

`luca@{cs.auc.dk,ru.is}`, `http://www.cs.auc.dk/~luca/BEATCS`

Fairness is an important concept that appears repeatedly in various forms in different areas of computer science, and plays a crucial role in the semantics and verification of reactive systems. Entire books are devoted to the notion of fairness—see, for instance, the monograph by Nissim Francez published in 1986—, and researchers in our community have painstakingly developed a taxonomy of various fairness properties that appear in the literature, such as unconditional fairness, weak fairness, strong fairness, and so on. This research is definitely important in light of the plethora of notions of fairness that have been proposed and studied in the literature.

But when is a temporal property expressing a fairness requirement? The authors of this column have recently developed a very satisfying answer to this fundamental question by offering three equivalent characterizations of “fairness properties” in the setting of linear-time temporal logic: a language-theoretic, a topological, and a game-theoretic characterization. This survey discusses these recent results in a very accessible fashion, and provides also a beautiful link between the study of fairness and classic probability theory.

I trust that you will enjoy reading it as much as I did. It is not often that one sees notions and results from several areas of mathematics and computer science combine so well to offer a formalization of a concept that confirms our intuition about it.

Bulletin of the EATCS no 90, pp. 89–108, October 2006

New Perspectives on Fairness

Daniele Varacca
Imperial College London, UK

Hagen Völzer
Universität zu Lübeck, Germany

Abstract

We define when a linear-time temporal property is a *fairness property* with respect to a given system. This captures the essence that is shared by most fairness assumptions that are used in the specification and verification of reactive concurrent systems, such as weak fairness, strong fairness, *k*-fairness, and many others. We give three characterisations for the family of all fairness properties: a language-theoretic, a topological, and a game-theoretic characterisation. It turns out that the fairness properties are the "large" sets from a topological point of view, i.e., they are the *co-meager* sets in the natural topology of runs of a given system. This insight provides a link to probability theory where a set is "large" when it has measure 1. While these two notions of largeness are very similar, they do not coincide in general. However, we show that they coincide for ω-regular properties and bounded Borel measures. That is, an ω-regular temporal property of a finite-state system has measure 1 under a bounded Borel measure if and only if it is a fairness property with respect to that system.

1 Introduction

When we model a concurrent system, we often make use of *nondeterminism*. Nondeterminism abstracts away from different scheduling policies or differences in speed of different parts of the system. Also, if we consider reactive systems, we use nondeterminism to allow different possible interactions with the environment. Furthermore, nondeterminism is used to model freedom of implementation.

A specification for a nondeterministic model of a system must allow several different behaviours. Specifications can thus be seen as sets of behaviours. We then say that a model satisfies the specification if all possible behaviours of the model belong to the specification.

Examples of specifications are *safety* and *liveness*. A safety specification informally requires that "some finite bad thing does not happen". If a behaviour violates the safety specification, we can recognise this in finite time. Once the "bad thing" has happened, any extension of the behaviour will violate the safety specification. A liveness specification informally requires that "some (possibly infinite) good thing will eventually happen". No finite behaviour should violate the specification, and therefore at any finite time we still have the possibility to eventually obtain a behaviour that belongs to the liveness specification.

When a model does not satisfy the specification, this can happen for several reasons. The model could be flawed and should be redesigned or the specification could be incorrect. Often, however, some behaviour of the model is not allowed by the specification, but such behaviour is "unlikely" to happen. How do we formalise this notion of unlikelihood?

We introduce nondeterminism to abstract away from some details of the implementation, but in some cases we may be abstracting away too much. For instance, if the nondeterminism is used to abstract away from scheduling policies, we could introduce some behaviour that no concrete scheduling policy would allow. The interaction with the environment can also be considered as a form of scheduling. Also in this case, some patterns of interaction may be allowed by the model, but they might not be happening in practice. This is a first sense in which a behaviour is unlikely.

To deal with this problem, we make use of *fairness assumptions*. Informally, a fairness assumption is an abstract description of a class of schedulers (or environments). A fairness assumption is a set of behaviours that are considered to be "fair". A model satisfies a specification under a fairness assumption, if all behaviours of the model that violate the specification are "unfair".

When can a set of behaviours be considered a fairness assumption? Informally, a scheduler is fair with respect to some (finite) behaviour if, whenever the behaviour is sufficiently often possible, then the scheduler guarantees it to happen sufficiently often. But how do we characterise this intuition formally? How do we formalise "possible", and "sufficiently often"? We will present, by means of examples, different degrees of "possible" and "sufficiently often". We will then show a formal characterisation of fairness that subsumes all the examples we present.

Another possible formalisation of unlikelihood is by means of probability theory. If the set of behaviours is endowed with a probability measure, we say that a set of behaviours is unlikely, if its probability is 0. In this sense a model satisfies a specification, if the set of the behaviours that violate the specification has probability 0.

We will compare this notion of probabilistic unlikelihood with the above notion of fairness, observing the similarities and the differences.

2 Examples of Fairness

We will show here some simple examples of the use of fairness assumptions.

2.1 Maximality

Consider the following system (Fig. 1), represented as a safe Petri net.

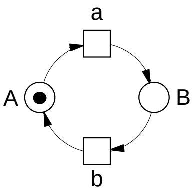


Figure 1: A simple process

As such, the system only says what can and what cannot happen. It does not say that something must happen at all. To say that something must happen, we can use the maximality assumption, which says that the system does not arbitrarily stop the computation. More precisely, a run (i.e., firing sequence) is *maximal* if it is infinite or if its final state does not enable any transition of the system. In the considered system, this means that after every a, there must be a b and that after every b, there must be an a. This leaves only the run $(ab)^{\omega}$, which is the unique maximal run of the system. Therefore, the system satisfies the property "infinitely often a" under the maximality assumption.

2.2 Weak fairness

Consider now the following system (Fig. 2) and assume maximality.

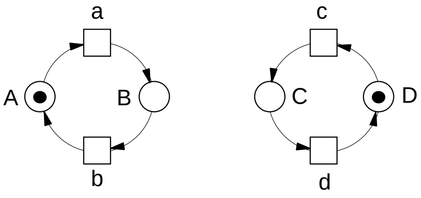


Figure 2: Two independent processes

Then, that system does not satisfy "infinitely often a" because the maximal run $(cd)^{\omega}$ does not. Although the overall system does not stop in this run, one of its components does.

In order to rule out such a behaviour, we assume *weak fairness* [15]. A run is *weakly fair* with respect to transition t if t is taken infinitely often or t is always eventually disabled. Therefore, the maximal run $(cd)^{\omega}$ is not weakly fair with respect to a. The system does in fact satisfy "infinitely often a" under weak fairness with respect to a and b.

Weak fairness with respect to all transitions is strictly stronger than maximality.

2.3 Strong fairness

In the next system below (Fig. 3), weak fairness is not sufficient to establish "infinitely often a" because the run $(cd)^{\omega}$ is weakly fair with respect to all transitions of the system. In particular, it is weakly fair with respect to a because a is always eventually disabled.

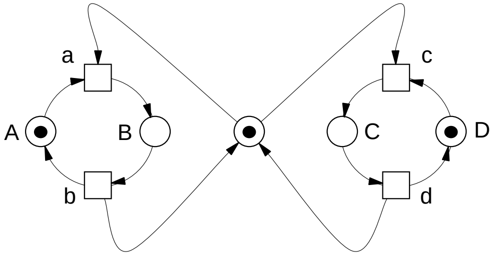


Figure 3: Two processes sharing a resource

However, we can consider $(cd)^{\omega}$ unfair with respect to a because a is infinitely often enabled but never taken. This kind of unfairness is captured by the notion of *strong fairness* [15]. A run is *strongly fair* with respect to a transition t if t is taken infinitely often or t is eventually always disabled. Strong fairness with respect to a and weak fairness with respect to b then establish "infinitely often a" in the system.

Strong fairness is obviously stronger than weak fairness.

2.4 *k*-Fairness

In the next system below (Fig. 4), strong fairness with respect to all transitions fails to establish "infinitely often e", because the run $(abcd)^{\omega}$ violates it but is strongly fair. In particular, it is strongly fair with respect to e because e is never enabled.

Among the fairness notions that establish "infinitely often e", there is the notion of *strong k-fairness* [8] for $k \geq 1$. A run is *strongly k-fair* with respect to

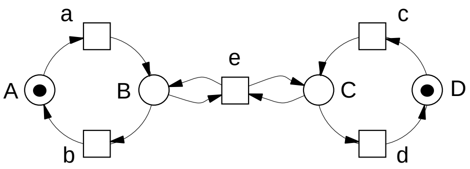


Figure 4: Two processes sharing an action

transition t if t is infinitely often taken or t is eventually never *k-enabled*, where t is *k-enabled* in a state s if there is a path of the system of length not more than k that starts in s and ends in a state that enables t. Weak fairness for all transitions and strong 1-fairness for e indeed establish "infinitely often e".

Strong $(k + 1)$-fairness is clearly stronger than strong k-fairness, and strong 0-fairness coincides with strong fairness.

Remark. The "unfairness" arising in the system in Fig. 4 is also known from the variant of the Dining Philosophers in which a philosopher picks up both his forks at the same time to eat. There, a philosopher may starve because his two neighbours "conspire" against him by eating alternately in such a way that his two forks are never available at the same time. Note that transition e in Fig. 4 needs two resources (B and C) at the same time. There are fairness notions that better capture the "unfairness" in this example (cf. [5, 25, 26]). However, we do not introduce them in detail here.

2.5 ∞-Fairness

Consider now the following infinite-state system (Fig. 5).

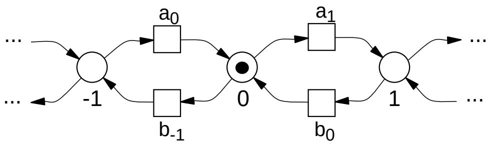


Figure 5: A nondeterministic walk on the integer line

Suppose we are interested here in the property "state 0 is visited infinitely often". This property is not established by strong k-fairness for any k because the diverging run $a_1 a_2 \ldots$ is strongly k-fair with respect to any transition for any $k \geq 0$. However, we can use the stronger notion of *strong ∞-fairness* [8]. A run is *strongly* ∞*-fair* with respect to a transition t, if t is infinitely often taken or t is eventually never ∞*-enabled*, where t is ∞*-enabled* in a state s if there is a path of the system (of any length) that starts in s and ends in a state that enables t. It is easy to see that ∞-fairness with respect to a_0 and b_0 establishes the required specification.

2.6 Fairness with respect to words

While strong ∞-fairness with respect to transitions is very strong, there are still some useful specifications that are not established by it. As an example, consider the following system and the specification "the finite word *ba* of transitions occurs infinitely often". The run $(abcd)^\omega$ does not satisfy the specification but it is strongly ∞-fair with respect to every transition, since every transition is taken infinitely often in this run. In such a case, we can extend the above fairness notions and define them with respect to finite words of transitions rather than with respect to a single transition only. For example, we can see that strong fairness with respect to the word *ba* establishes the specification considered above.

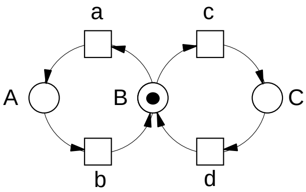


Figure 6: A recurrent free choice

2.7 Other examples

Another remarkable notion is *equifairness* [9]. Equifairness with respect to a and c in Fig. 6 prescribes that each fair run has infinitely many positions such that the number of previous occurrences of a equals the number of previous occurrences of c.

Fairness notions that were developed for the verification of randomised systems are *extreme fairness* [21] and α*-fairness* [16]. There are many more fairness notions in the literature, which we cannot all mention here. Overviews can be found in [9, 11, 4, 10, 14].

3 Formal Setting

Most researchers would agree that the above are all examples of fairness assumptions. The intuitive reason is that in all cases, we consider a run to be fair if whenever some transition (or some sequence of transitions) is sufficiently often possible, then it is sufficiently often executed.

This intuitive explanation lacks precision. What is the most general sense of “sufficiently often”? What do we mean by “possible”? Can we consider a notion more general than “transition”? In order to answer these questions, we need first to describe a precise formal setting.

3.1 Systems and runs

Let Σ be a countable set of *states*. Σ^* and Σ^ω denote the set of finite, and infinite sequences over Σ respectively. The set of all sequences $\Sigma^* \cup \Sigma^\omega$ is denoted as Σ^∞. We use the symbols α, β for denoting finite sequences, and x, y for arbitrary sequences. The length of a sequence x is denoted by $|x|$ ($= \omega$ if x is infinite). Concatenation of sequences is denoted by juxtaposition; $\sqsubseteq$ denotes the usual *prefix order* on sequences. Given a set X of sequences, we denote by $\max(X)$ the set of maximal elements of X under the prefix order. By $x{\uparrow} = \{y \mid x \sqsubseteq y\}$ and $x{\downarrow} = \{y \mid y \sqsubseteq x\}$ we denote the set of all *extensions* and *prefixes* of a sequence x respectively. The least upper bound of a sequence $(\alpha_i)_{i=0,1,\ldots}$ of finite sequences where $\alpha_i \sqsubseteq \alpha_{i+1}$ is denoted by $\sup_i \alpha_i$. For a sequence $x = s_0, s_1, \ldots$ and a position i where $0 \leq i < |x|$ of x, x_i denotes the i-th prefix $s_0, \ldots, s_i$ of x.

A *system* M is a tuple $\langle \Sigma, R \subseteq \Sigma \times \Sigma,\ \Sigma_0 \subseteq \Sigma \rangle$, where R is a *transition relation* between states, and Σ_0 is a set of *initial states*. The system is *finite* if Σ is. A *path* of a system M is a sequence in Σ^∞ that starts in an initial state and every two consecutive states are in the transition relation. The set of all paths of M is denoted by $L(M)$.

3.2 Temporal properties

A *temporal property* (*property* for short) is a set of sequences $E \subseteq \Sigma^\infty$. We say that E is *finitary* if $E \subseteq \Sigma^*$ and E is *infinitary* if $E \subseteq \Sigma^\omega$. Furthermore,

- E is *downward-closed* if $x \in E$ and $y \sqsubseteq x$ implies $y \in E$.
- E is *complete* if $\alpha_i \in E$ for $i \in \mathbb{N}$ with $\alpha_i \sqsubseteq \alpha_{i+1}$ implies $\sup_i \alpha_i \in E$.

We say that some sequence x *satisfies* a property E if $x \in E$, otherwise we say that x *violates* E.

A property S is a *safety property* if for any sequence x violating S, there exists a finite prefix α of x that violates S and each extension of a sequence violating S violates S as well, i.e.:

$$\forall x \notin S : \exists \alpha \sqsubseteq x : \alpha\!\uparrow \cap\, S = \emptyset.$$

Equivalently, a property is a safety property precisely when it is *downward-closed* and *complete*. We can think of a safety property S as a tree where nodes are labelled with elements of Σ such that S is the set of all labelled paths starting in the root of the tree. The set $L(M)$ is a safety property for each system M. The set of all sequences Σ^∞ is also a safety property and can be seen a the set of runs of a "universal" system.

Consider a safety property S and a finite sequence $\alpha \in S$. A property E is *live in α* with respect to S, if there exists a sequence $x \in E \cap S$ such that $\alpha \sqsubseteq x$. Intuitively, E is live in a finite run of a system if the system has still a chance to satisfy E in the future[1]. A property E is a *liveness property for S* if E is live in every $\alpha \in S \cap \Sigma^*$. In this situation we also say that (S, E) is *machine-closed* [1, 4]. If $S = \Sigma^\infty$, then we simply say that E is a *liveness* property.

A property is *ω-regular* if it is a property accepted by some Büchi automaton, or, equivalently a property definable in Monadic Second Order logic (see e.g. [23]).

Examples. $\Sigma^{\leq k} = \{\alpha \in \Sigma^* \mid |\alpha| \leq k\}$ is a safety property for each $k \in \mathbb{N}$; Σ^* and Σ^ω are examples of liveness properties. While Σ^* is a liveness property with respect to each safety property S, Σ^ω is a liveness property with respect to S only if $\max(S) \subseteq \Sigma^\omega$; $\max(S)$ is always a liveness property with respect to S. Σ^∞ is the only property that is a safety as well as a liveness property.

3.3 Topological notions

A *topology* on a nonempty set Ω is a family $\mathscr{T} \subseteq 2^\Omega$ that is closed under union and finite intersection such that $\Omega, \emptyset \in \mathscr{T}$. The elements of $\mathscr{T}$ are called *open sets*. A family $\mathscr{B} \subseteq \mathscr{T}$ is a *base* for $\mathscr{T}$ if every open set $G \in \mathscr{T}$ is the union of members of $\mathscr{B}$.

The complement of an open set is called a *closed set*. The *closure* of a set $X \subseteq \Omega$, denoted by $\overline{X}$, is the smallest closed set that contains X. A set X is closed if and only if $X = \overline{X}$. A set X is *dense* if $\overline{X} = \Omega$. Equivalently, a set X is dense if for every nonempty open set G, $G \cap X$ is nonempty. The family of open sets $\mathscr{T}$ is not closed under countable intersection in general: A G_δ *set* is a set that is the intersection of countably many open sets.

[1] While there is life, there is hope. –*Cicero*

Given a safety property S, the *Scott topology* on S is the family of sets $G \subseteq S$ such that

$$\forall x \in G : \exists \alpha \sqsubseteq x : \alpha{\uparrow} \cap S \subseteq G.$$

The family $\{\alpha{\uparrow} \cap S \mid \alpha \in \Sigma^*\}$ is a basis for the Scott topology. Note that open sets are generated by finitary properties Q by $G = Q{\uparrow} \cap S = \bigcup_{\alpha \in Q} \alpha{\uparrow} \cap S$, i.e., there is an exact correspondence between open sets and finitary properties. Open sets can therefore be interpreted as *observations* that can be recognised in finite time.

It is easy to verify that safety properties are exactly the closed sets and that liveness properties are exactly the dense sets of the Scott topology on Σ^∞. It is a general theorem, that in any topological space, any set is the intersection of a closed and a dense set. Hence every temporal property can be obtained as the intersection of a safety and a liveness property [2].

Given a safety property S, we sometimes concentrate our attention to the set of maximal (finite or infinite) sequences $\max(S)$. Note that, since safety properties are downward closed, the set of maximal sequences $\max(S)$ uniquely identifies the property S, and so we can easily switch between the two points of view.

The *restriction* of the Scott topology on S to $\max(S)$ is the family of sets $(G \cap \max(S))$ where G is an open set of the Scott topology on S. The restriction of the Scott topology to $\max(\Sigma^\infty) = \Sigma^\omega$ is sometimes called the *Cantor topology*.

4 Fairness Properties

We have now all the preliminary tools to formally define fairness. We will present characterisations from three different points of view. Moreover, we will present the properties this notion enjoys.

4.1 First characterisation

In Section 2, we have seen examples of fairness of increasing strength that all fit the informal pattern "if something is sufficiently often possible, then it is sufficiently often taken". For example, ∞-fairness with respect to a word w instantiates "is possible" by "is live" and "something" by "the word w". Can we find a more general notion of fairness without doing violence to our intuition?

In fact, we can by instantiating the generic term "something" as a finitary property $Q \subseteq \Sigma^*$ where Q is "possible" in a finite run α if Q is live in α and Q is "taken" in α if $\alpha \in Q$. Furthermore, we choose to instantiate "sufficiently often" as "infinitely often". Hence we say that a finitary property Q is *infinitely often* satisfied by a sequence x (or that Q is *recurrent* in x) if infinitely many prefixes of x are in Q.

This gives us the following, strong notion of fairness:

Definition 1. Consider a safety property $S \subseteq \Sigma^\infty$. We say that a maximal sequence $x \in \max(S)$ is *fair* in S with respect to a finitary property Q if the following holds:

- if for infinitely many $i \in \mathbb{N}$, the property Q is live in x_i with respect to S , then for infinitely many $j \in \mathbb{N}$, x_j satisfies Q.

The set of fair runs in S w.r.t. Q is denoted as $\text{fair}(S, Q)$.

Note that every finite maximal run is vacuously fair, as Q cannot be infinitely often live in a finite run. Note also that a property Q is infinitely often live in a sequence x if and only if it is *always* live in x, that is if it is live in all prefixes of x.

Examples. If Q is the set of all finite sequences that end with a given transition t, then $\text{fair}(S, Q)$ is exactly strong ∞-fairness with respect to t as introduced in Section 2.5. This is easily generalised to ∞-fairness with respect to a word.

Definition 1 presents the strongest form of fairness we consider with respect to some finitary property Q. That notion could also be called ∞-*fairness* with respect to Q. Any weaker form of fairness, such as strong and weak fairness, can be obtained by weakening. We thus define that a property is a fairness property if it *contains* all fair runs with respect to some Q.

Definition 2. We say E is a *fairness property for S* if there exists a finitary property Q such that $\text{fair}(S, Q) \subseteq E$.

The definition easily implies the following observation.

Proposition 3. *A property E is a fairness property for S if and only if $E \cap \max(S)$ is.*

However it is sometimes convenient to consider non-maximal sequences, as we will discuss in Section 4.3.

Examples. Any property weaker than ∞-fairness (such as strong k-fairness etc.) is a fairness property according to Definition 2.

Therefore all fairness notions introduced in Section 2 generate fairness properties with respect to a given system. However, could we have chosen a more general definition? We postpone this discussion to Section 4.5. Before, we will provide two further independent characterisations.

4.2 Second characterisation

In the introduction, we argued that unfair runs are unlikely in an intuitive sense. Alternatively we could say that *most* runs are fair. We will later examine a probabilistic interpretation of "most". But can we formalise the notion of "most runs" without using probabilities? It turns out that we can, using topology.

In a topological space, we say that a set is *nowhere dense* if its closure does not contain any nonempty open set. For an intuition on nowhere dense sets, imagine B to be a set of "dirty" points. If B is a dense set, then it pollutes the whole topological space: wherever you go in the topological space, you will have some dirty point in the neighbourhood. If B is a "somewhere dense" set, then it pollutes part of the space. There are regions where you will be always near a dirty point, but possibly also clean neighbourhoods. Finally, if B is nowhere dense, then every clean point lives in a clean neighbourhood. Intuitively a nowhere dense set is small because the rest of the topological space can stay clear of it.

A set is *meager*, if it is the countable union of nowhere dense sets. Topologically, a countable union of small sets is still small. This was observed by the French mathematician René-Louis Baire, who proved that the unit interval of the real line cannot be obtained as the countable union of nowhere dense sets. This result can be thought of as a generalisation of Cantor's theorem, which states that the unit interval is not obtained as the countable union of points [19].

The complement of a "small" set is therefore to be thought of as "large". The complement of a meager set is called *co-meager* (or *residual*). In many topologies, including the Scott topology, co-meager sets can be equivalently characterised as follows:

Proposition 4. *In the Scott topology, a set is co-meager if and only if it contains a dense* G_δ *set.*

As announced, co-meager sets are precisely the fairness properties:

Theorem 5. *A property E is a fairness property for S if and only if $E \cap S$ is co-meager in the Scott topology of S.*

This point of view formalises the idea that "most" runs are fair. Indeed a property is a fairness property if (topologically) most runs belong to it.

Examples. The set of maximal sequences $\max(S)$ of a safety property S can be obtained as the intersection $\bigcap_{n\in\mathbb{N}} X_n$, where X_n is the set of sequences that are maximal or are longer than n. All such X_n are open in the Scott topology. This shows that maximality is a G_δ set. We already stated that $\max(S)$ is dense, i.e., a liveness property w.r.t. S, hence it is a fairness property.

4.3 Third characterisation

In the 1930ies, a group of Polish mathematicians would meet in a cafe, called the Scottish Café, in the now Ukrainian city of L'viv. During these meetings, they were posing each other problems and seeking the solution together. The minutes of these meetings were kept by the landlord and some of them were published later [18].

One of the problems, posed by Stanisław Mazur, and solved by him together with Stefan Banach involves the following game[2], since known as the *Banach-Mazur* game.

Let S be a safety property, and E any property. The game $G(S, E)$ is played by the two players called *Alter* and *Ego*. The state of a play is a finite sequence of S. At every move one player extends the current sequence by a finite, possibly empty sequence α_i yielding the sequence $\alpha_0 \dots \alpha_i \in S$. Alter has the first move. The play goes on forever converging to a finite sequence α or infinite sequence x in S. Ego wins if $x \in E$ (resp. $\alpha\!\uparrow\, \subseteq E$), otherwise Alter wins.

A *strategy* is a mapping $f : \Sigma^* \to \Sigma^*$ such that for each $\alpha \in S$, we have $\alpha f(\alpha) \in S$. A strategy f is *winning* for player P, if for each strategy g of the other player, P wins the play that results from P playing according to f and the other player playing according to g.

The question Mazur posed was: how do we characterise the sets for which Ego has a winning strategy? The answer is in the following theorem.

Theorem 6. *Ego has a winning strategy for the game $G(S, E)$ if and only if $E \cap S$ is co-meager in the Scott topology on S.*

Which obviously implies

Theorem 7. *A set E is a fairness property for S if and only if Ego has a winning strategy for the game $G(S, E)$.*

Note that, by Proposition 3, it is not restrictive to consider just target sets E that contain only maximal runs.

The intuition behind this characterisation is that, while fairness restricts the allowed behaviour, it should not restrict it too much. Ego, who wants to produce a fair run, can enforce some (live) choice to be taken infinitely often while she cannot prevent other choices being taken infinitely often (by Alter).

Examples. We can use Theorem 7 to prove that fair(S, Q) is a fairness property. When Q is not live in α, Ego does nothing. Otherwise, Ego extends to a finite sequence in Q. This is clearly a winning strategy for Ego for the target fair(S, Q).

[2] The original definition is slightly different and formulated in a different context: see also [19, 7, 20].

Theorem 7 can also be used to prove that a property is *not* a fairness property. Consider the system M of Section 2.2, and consider the set X of infinite runs of M that have the suffix $(cd)^{\omega}$. The set X is a liveness but not a fairness property for that system. Ego does not have a winning strategy for the game $G(L(M), X)$, because indeed Alter has a winning strategy: when it is his turn, Alter should just run the left-hand side component of the system, making sure that there are infinitely many a's and b's in the resulting sequence.

In the above example, we have shown that Ego does not have a winning strategy by showing that Alter has a winning strategy. A set for which one of the two players has a winning strategy is called *determinate*. The class of determinate properties is quite large. All *Borel sets*[3] are determinate [7], of which ω-regular properties constitute, in a sense, a very simple subclass [23]. In order to show the existence of an indeterminate set, one needs the axiom of choice.

4.4 Characteristics of fairness

We have described the same class of properties from three different points of view. We now state some characteristics of this class.

The characteristics we are going to list intuitively confirm our intuition on co-meagerness as "largeness". To help the intuition we will write "large" for co-meager, and "small" for meager. We will call a set *intermediate* if it is neither large nor small.

1. If a set is large, its complement is not.

2. Any superset of a large set is large.

3. The intersection of countably many large sets is large.

4. Intersection with a large set preserves size, i.e, if A is large and B is small (resp. intermediate, large), then $A \cap B$ is small (resp. intermediate, large).

5. When S is uncountable, every countable set is small, but there are also uncountable sets that are small.

6. Every large set is dense.

Property (6) says that for every fairness property E for S, the pair (S, E) is machine-closed, a property that has been described as the main feature of fairness

[3] The smallest family of sets that contains the Scott open sets and that is closed under complement and countable union.

by Apt, Francez, and Katz [4] and by Lamport [14]. Property (3) is important for modular specification. Fairness is usually imposed componentwise to the system (with respect to different transitions or processes); (3) assures that the overall fairness assumption, i.e., the intersection of all fairness assumptions for the components is again a fairness assumption.

4.5 Canonicity of the notion

How canonical is our definition of fairness? The fact that it has three independent characterisations makes this notion interesting. But could there be a more liberal definition of fairness?

Roughly, the answer is no if we insist on (3) and (6) in Section 4.4 above. More precisely:

Theorem 8. *Fairness is a maximal class of dense determinate properties that is closed under finite intersection.*

5 Probabilities

We have argued that unfair runs should be unlikely. We have shown a topological view of likelihood. A more common interpretation, however, is by means of probability theory. In this section we present this point of view. We show that probabilistic and topological likelihood are in general different notions, but that under some reasonable conditions, they in fact coincide.

5.1 Definitions

First we recall the standard setting of how probability is adjoined to systems.

A *σ-algebra* over a nonempty set X is a family $\mathscr{A}$ of subsets of X that contains the empty set and is closed under complementation and countable union. Given a topology, the *Borel* σ-algebra of the topology is the smallest σ-algebra that contains the open sets. A *probability measure* on a σ-algebra $\mathscr{A}$ over X is a function $\mu : \mathscr{A} \to [0, 1]$ such that $\mu(X) = 1$ and for any sequence of pairwise disjoint sets $(Y_i)_{i\in\mathbb{N}}$, $\mu(\bigcup_{i\in\mathbb{N}} Y_i) = \sum_{i\in\mathbb{N}} \mu(Y_i)$. A *Borel* probability measure of a topology is a probability measure over the Borel σ-algebra of the topology. Given a probability measure μ on $\mathscr{A}$, and two sets $A, B \in \mathscr{A}$, the *probability of A conditional to B*, is defined as $\mu(A \mid B) = \mu(A \cap B)/\mu(B)$.

Given a safety property S, consider a Borel probability measure μ over the restriction of the Scott topology to $\max(S)$. We say that μ is a *Markov measure* when $\mu(\alpha ss'{\uparrow} \mid \alpha s{\uparrow}) = \mu(\beta ss'{\uparrow} \mid \beta s{\uparrow})$ for all $\alpha, \beta \in S \cap \Sigma^*$ and $s, s' \in \Sigma$. We say that μ is *positive* if $\mu(\alpha\uparrow) > 0$ for each $\alpha \in S$, μ is said to be *bounded* if there exists a $c > 0$ such that $\mu(\alpha s\uparrow \mid \alpha\uparrow) > c$ for each $\alpha s \in S$. A Borel set $X \subseteq \max(S)$ is *μ-large* (or *probabilistically large* when μ is understood from the context) if $\mu(X) = 1$.

Example. Given a finite system M, consider a Markov chain on Σ that assigns positive probabilities to transitions iff they belong to R. This generates a Markov bounded measure on $\max(L(M))$.

5.2 Similarities and differences

Topological and probabilistic largeness are very similar notions. Oxtoby's classic book [19] is devoted to study these similarities. For instance all the properties characterising topological largeness described in Section 4.4 are valid also for probabilistic largeness[4].

Despite all the common properties, the two notions do not coincide in general: in fact there are sets that are topologically large but not probabilistically large as well as sets where it is the other way around.

As an example, consider the system in Fig. 5 in Section 2.5 together with the Markov measure such that each a_i has probability $p \neq 1/2$ and each b_i has probability $1-p$, i.e., we are looking at an asymmetric random walk on the integer line. It is well-known that the property X_1 = "state 0 is visited infinitely often" has probability 0. However, it is topologically large as X_1 is established by ∞-fairness as discussed in Section 2.5. Note that there is also a simple winning strategy for Ego.

We can also reformulate the above example in a finite-state system: Consider the system in Fig. 6 in Section 2.6 together with the Markov measure such that a has probability $p \neq 1/2$ and c has probability $1 - p$. Then, equifairness (cf. Sect. 2.7), i.e., X_2 = "the number of previous a's equals the number of previous c's infinitely often" has probability 0 but is topologically large. (Ego's winning strategy consists in evening up the count of the letters.)

5.3 Coincidence theorem

In light of the above examples, it was quite surprising to discover that under not very restrictive hypotheses, the two notions of largeness in fact coincide.

The restrictions we have to impose are the following: we restrict our attention to ω-regular properties on finite systems, and we need to consider only bounded

[4] Property (6) is valid for most probability measures.

measures. Note that all properties that can be described using standard temporal logics such as LTL are ω-regular.

In the first counterexample above, the system is infinite. In the second counterexample, we consider a bounded measure over a finite system, but the property X_2 is not ω-regular.

Theorem 9. *Let M be a finite system, μ a bounded Borel measure on $\max(L(M))$, and X an ω-regular property. Then X is topologically large in $L(M)$ if and only if X is also μ-large.*

The key observation behind the proof is that, for ω-regular properties on finite systems, if Ego has a winning strategy, then she has a memoryless winning strategy [7]. Another important fact is here that, each ω-regular property X is determinate, as already stated in Section 4.3. For the details of the proof see [24].

5.4 Consequences

The above coincidence result has several pleasing consequences. First, it implies that for ω-regular properties, probabilistic scheduling is “fair enough”, i.e., each ω-regular fairness property has probability 1 under such scheduling.

Secondly, the result can be applied to model checking. On the one hand, we can use qualitative probabilistic model checking techniques to decide whether there exists a fairness assumption under which a given system satisfies its linear-time specification. On the other hand, we can use the three characterisations of fairness to further our understanding of probabilistic model checking. We refer the interested reader to our paper [24].

Thirdly, the above result gives a rather nice proof of the folk theorem that “in qualitative probabilistic model checking the actual probability values do not matter”. It has been long well known that a system satisfies an ω-regular specification with probability 1 regardless of what the precise probabilities associated to the local choices are. Theorem 9 is a formalisation of this intuition and allows us to reason about properties having probability 1 without mentioning probabilities at all.

6 Historical Remarks

While safety and liveness have had a formal characterisation for a long time–given by Lamport [13] and Alpern and Schneider [2]–there was no satisfactory characterisation of fairness. Apt, Francez, and Katz [4] gave three criteria that each fairness assumption should meet. Among them, machine-closure[5] is the most

[5] Called *feasibility* in [4].

prominent. Lamport [14] reviewed their criteria and argues that fairness should be equated with machine-closure (i.e. density). Kwiatkowska [12] proposed to equate fairness with dense G_δ sets.

A couple of papers used the notion that we have described as fairness in different contexts without actually attempting to define fairness: Ben-Eliyahu and Magidor [6] observed that some popular fairness notions describe co-meager sets. Alur and Henzinger [3] propose that machine-closure should be strengthened to what they call *local liveness*, which is the same as fairness defined above. They gave the game-theoretic definition. The Banach-Mazur game has also been considered by Pistore and Vardi [20] as well as Berwanger, Grädel, and Kreutzer [7]. Berwanger et. al. [7] proved the memoryless determinacy result that lead to the coincidence theorem above.

The correspondence of safety and liveness to closed and dense sets given by Alpern and Schneider [2] goes back to G. Plotkin (see [2]) who in turn was motivated by Smyth [22]. Interestingly, Alpern and Schneider [2] write “Plotkin nevertheless is unhappy with our definition of liveness because it is not closed under intersection”. Note that in a sense, fairness with respect to the universal system Σ^∞ is the largest subclass of liveness that is closed under finite intersection as formally stated in Theorem 8. Manna and Pnueli [17] gave an alternative classification of temporal properties that is based on topology.

For more information, we refer the reader to [27, 24].

References

[1] Martín Abadi and Leslie Lamport. The existence of refinement mappings. *Theoretical Computer Science*, 82:253–284, 1991.

[2] Bowen Alpern and Fred B. Schneider. Defining liveness. *Information Processing Letters*, 21:181–185, 1985.

[3] Rajeev Alur and Thomas A. Henzinger. Local liveness for compositional modeling of fair reactive systems. In Pierre Wolper, editor, *CAV*, volume 939 of *Lecture Notes in Computer Science*, pages 166–179. Springer, 1995.

[4] Krysztof R. Apt, Nissim Francez, and Shmuel Katz. Appraising fairness in languages for distributed programming. *Distributed Computing*, 2:226–241, 1988.

[5] Paul C. Attie, Nissim Francez, and Orna Grumberg. Fairness and hyperfairness in multi-party interactions. *Distributed Computing*, 6:245–254, 1993.

[6] Rachel Ben-Eliyahu and Menachem Magidor. A temporal logic for proving properties of topologically general executions. *Information and Computation*, 124(2):127–144, 1996.

[7] Dietmar Berwanger, Erich Grädel, and Stephan Kreutzer. Once upon a time in a west - determinacy, definability, and complexity of path games. In Moshe Y. Vardi

and Andrei Voronkov, editors, *LPAR*, volume 2850 of *Lecture Notes in Computer Science*, pages 229–243. Springer, 2003.

[8] Eike Best. Fairness and conspiracies. *Information Processing Letters*, 18:215–220, 1984. Erratum ibidem 19:162.

[9] Nissim Francez. *Fairness*. Springer, 1986.

[10] Yuh-Jzer Joung. On fairness notions in distributed systems, part I: A characterization of implementability. *Information and Computation*, 166:1–34, 2001.

[11] Marta Z. Kwiatkowska. Survey of fairness notions. *Information and Software Technology*, 31(7):371–386, 1989.

[12] Marta Z. Kwiatkowska. On topological characterization of behavioural properties. In G. Reed, A. Roscoe, and R. Wachter, editors, *Topology and Category Theory in Computer Science*, pages 153–177. Oxford University Press, 1991.

[13] Leslie Lamport. Formal foundation for specification and verification. In M.W. Alford, J.P Ansart, G. Hommel, L. Lamport, B. Liskov, G.P. Mullery, and F.B. Schneider, editors, *Distributed Systems: Methods and Tools for Specification*, volume 190 of *LNCS*. Springer, 1985.

[14] Leslie Lamport. Fairness and hyperfairness. *Distributed Computing*, 13(4):239–245, 2000.

[15] Daniel J. Lehmann, Amir Pnueli, and Jonathan Stavi. Impartiality, justice and fairness: The ethics of concurrent termination. In Shimon Even and Oded Kariv, editors, *ICALP*, volume 115 of *Lecture Notes in Computer Science*, pages 264–277. Springer, 1981.

[16] Orna Lichtenstein, Amir Pnueli, and Lenore D. Zuck. The glory of the past. In Rohit Parikh, editor, *Logic of Programs*, volume 193 of *Lecture Notes in Computer Science*, pages 196–218. Springer, 1985.

[17] Zohar Manna and Amir Pnueli. A hierarchy of temporal properties. In *Proceedings of the 9th Annual ACM Symposium on Principles of Distributed Computing*, pages 377–408. ACM, 1990.

[18] R. Daniel Mauldin. *The Scottish Book: Mathematics from the Scottish Cafe*. Birkhäuser, 1981.

[19] John C. Oxtoby. *Measure and Category. A Survey of the Analogies between Topological and Measure Spaces*. Springer, 1971.

[20] Marco Pistore and Moshe Y. Vardi. The planning spectrum - one, two, three, infinity. In *LICS*, pages 234–243. IEEE Computer Society, 2003.

[21] Amir Pnueli. On the extremely fair treatment of probabilistic algorithms. In *STOC*, pages 278–290. ACM, 1983.

[22] Michael B. Smyth. Power domains and predicate transformers: A topological view. In Josep Díaz, editor, *ICALP*, volume 154 of *Lecture Notes in Computer Science*, pages 662–675. Springer, 1983.

[23] Wolfgang Thomas. Automata on infinite objects. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B: Formal Models and Semantics. Elsevier, 1990.

[24] Daniele Varacca and Hagen Völzer. Temporal logics and model checking for fairly correct systems. In *LICS*, pages 389–398. IEEE Computer Society, 2006.

[25] Hagen Völzer. Refinement-robust fairness. In Lubos Brim, Petr Jancar, Mojmír Kretínský, and Antonín Kucera, editors, *CONCUR*, volume 2421 of *Lecture Notes in Computer Science*, pages 547–561. Springer, 2002.

[26] Hagen Völzer. On conspiracies and hyperfairness in distributed computing. In Pierre Fraigniaud, editor, *DISC*, volume 3724 of *Lecture Notes in Computer Science*, pages 33–47. Springer, 2005.

[27] Hagen Völzer, Daniele Varacca, and Ekkart Kindler. Defining fairness. In Martín Abadi and Luca de Alfaro, editors, *CONCUR*, volume 3653 of *Lecture Notes in Computer Science*, pages 458–472. Springer, 2005.

The Distributed Computing Column

by

Mario Mavronicolas

Department of Computer Science, University of Cyprus
75 Kallipoleos St., CY-1678 Nicosia, Cyprus
mavronic@cs.ucy.ac.cy

Eight Open Problems in Distributed Computing

James Aspnes
Dept. of Computer Science
Yale University
New Haven, CT 06520-8285
aspnes@cs.yale.edu

Costas Busch
Dept. of Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12180
buschc@cs.rpi.edu

Shlomi Dolev
Dept. of Computer Science
Ben-Gurion University of the Negev
Beer-Sheva, Israel 84105
dolev@cs.bgu.ac.il

Panagiota Fatourou
Dept. of Computer Science
University of Ioannina
45110 Ioannina, Greece
faturu@cs.uoi.gr

Chryssis Georgiou
Dept. of Computer Science
University of Cyprus
CY-1678 Nicosia, Cyprus
chryssis@cs.ucy.ac.cy

Alex Shvartsman
Dept. of Computer Science and Engineering
University of Connecticut
Storrs, CT 06269
aas@cse.uconn.edu

Bulletin of the EATCS no 90, pp. 109–126, October 2006

Paul Spirakis
RACTI
265 00 Rio, Patras
spirakis@cti.gr

Roger Wattenhofer
Computer Engineering and Networks Lab.
ETH Zurich
8092 Zurich, Switzerland
wattenhofer@tik.ee.ethz.ch

Abstract

Distributed Computing Theory continues to be one of the most active research fields in Theoretical Computer Science today. Besides its foundational topics (such as consensus and synchronization), it is currently being enriched with many new topics inspired from modern technological advances (e.g., the Internet). In this note, we present eight open problems in Distributed Computing Theory that span a wide range of topics – both classical and modern.

1 Wait-Free Consensus

A *consensus protocol* is a distributed algorithm where n processes collectively arrive at a common decision value starting from individual process inputs. It must satisfy *agreement* (all processes decide on the same value), *validity* (the decision value is an input to some process), and *termination* (all processes eventually decide). A protocol in an asynchronous shared-memory system is *wait-free* if each process terminates in a finite number of its own steps regardless of scheduling. From the FLP impossibility result [31, 54], wait-free consensus is impossible. However, it becomes possible using randomization with the termination condition relaxed to hold with probability 1.

The **open question** that then arises is the complexity of solving consensus, measured by the expected number of register operations carried out by all processes (*total work*) or by any one process (*per-process work*).

This complexity depends strongly on assumptions about the power of the adversary scheduler. For an *adaptive adversary* that chooses the next process to run based on total knowledge of the current state of the system, the best known protocol using only atomic read-write registers takes $O(n^2 \log n)$ expected total work [14]. If counters supporting increment, decrement, and read operations are available, this drops to $O(n^2)$ expected total work [4]. No faster protocol is known using any objects that can be built from atomic registers, and there is a lower bound of $\Omega(n^2/\log^2 n)$ that holds even given powerful tools like unit-cost snapshots [6].

Closing the gap between the upper and lower bounds is interesting because all known polynomial-time wait-free consensus protocols are based on collecting enough random votes that one standard deviation in the total is larger than the $n-1$ votes that can be "hidden" by the adversary by selectively stopping processes, and it is not hard to show that simple variants on voting cannot yield subquadratic protocols. A faster protocol would thus require a significantly new approach. Conversely, an $\Omega(n^2)$ lower bound would show that voting is optimal.

With a weaker adversary that cannot observe coin flips that have not yet been made public, consensus can be solved in $O(\log n)$ work per process using multi-writer registers [11]. There is no corresponding non-trivial lower bound. It would be interesting to see if an $\Omega(\log n)$ lower bound could be proved for multi-writer registers or even with strong objects like unit-cost snapshots. Closing the gap in both models would show whether the cost of weak-adversary consensus arises from fundamental limitations of grouping local coin-flips together or merely from the weakness of atomic registers.

2 Oblivious Routing

A typical distributed computing environment consists of several processing units which communicate through some underlying multi-hop network. The network is usually modeled after a graph, possibly weighted, where nodes represent the processing units and the edges the communication links. The nodes communicate by exchanging messages in the form of packets. *Routing* is the task of selecting the paths that the packets will follow in the network. Ideally the selected paths should have small *congestion*, that is, the maximum number of paths crossing any edge should be small, and the paths should have small *stretch*, that is, the ratio between the selected path and the respective shortest path should be as small as possible.

Oblivious routing is a type of distributed routing suitable for dynamic packet arrivals. In oblivious routing, the path for a newly injected packet is selected in a way that it is not affected by the path choices of the other packets in the network. Räcke [66] gives an existential result that shows that for any network there exists an oblivious routing algorithm with congestion within factor $\log^3 n$ from that of the optimal off-line centralized algorithm, where n is the number of nodes. This oblivious algorithm constructs a path by choosing a logarithmic number of random intermediate nodes in the network. Azar *et al.* [13] showed that the probabilities for the random intermediate nodes can be computed a priori in polynomial time.

Even though congestion is a fundamental metric for the performance of routing algorithms, stretch is important too, since it represents the extra delay of the packets when there is no congestion. Ideally, stretch should be a constant. So far, the main research on oblivious routing algorithms has focused on optimizing the congestion while ignoring the stretch. For example, a packet may have destination to a neighbor node of the source and still the path chosen by an oblivious algorithm may be as long as the number of nodes in the network.

An interesting **open problem** is to examine the circumstances in which congestion and stretch can be optimized simultaneously.

There is a simple counter-example network that shows that in general the two metrics are orthogonal to each other: take an adjacent pair of nodes u, v and $\Theta(\sqrt{n})$ disjoint paths of length $\Theta(\sqrt{n})$ between u and v. For packets travelling from u to v, any routing algorithm that minimizes congestion has to use all the paths, however, in this way some packets follow long paths, giving high stretch. Nevertheless, there are special cases of interesting networks where congestion and dilation can be minimized simultaneously. For example, in grids [15], and in networks of uniformly distributed nodes in convex-like areas [16], the congestion is within a poly-logarithmic factor from optimal and stretch is constant.

A second interesting **open problem** is to find other classes of networks where the congestion and stretch are minimized simultaneously.

Possible candidates for such networks could be for example bounded-growth networks, or networks whose nodes are uniformly distributed in closed polygons, which describe interesting cases of wireless networks. Another interesting open problem is to find classes of networks in which oblivious routing gives $C + D$ close to the off-line optimal, where C is the congestion and D is the maximum path length. Such a result will have immediate consequences in packet scheduling algorithms since it is known from [52] that it is feasible to deliver the packets in time proportional to $C + D$.

3 Stability of Continuous Consensus

Consensus is a fundamental task in distributed computing, it allows to reduce a distributed task to a centralized task by agreeing on the system state, the inputs and (hence) the common transition. One shot consensus cannot be self-stabilizing [22] since it can terminate with disagreeing outputs. On the other hand, on-going consensus may stabilize to eventually ensure that when a new consensus instance is invoked the safety property for the output of this instance is correct [23].

In the scope of on-going (self-stabilizing) consensus task one may consider the sequence of inputs and outputs of instances [20, 24] and require stability of outputs as long as the inputs allow such a stability. For example, when one consensus instance output has been 1, and the next instance has 1 as a possible output value, then 1 should be preferred. Namely, we would like to minimize the number of times the output is changed.

The **open problem** is, to determine the most stable (consensus) function to use, given flexibility in deciding on the output of the system.

Namely, given a particular sequences of input changes, choose the function that changes output as least as possible, assuming that the function from the inputs to the common output is only restricted to ensure that the output has a value equal to at least $t + 1$ inputs. For example, if the system can remember (in memory) the last output, the system may stick to the output as long as it can: say the system includes five processors, at most two of which maybe faulty, i.e., $t = 2$. In case the inputs are 1, 1, 1, 1, 1 the system must output 1, then if the inputs are repeatedly changed to 1, 1, 1, 1, 0 and then to, say, 1, 1, 0, 1, 0 the system may stay with a stable output 1, but once the inputs are changed to, say, 1, 0, 0, 1, 0 the system output must be changed to 0.

The case of a geodesic path of input changes, where each input can be changed at most once is considered in [20, 24]. The upper bound for the memoryless binary input case in [20] is $2t + 1$ (where the majority of the first $2t + 1$ inputs defines the output). Multi-valued consensus extends the case of binary-valued consensus, allowing the inputs (and the output) to be a non-necessarily binary value.

An upper bound for the number of output changes for a memoryless symmetric system (where the function has the same output regardless of the position of inputs in the input vector) is presented in [20]. The upper bound is a factor of approximately 2 away from a corresponding lower bound shown using concepts from Algebraic Topology.

Closing this gap, as well as considering non geodesic input path changes that are useful to separate the performance and to evaluate consensus functions are **open questions**.

Also in the case of multi-valued consensus one may only require an output value that is within the range of values of the correct values, further exploring functions is also open, and we believe that it is fruitful field of research with application to several domains, including sensor activated devices, stable aggregation of distributed information and alike.

4 Complexity of Implementing Atomic Snapshots

A *snapshot* object consists of m components (shared variables), each storing a value. Processes can perform UPDATE operations to change the value of each individual component, and SCANS, each of which returns a consistent view of the contents of all the components. These operations can be performed simultaneously by different processes. Snapshots have been widely used to facilitate the design and verification of numerous distributed algorithms because they provide an immediate solution to the fundamental problem of calculating consistent views of shared variables; this happens while these variables may be concurrently updated by other processes.

A snapshot *implementation* from registers uses shared registers to simulate the snapshot components and provides algorithms for SCAN and UPDATE. Assuming that processes may fail by crashing, an implementation is *wait-free* if each nonfaulty process terminates executing a SCAN/UPDATE within a finite number of its own steps. An implementation is *linearizable* if (roughly speaking) the execution of a SCAN or an UPDATE operation in any execution of the implementation appears to take effect instantaneously.

Since snapshots have several applications, the design of efficient snapshot implementations is crucial. The *time complexity* of SCAN (UPDATE) of an implementation is the maximum number of steps executed by a process to perform a SCAN (UPDATE, respectively) in any execution of the implementation. The *time complexity* of the implementation is the maximum of the time complexities of its SCAN and UPDATE. Despite the numerous work that has been performed on designing efficient snapshot implementations (see [30] for a survey), their time complexity is not yet fully understood. Some implementations use a small number of registers but they have large time complexity while others employ more registers to achieve better time complexity.

It is known [27] that at least m registers are required to implement an m-component snapshot. An implementation that uses only m registers is provided in [1, 28]. Its time complexity is $O(mn)$ for both SCAN and UPDATE, where n is the number of processes in the system. A lower bound of $\Omega(mn)$ on the time complexity of SCAN for space-optimal implementations (that use only m registers), proved in [28], shows that this implementation is optimal. An implementation that uses n registers and has time complexity $O(n)$ for SCAN and $O(n \log n)$ for UPDATE (or vice versa) is provided by combining results in [2, 10, 42]. The fastest known implementation [8] has time complexity $O(n)$ for both SCAN and UPDATE and uses $O(n^2)$ registers. Another implementation with time complexity $O(n)$ which, however, uses an unbounded number of registers can be obtained by combining results in [2, 41]. Lower bounds on the space-time tradeoff are provided in [29], where it is proved that the time complexity of SCAN in any implementation that uses a fixed number of registers grows without bound as n increases.

Bridging the gap between the lower bounds provided in [29] and the best known upper bounds (discussed above) is a challenging **open problem**.

Even less is known for the time complexity of UPDATE. A lower bound of $\Omega(m)$ on the time complexity of UPDATE is proved in [7]. This lower bound extends a similar result presented in [5] for the weaker version of a *single-writer* snapshot (where each component can be updated by only one process associated to the component). Since the best known snapshot implementation [8] has time complexity $O(n)$ for UPDATE, it is unknown if this lower bound is optimal.

Proving better lower bounds for the time complexity of UPDATE or designing more efficient algorithms (in terms of the UPDATE time complexity) is an intriguing **open problem**.

The identification of tradeoffs between the number of registers used in an implementation, the time complexity of SCAN, and the time complexity of UPDATE is another interesting **open problem**.

The lower bounds proved in [28, 29] hold for deterministic algorithms and they can be possibly beaten by employing randomization. Some randomized implementations for the weaker version of single-writer snapshot objects are presented in [9]. Finding efficient randomized implementations for multi-writer snapshot objects remains a challenging **open problem**.

5 Pure Nash Equilibria in Selfish Routing

In modern non-cooperative networks, such as the Internet, participants, acting selfishly, wish to efficiently route their traffic from some source to some destination with the least possible delay. In such environments, *Nash Equilibria* [62, 63] represent steady states of the system where no user may profit by unilaterally changing its strategy.

Koutsoupias and Padadimitriou [47], formulated the study of selfish routing in non-cooperative networks by casting the problem as a non-cooperative game, known in the literature as the KP-model; n selfish users wish to route their unsplitable traffic onto m parallel links from a source to a destination. Each link has a certain capacity representing the rate at which the link processes traffic, and users have complete knowledge of the system's parameters such as the link capacities and the traffic of other users. Also, users choose how to route their traffic based on a common payoff function, which essentially captures the delay to be experienced on each link. However, modern non-cooperative systems present incomplete information on various aspects of their behavior. For example, it is often the case, that network users have incomplete information regarding the link capacities. Such uncertainty may be caused if the network links are complex paths created by routers which are constructed differently on separate occasions according to the presence of congestion or link failures.

Gairing *et al.* [32] were the first to consider an extension of the KP-model with incomplete information. Their model considers a game of parallel links with incomplete information on the traffics of the users. The payoff functions employed by the users take into account probabilistic information on the user traffics. The authors show (along with other interesting results) that their model always admits a Pure Nash Equilibrium and propose a polynomial-time algorithm for computing such equilibria for some special cases.

In [38] an extension of the KP-model was introduced, where the network links may present a number of different capacities and each user's uncertainty about the capacity of the links, called *belief*, is modeled via a probability distribution over all the possibilities. It is assumed that the users may have different sources of information regarding the network and, therefore, take their probability distributions to be distinct from one another. This gives rise to a model with user-specific payoff functions, where each user uses its distinct probability distribution to take decisions as to how to route its traffic. In particular, the model is an instance of weighted congestion games with user-specific functions studied by Milchtaich [59].

The authors of [38], among other problems, studied the existence of Pure Nash Equilibria in this new model; they proposed Polynomial-time algorithms for computing pure Nash equilibria for some special cases and they showed that the negative results of [59], for the non-existence of pure Nash equilibria in the case of three users, do not apply to their model.

The problem of existence of pure Nash Equilibria for this new model in the general case is a non-trivial problem; as of this writing, it remains **open**.

Given the non-existence result for weighted congestion games with user specific payoff-functions [59], a natural step is to disprove the existence of pure Nash Equilibria for the new model described in [38].

It is conjectured that the model introduced in [38] always admits a Pure Nash equilibrium in general. Proving or disproving this conjecture is an interesting **open challenge**.

Work for answering this question has been carried out in various directions. In [33] it was shown that the game introduced in [38] is not an *ordinal potential game*, since it has been shown that the state space of an instance of the game contains a cycle. Therefore, potential functions [60], a powerful method for proving existence of Nash Equilibria, cannot be used for this model. Further attempts by the authors of [38], including applying graph-theoretic methods and inductive arguments have not been successful. The arguments end up failing mainly due to the arbitrary relation between the different user beliefs on the capacity of the network links.

Typically, simple counter-examples to the existence of pure Nash Equilibria considering a small number of resource (links) and users are used for such purposes (for example, in [59], the counter-example involves 3 users and 3 resources). This appears not to be the case for the new model: in [38] was shown that for the case of three users (and arbitrary number of links) pure Nash Equilibria always exist; also simulations ran on numerous instances of the model (dealing with small number of users and links) suggest the existence of pure NE.

6 Adverse Cooperative Computing

The problem of cooperatively performing a collection of tasks in a decentralized setting where the computing medium is subject to adversarial perturbations is one of the fundamental problems in distributed computing. Such perturbations can be caused by processor failures, unpredictable delays, and communication breakdowns. To develop efficient solutions for computation problems ranging from distributed search such as SETI@home to parallel simulation and GRID computing, it is important to understand efficiency trade-offs characterizing the ability of p processors to cooperate on t independent tasks in the presence of adversity. This basic problem of cooperation has been studied in a variety of models, including shared-memory [3, 40, 43, 45, 58], message-passing [18, 19, 21, 26, 34, 48], in partitionable networks [25, 37, 39], and also in the settings with limited communication, e.g., [36, 57, 65]. Developing efficient algorithms solving such task-performing problems in adversarial settings has proven to be difficult.

Here we tackle the problem of distributed cooperation in deterministic shared-memory settings where the processors are subject to arbitrary failures and delays. Kanellakis and Shvartsman [43] introduced and studied an abstraction of this problem, called Write-All, formulated in terms of p processors writing to t distinct memory locations in the presence of an adaptive adversary that introduces dynamic failures or delays. Here writing to a memory location models an independent task that can be performed by a single processor in constant time. The efficiency of algorithms in such settings is measured in terms of *work* that accounts for all steps taken by the processors in solving the problem. The upper bound for Write-All with synchronous crash-prone processors was shown to be $O(t + p \log^2 t / \log \log t)$, where $p \leq t$, and at least one processor is non-faulty. The algorithm exhibiting this bound has optimal work of $O(t)$ when $p \leq t \log \log t / \log^2 t$. However, Kedem, Palem, Raghunathan, and Spirakis [45] showed that when $p = t$, the work lower bound for Write-All is $\Omega(t \log t)$, thus no optimal algorithm for Write-All exists for the full range of processors ($p = t$). Although a small gap of $\log t / \log \log t$ remains between the upper and lower bounds, the problem can be considered substantially solved for synchronous processors.

Solutions for the Write-All problem become significantly more challenging when asynchrony is introduced. The most efficient deterministic asynchronous algorithm known for Write-All is the elegant algorithm of Anderson and Woll [3] that has work $O(t \cdot p^{\varepsilon})$ for $p \leq t$ and any $\varepsilon > 0$. The strongest corresponding lower bound, due to Buss, Kanellakis, Ragde, and Shvartsman [17], is $\Omega(t + p \log p)$, and it holds even if no processor crashes. Note that in complexity-theoretic terms, the relative gap between these bounds on work is very large (i.e., polynomial in p, being p^{ε} for $p = t$), since the lower bound is only a logarithm away from linear work. Given that this gap is now 15 years old, and that this problem continues to be of interest, it appears that narrowing this gap is extremely challenging.

Thus we formulate our first, two-pronged, **open problem** as follows: (a) can a stronger than $\Omega(t \log t)$ lower bound on work be shown for asynchronus Write-All problem, and/or (b) is there an algorithm for asynchronous processors that solves the problem with work asymptotically less than $O(t^{1+\varepsilon})$ for $p = t$?

Next observe that an optimal algorithm for Write-All must have work $\Theta(t)$, however the lower bounds on work of $\Omega(t + p \log p)$ make optimality out of reach when $p = \Omega(t)$. Also note that the algorithm [3] has work complexity $\omega(t)$ for all but a trivial number p of processors. The quest then is to obtain work-optimal solutions for this problem using the largest possible, and non-trivial compared to t, number of processors p in order to maximize the parallelism of the solution. Recently Malewicz [56] presented the first qualitative advancement in the search for optimal work complexity by exhibiting an algorithm that has work $\Theta(t)$ using a non-trivial number g of processors, where $g = \sqrt[4]{t / \log t}$. Using different techniques, Kowalski and Shvartsman [49] exhibited an algorithm that has work complexity of $O(t + p^{2+\varepsilon})$, achieving optimality for a larger range of processors, specifically for $p = O(t^{1/(2+\varepsilon)})$.

Our second and final **open problem** is as follows: Is it possible to solve the asynchronous Write-All problem with optimal work $O(t)$ using the number of processors $p = t^{\delta}$ for $\delta > 1/2$?

Summing up, we presented the Write-All problem that abstracts the distributed cooperation problem in the presence of adversity. Despite substantial research, there is a dearth of efficient deterministic shared-memory algorithms for Write-All with asynchronous crash-prone processors.

The most challenging **open problems** in this area deal with closing the gap between the lower and upper bounds on work, and with the development of work-optimal algorithms that use the largest possible number of processors in order to achieve high speedup in solving the problem of distributed cooperation.

7 Distributed Approximations

Initiated by Papadimitriou and Yannakakis [64], the distributed approximation of linear programs has attracted the interests of researchers for some time. Most of the past efforts considered the special class of packing and its dual (covering) linear programs. Note that many hard problems (e.g. dominating set, coloring etc.) can be cast in the form of Integer Linear Programs (ILPs) and their distributed complexity of a good approximation is, up to now, a major open issue.

It is fair to assume a setting of a network with classical message passing capability, in which a node can send a message of size $O(\log n)$ bits to each neighbor in the net, in each communication step. Here n is the network size. We can also assume that each network node has a distinct id of size $O(\log n)$ bits. This is a synchronous communication model where the computation is assumed to advance in (global) rounds. Imagine then a general ILP setting, where, for example, there are n "producing" nodes and m "accepting" nodes. In general m is less than n. Each producer, call her i, has to derive an integer x_i (this can be negative. In that case the producer demands x_i units). For each "accepting" node j, when an x_i arrives to it, it has an associated cost (or benefit) $a_i^j * x_i$. For each accepting node j, the sum of all $a_i^j * x_i$, $(i = 1 \dots n)$, must be at most an integer quantity b_j. Here a_j^i, b_j are integers. The x_i's produced have each a cost (or benefit) $c_i * x_i$ where c_i is an integer. Now, the whole system must minimize the sum of all $c_i * x_i$, $(i = 1 \dots n)$. Or, at least approximate this minimum. The reader can recognize that this is the general case of an ILP.

The distributed complexity (i.e. number of rounds to achieve a good approximation) of the general integer-linear programs is a major **open problem**.

Note that the "accepting" nodes can elect a leader and then she can get all the coefficients needed to solve the problem locally. But, the restriction in the message size, leads to an awful number of rounds, e.g. around n. We want here a small number of rounds (constant number would be fine). Note that we do not assume that the a^i_j form a metric. They can also be arbitrarily large.

Facility location is an example of a non-positive linear integer program that is not a covering or packing one. Only very recently, the works [61, 35] made some progress in the distributed approximation of the facility location problem. In the facility location problem, the network is a (usually complete but this does not help much) bipartite graph, where two node sets, C and F share edges between them. Here C is the set of clients and F is the set of facilities. Each facility i has a non-negative opening cost f_i. The connection cost between facility i and client j is an integer c^j_i. Let y_i, x^j_i be zero/one variables where $y_i = 1$ indicates that facility i is open and $x^j_i = 1$ indicates that client j is connected to facility i. The system has to minimize the sum of all opening and connection costs. Note that any good solution that works with the relaxed linear problem first, must round the non-integer solutions. Even this (e.g. randomized rounding) has to be well done in a distributed way.

An interesting variation of such problems is a selfish distributed optimization situation. Let me motivate this by a Stackelberg game: Here each node i wants to send flow x_i to a destination node j. The total flow sums to (say) a value r. Some of the nodes (belonging to a subset L) are not selfish but they agree to work together (e.g. under an elected leader). The rest route their flows selfishly to avoid big delays. But the nodes in L can put their flows in such a way so that the Nash Equilibrium reached by the other nodes is very close to an optimal flow routing (e.g. of min total latency).

Finding distributed solutions to the selfish optimization problem described above, remains an **open problem**

For centralized solutions, one can see [44]. In fact, [44] shows that the leader (i.e., the centralized equivalent to L) can put its flow in such a way so that the Nash equilibrium reached by the other selfish flows is indeed the optimal, provided that the leader controls a sizeable portion of the overall flow. For recent developments on distributed approximations to problems related to linear and integer programming, we refer the reader to [51].

8 Sensor Networks: Locality for Geometric Graphs

Wireless sensor networks currently exhibit an incredible research momentum. Computer scientists and engineers from all flavors are embracing the area. Sensor networks are explored by researchers from hardware technology to operating systems, from antenna design to middleware, from graph theory to computational geometry. The distributed algorithms community should join this big interdisciplinary party.

In the last twenty years, so-called *local* algorithms have been a thriving theoretical research subject. In a *k-local algorithm*, for some parameter *k*, each node can communicate at most *k* times with its neighbors. Hence, even in synchronous operation nodes can at most gather information about their *k*-neighborhood. Early work for this model includes some of the most wonderful results in distributed computing, such as Luby's randomized independent set algorithm [55], sparse partitions by Awerbuch and Peleg [12], or Linial's $\Omega(\log n)$ lower bound [53].

There have been recent advances on both upper [51] and lower [50] bounds; however, many basic questions (e.g. a deterministic local construction of a maximal independent set) are still **open** (see also Section 7).

Until recently this theory was a bit *l'art pour l'art*. Sensor networks may be a first real-world application domain for local algorithms. Due to their wireless nature, the links of a sensor network are often unstable, in other words, the network is dynamic. In such an environment it is often impossible to run a centralized algorithm, as the network topology which serves as an input for an algorithm is usually different from the topology after running the algorithm. Local algorithms on the other hand are able to compromise approximation quality (efficacy) for communication time (efficiency) in order to keep up with the network dynamics.

Unfortunately, local algorithms are not exactly tailored for sensor networks. Apart from various other modeling issues [67], local algorithms are often developed for general graphs; in sensor networks, however, geometry comes into play as the distribution of nodes in space and the propagation range of wireless links usually adhere to geometric constraints. Several models inspired by both graph theory and geometry are possible; in a recent survey [67], a few such models are presented.

So far, very little is known about local algorithms for geometric graphs. To give a specific example, even for a simple model known as *unit disk graph*, the local complexity of typical coordination tasks (e.g., computing a dominating set) is an **open problem**. In fact, to the best of our knowledge, the currently best algorithm for this problem on unit disk graphs remains an algorithm for general graphs [51].

References

[1] Y. Afek, H. Attiya, D. Dolev, E. Gafni, M. Merritt and N. Shavit, "Atomic Snapshots of Shared Memory", *Journal of the ACM*, Vol 40, No 4, pp. 873–890, September 1993.

[2] J. H. Anderson, “Multi-Writer Composite Registers”, *Distributed Computing,* Vol. 7, No 4, pp. 175–195, April 1994.

[3] R. J. Anderson and H. Woll, “Algorithms for the Certified Write-All Problem”, *SIAM Journal on Computing,* Vol. 26, No. 5, pp. 1277–1283, October 1997.

[4] J. Aspnes, “Time- and Space-Efficient Randomized Consensus”, *Journal of Algorithms,* Vol. 14, No. 2, pp. 414-431, May 1993.

[5] A. Israeli and A. Shirazi, “The Time Complexity of Updating Snapshot Memories”, *Information Processing Letters,* Vol. 65, No. 1, pp. 33–40, January 1998.

[6] J. Aspnes, “Lower Bounds for Distributed Coin-Flipping and Randomized Consensus”, *Journal of the ACM,* Vol. 45, No. 3, pp. 415-450, May 1998.

[7] H. Attiya, F. Ellen and P. Fatourou, “The Complexity of Updating Multi-Writer Snapshot Objects”, *Proceedings of the 8th International Conference on Distributed Computing and Networking,* to appear, December 2006.

[8] H. Attiya and A. Fouren, “Adaptive and Efficient Algorithms for Lattice Agreement and Renaming”, *SIAM Journal on Computing,* Vol. 31, No. 2, pp. 642–664, October 2001.

[9] H. Attiya, M. Herlihy and O. Rachman, “Atomic Snapshots Using Lattice Agreement”, *Distributed Computing,* Vol. 8, No. 3, pp. 121–132, March 1995.

[10] H. Attiya and O. Rachman, “Atomic Snapshots in $O(n \log n)$ Operations”, *SIAM Journal on Computing,* Vol. 27, No. 2, pp. 319–340, April 1998.

[11] Y. Aumann, “Efficient Asynchronous Consensus with the Weak Adversary Scheduler”, *Proceedings of the 16th Annual ACM Symposium on Principles of Distributed Computing*, pp. 209-218, August 1997.

[12] B. Awerbuch and D. Peleg, “Sparse Partitions”, *Proceedings of the 31st Annual IEEE Symposium on Foundations of Computer Science,* Vol. 2, pp. 503–513, October 1990.

[13] Y. Azar, E. Cohen, A. Fiat, H. Kaplan and H. Racke, “Optimal Oblivious Routing in Polynomial Time”, *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pp. 383–388, June 2003.

[14] G. Bracha and O. Rachman, “Randomized Consensus in Expected $O(n^2 \log n)$ Operations”, *Proceedings of the 5th International Workshop on Distributed Algorithms,* pp. 143-150, October 1991.

[15] C. Busch, M. Ismail and J. Xi, “Optimal Oblivious Path Selection on the Mesh”, *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium,* pp. 82–91, April 2005.

[16] C. Busch, M. Ismail and J. Xi, “Oblivious Routing on Geometric Networks”, *Proceedings of the 17th Annual ACM Symposium on Parallelism in Algorithms and Architectures*, pp. 316–324, July 2005.

[17] J. Buss, P. Kanellakis, P. Ragde and A. Shvartsman, "Parallel Algorithms with Processor Failures and Delays", *Journal of Algorithms,* Vol. 20, No. 1, pp. 45–86, January 1996.

[18] B. Chlebus, R. De Prisco and A. Shvartsman, "Performing Tasks on Restartable Message-Passing Processors", *Distributed Computing,* Vol. 14, No. 1, pp. 49–64, January 2001.

[19] B. Chlebus, L. Gąsieniec, D. Kowalski and A. Shvartsman, "Bounding Work *and* Communication in Robust Cooperative Computation", *Proceedings of the 16th International Symposium on Distributed Computing,* pp. 295–310, October 2002.

[20] L. Davidovitch, S. Dolev and S. Rajsbaum, "Consensus Continue? Stability of Multi-Valued Continuous Consensus!", *Proceedings of the 6th Workshop on Geometric and Topological Methods in Concurrency and Distributed Computing,* pp. 21-24, October 2004.

[21] R. De Prisco, A. Mayer and M. Yung, "Time-Optimal Message-Efficient Work Performance in the Presence of Faults", *Proceedings of the 13th Annual ACM Symposium on Principles of Distributed Computing,* pp. 161–172, August 1994.

[22] S. Dolev, *Self-Stabilization,* MIT Press, 2000.

[23] S. Dolev, R. Kat and E. Schiller, "When Consensus Meets Self-Stabilization, Self-Stabilizing Failure Detector, Consensus and Replicated State-Machine", Technical Report, Department of Computer Science, Ben-Gurion University of the Negev, 2006.

[24] S. Dolev and S. Rajsbaum, "Stability of Long-Lived Consensus", *Journal of Computer and System Sciences,* Vol. 67, No. 1, pp. 26-45, August 2003.

[25] S. Dolev, R. Segala and A. Shvartsman, "Dynamic Load Balancing with Group Communication", *Proceedings of the 6th International Colloquium on Structural Information and Communication Complexity,* pp. 111–125, July 1999.

[26] C. Dwork, J. Halpern and O. Waarts, "Performing Work Efficiently in the Presence of Faults", *SIAM Journal on Computing,* Vol. 27, No. 5, pp. 1457–1491, October 1998.

[27] P. Fatourou, F. Fich and E. Ruppert, "Space-Optimal Multi-Writer Snapshot Objects are Slow", *Proceedings of the 21st Annual ACM Symposium on Principles of Distributed Computing,* pp. 13–20, July 2002.

[28] P. Fatourou, F. Fich and E. Ruppert, "A Tight Time Lower Bound for Space-Optimal Implementations of Multi-Writer Snapshots", *Proceedings of the 35th Annual ACM Symposium on Theory of Computing,* pp. 259–268, June 2003.

[29] P. Fatourou, F. Fich and E. Ruppert, "Time-Space Tradeoffs for Implementations of Snapshots", *Proceedings of the 38th Annual ACM Symposium on Theory of Computing,* pp. 169-178, May 2006.

[30] F. Fich, "How Hard is it To Take a Snapshot?", *Proceedings of the 31st Annual Conference on Current Trends in Theory and Practice of Informatics*, Vol. 3381, pp. 27–35, January 2005.

[31] M. J. Fischer, N. A. Lynch and M. S. Paterson, "Impossibility of Distributed Consensus with One Faulty Process", *Journal of the ACM*, Vol. 32, No. 3, pp. 374-382, April 1985.

[32] M. Gairing, B. Monien and K. Tiemann, "Selfish Routing with Incomplete Information", *Proceedings of the 17th Annual ACM Symposium on Parallelism in Algorithms and Architectures*, pp. 203–212, July 2005.

[33] M. Gairing, B. Monien and K. Tiemann, "Routing (Un-)Splittable Flow in Games with Player-Specific Linear Latency Functions", *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming*, Vol. 4051, pp. 501-512, July 2006.

[34] Z. Galil, A. Mayer and M. Yung, "Resolving Message Complexity of Byzantine Agreement and Beyond", *Proceedings of the 36th IEEE Symposium on Foundations of Computer Science*, pp. 724–733, October 1995.

[35] J. Gehweiler, C. Lammersen and C. Sohler, "A Distributed $O(1)$-Approximation Algorithm for the Uniform Facility Location Problem", *Proceedings of the 18th Annual ACM Symposium on Parallelism in Algorithms and Architectures*, pp. 237-243, July 2006.

[36] S. Georgiades, M. Mavronicolas and P. Spirakis, "Optimal, Distributed Decision-Making: The Case of no Communication", *Proceedings of the 12th International Symposium on Fundamentals of Computation Theory*, Vol. 1684, pp. 293–303, August/September 1999.

[37] C. Georgiou, A. Russell and A. Shvartsman, "Work-Competitive Scheduling for Cooperative Computing with Dynamic Groups", *SIAM Journal on Computing*, Vol. 34, No. 4, pp. 848–862, 2005.

[38] C. Georgiou, T. Pavlides and A. Philippou, "Network Uncertainty in Selfish Routing", *CD-ROM Proceedings of the 20th IEEE International Parallel and Distributed Processing Symposium*, April 2006.

[39] C. Georgiou and A. Shvartsman, "Cooperative Computing with Fragmentable and Mergeable Groups", *Journal of Discrete Algorithms*, Vol. 1, No. 2, pp. 211–235, April 2003.

[40] J. Groote, W. Hesselink, S. Mauw and R. Vermeulen, "An Algorithm for the Asynchronous Write-All Problem Based on Process Collision", *Distributed Computing*, Vol. 14, No. 2, pp. 75–81, April 2001.

[41] M. Inoue, W. Chen, T. Masuzawa and N. Tokura, "Linear Time Snapshots Using Multi-Writer Multi-Reader Registers", *Proceedings of the 8th International Workshop on Distributed Algorithms*, Vol. 857, pp. 130–140, September/October 1994.

[42] A. Israeli, A. Shaham, A. Shirazi and T. Masuzawa, “Linear-Time Snapshot Implementations in Unbalanced Systems”, *Mathematical Systems Theory*, Vol. 28, No. 5, pp. 469–486, September/October 1995.

[43] P. Kanellakis and A Shvartsman, *Fault-Tolerant Parallel Computation*, Kluwer Academic Publishers, 1997.

[44] A. Kaporis and P. Spirakis, “The Price of Optimum in Stackelberg Games on Arbitrary Single Commodity Networks and Latency Functions”, *Proceedings of the 18th Annual ACM Symposium on Parallelism in Algorithms and Architectures*, pp. 19-28, July 2006.

[45] Z. Kedem, K. Palem, A. Raghunathan and P. Spirakis, “Combining Tentative and Definite Executions for Dependable Parallel Computing”, *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, pp. 381–390, May 1991.

[46] Z. Kedem, K. Palem and P. Spirakis, “Efficient Robust Parallel Computations”, *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, pp. 138–148, May 1990.

[47] E. Koutsoupias and C. H. Papadimitriou, “Worst-Case Equilibria”, *Proccedings of the 16th International Symposium on Theoretical Aspects of Computer Science*, Vol. 1563, pp. 404–413, March 1999.

[48] D. Kowalski and A. Shvartsman, “Performing Work with Asynchronous Processors: Message-Delay-Sensitive Bounds”, *Information and Computation*, Vol. 203, No. 2, pp. 181–210, December 2005.

[49] D. Kowalski and A. Shvartsman, “Writing-All Deterministically and Optimally Using a Non-Trivial Number of Asynchronous Processors”, *Procedings of the 16th Annual ACM Symposium on Parallelism in Algorithms and Architectures*, pp. 311–320, June 2004.

[50] F. Kuhn, T. Moscibroda and R. Wattenhofer, “What Cannot be Computed Locally”, *Proceedings of the 23rd Annual ACM Symposium on the Principles of Distributed Computing*, pp. 300–309, July 2004.

[51] F. Kuhn, T. Moscibroda and R. Wattenhofer, “The Price of Being Near-Sighted”, *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 980-989, January 2006.

[52] F. Leighton, B. Maggs and S. Rao, “Packet Routing and Job-Shop Scheduling in $O(Congestion + Dilation)$ Steps”, *Combinatorica*, Vol. 14, No. 2, pp. 167–186, June 1994.

[53] N. Linial, “Locality in Distributed Graph Algorithms”, *SIAM Journal on Computing*, Vol. 21, No. 1, pp. 193–201, February 1992.

[54] M. Loui and H. Abu-Amara, “Memory Requirements for Agreement among Unreliable Asynchronous Processes”, *Advances in Computing Research*, Vol. 4, pp. 163-183, 1987.

[55] M. Luby, "A Simple Parallel Algorithm for the Maximal Independent Set Problem", *SIAM Journal on Computing*, Vol. 15, No. 4, pp. 1036-1053, November 1986.

[56] G. Malewicz, "A Work-Optimal Deterministic Algorithm for the Certified Write-All Problem with a Nontrivial Number of Asynchronous Processors", *SIAM Journal on Computing*, Vol. 34, No. 4, pp. 993–1024, April/May 2005.

[57] G. Malewicz, A. Russell and A. Shvartsman, "Distributed Scheduling for Disconnected Cooperation", *Distributed Computing*, Vol. 18, No. 6, pp. 409–420, June 2006.

[58] C. Martel, A. Park and R. Subramonian, "Work-Optimal Asynchronous Algorithms for Shared Memory Parallel Computers", *SIAM Journal on Computing*, Vol. 21, No. 6, pp. 1070–1099, December 1992.

[59] I. Milchtaich, "Congestion Games with Player-Specific Payoff Functions", *Games and Economic Behavior*, Vol. 13, No. 1, pp. 111–124, April 1996.

[60] D. Monderer and L. S. Shapley, "Potential Games", *Games and Economic Behavior*, Vol. 14, No. 1, pp. 124–143, May 1996.

[61] T. Moscibroda and R. Wattenhofer, "Facility Location: Distributed Approximation", *Proceedings of the 24th Annual ACM Symposium on the Principles of Distributed Computing*, pp. 108-117, July 2005.

[62] J. F. Nash, "Equilibrium Points in *n*-Person Games", *Proceedings of the National Acanemy of Sciences of the United States of America*, Vol. 36, pp. 48–49, 1950.

[63] J. F. Nash, "Non-Cooperative Games", *Annals of Mathematics*, Vol. 54, No. 2, pp. 286–295, 1951.

[64] C. Papadimitriou and M. Yannakakis, "Linear Programming Without the Matrix", *Proceedings of the 25th Annual ACM Sumposium on Theory of Computing*, pp. 121-129, May 1993.

[65] C. Papadimitriou and M. Yannakakis, "On the Value of Information in Distributed Decision-Making", *Proceedings of the 10th Annual ACM Symposium on Principles of Distributed Computing*, pp. 61–64, August 1991.

[66] H. Racke, "Minimizing Congestion in General Networks", *Proceedings of the 43rd Annual Symposium on the Foundations of Computer Science*, pp. 43–52, November 2002.

[67] S. Schmid and R. Wattenhofer, "Algorithmic Models for Sensor Networks", *Proceedings of the 14th International Workshop on Parallel and Distributed Real-Time Systems*, April 2006.

The Formal Language Theory Column

by

Arto Salomaa
Turku Centre for Computer Science
University of Turku
Joukahaisenkatu 3–5 B, 20520 Turku, Finland
asalomaa@utu.fi

The ultimate equivalence problem for uniform HDT0L systems

Juha Honkala
Department of Mathematics
University of Turku
FIN-20014 Turku, Finland
juha.honkala@utu.fi

Abstract

We discuss the ultimate equivalence problem for uniform HDT0L systems and give bounds for the problem.

1 Introduction

Culik II has proved the decidability of the ultimate sequence equivalence problem for D0L systems, [1]. The approach of Culik II uses heavily the balance properties of sequence equivalent D0L systems. For interesting results concerning the ultimate equivalence problem of HD0L systems see [6].

The ultimate equivalence problem remains open for HDT0L systems. In this note we prove that the problem is decidable for compatible uniform HDT0L systems.

Bulletin of the EATCS no 90, pp. 127–130, October 2006

For the basics concerning HDT0L systems we refer to [4, 5, 2].

2 Definitions and results

We use standard language-theoretic notation and terminology. In particular, the *length* of a word w is denoted by $|w|$ and a morphism $g : X^* \longrightarrow Y^*$ is called *uniform* if $|g(a)| = |g(b)|$ for all $a, b \in X$.

An *HDT0L system* is a construct $G = (X, Y, g_1, \ldots, g_n, g, w)$, where X and Y are finite alphabets, $n \geq 1$ is an integer, $g_i : X^* \longrightarrow X^*$, $1 \leq i \leq n$, and $g : X^* \longrightarrow Y^*$ are morphisms and $w \in X^*$ is a word. G is called *uniform* if $g_1, \ldots, g_n, g$ are uniform morphisms.

Let $G = (X_1, Y, g_1, \ldots, g_n, g, w_1)$ and $H = (X_2, Y, h_1, \ldots, h_n, h, w_2)$ be HDT0L systems. Then G and H are called *ultimately sequence equivalent* if there is a nonnegative integer k_0 such that

$$gg_{i_k} \ldots g_{i_1}(w_1) = hh_{i_k} \ldots h_{i_1}(w_2) \tag{1}$$

whenever $k \geq k_0$ and $i_1, \ldots, i_k \in \{1, \ldots, n\}$. G and H are called *sequence equivalent* if (1) holds for all $k \geq 0$ and $i_1, \ldots, i_n \in \{1, \ldots, n\}$.

Let $G = (X_1, Y, g_1, \ldots, g_n, g, w_1)$ and $H = (X_2, Y, h_1, \ldots, h_n, h, w_2)$ be uniform HDT0L systems. Then G and H are called *compatible* if $|w_1| = |w_2| \geq 1$, $|g_i(x_1)| = |h_i(x_2)| \geq 1$ and $|g(x_1)| = |h(x_2)| \geq 1$ whenever $i = 1, \ldots, n$, $x_1 \in X_1$, $x_2 \in X_2$.

Theorem 1. *Assume that uniform HDT0L systems $G = (X_1, Y, g_1, \ldots, g_n, g, w_1)$ and $H = (X_2, Y, h_1, \ldots, h_n, h, w_2)$ are compatible. Then G and H are ultimately sequence equivalent if and only if*

$$gg_{i_k} \ldots g_{i_1}(w_1) = hh_{i_k} \ldots h_{i_1}(w_2) \tag{2}$$

whenever $k \geq card(X_1) + card(X_2) - 1$ and $1 \leq i_1, \ldots, i_k \leq n$.

Proof. Without loss of generality assume that $X_1 \cap X_2 = \emptyset$. Let $X = X_1 \cup X_2$ and define the morphisms $f_i : X^* \longrightarrow X^*$, $1 \leq i \leq n$, and $f : X^* \longrightarrow Y^*$ by

$$f_i(x) = \begin{cases} g_i(x) & \text{if } x \in X_1 \\ h_i(x) & \text{if } x \in X_2 \end{cases}$$

and

$$f(x) = \begin{cases} g(x) & \text{if } x \in X_1 \\ h(x) & \text{if } x \in X_2 \end{cases}.$$

Because G and H are compatible, the morphisms f_i, $1 \leq i \leq n$, and f are uniform. If $k \geq 0$ and $1 \leq i_1, \ldots, i_k \leq n$, then (2) holds if and only if

$$ff_{i_k} \ldots f_{i_1}(w_1) = ff_{i_k} \ldots f_{i_1}(w_2).$$

Next, assume that p is a nonnegative integer and define the equivalence relation R_p on the set X^* as follows. If $u, v \in X^*$, then uR_pv holds if and only if we have

$$f f_{i_k} \dots f_{i_1}(u) = f f_{i_k} \dots f_{i_1}(v)$$

whenever $k \geq p$ and $1 \leq i_1, \dots, i_k \leq n$. Clearly,

$$R_0 \subseteq R_1 \subseteq R_2 \subseteq \dots .$$

Observe that uR_pv implies $|u| = |v|$. Furthermore, if $u = a_1 \dots a_s$ and $v = b_1 \dots b_s$ where $a_j, b_j \in X$, $1 \leq j \leq s$, then uR_pv if and only if $a_jR_pb_j$ for all $j \in \{1, \dots, s\}$.

Assume that $R_p = R_{p+1}$. We claim that then $R_{p+1} = R_{p+2}$. To see this, it suffices to show that $R_{p+2} \subseteq R_{p+1}$. Assume that $uR_{p+2}v$ where $u, v \in X^*$. Then $f_i(u)R_{p+1}f_i(v)$ for all $i \in \{1, \dots, n\}$. Hence $f_i(u)R_pf_i(v)$ for all i, which implies that $uR_{p+1}v$. This concludes the proof that $R_p = R_{p+1}$ implies $R_{p+1} = R_{p+2}$ and, hence, that $R_p = R_{p+j}$ for all $j \geq 0$.

Next, let r_p be the number of equivalence classes of $R_p \cap (X \times X)$. Then

$$\text{card}(X_1) + \text{card}(X_2) \geq r_0 \geq r_1 \geq \dots .$$

Consequently, there exists an integer $t \leq \text{card}(X_1) + \text{card}(X_2) - 1$ such that

$$r_t = r_{t+1}.$$

This implies

$$R_t \cap (X \times X) = R_{t+1} \cap (X \times X).$$

Hence $R_t = R_{t+1}$.

Assume now that G and H are ultimately sequence equivalent. Then there exists a nonnegative integer p such that $w_1R_pw_2$. Hence $w_1R_tw_2$. In other words, (2) holds whenever $k \geq \text{card}(X_1) + \text{card}(X_2) - 1$ and $1 \leq i_1, \dots, i_k \leq n$. □

To continue recall the following result from [3].

Theorem 2. *Suppose $G = (X_1, Y, g_1, \dots, g_n, g, w_1)$ and $H = (X_2, Y, h_1, \dots, h_n, h, w_2)$ are uniform HDT0L systems such that $|w_1| = |w_2|$. Then G and H are sequence equivalent if and only if we have*

$$gg_{i_k} \dots g_{i_1}(w_1) = hh_{i_k} \dots h_{i_1}(w_2)$$

whenever $0 \leq k \leq \max\{1, card(X_1) + card(X_2) - 2\}$ and $1 \leq i_1, \dots, i_k \leq n$.

Theorems 1 and 2 imply our final result.

Theorem 3. *Assume that uniform HDT0L systems* $G = (X_1, Y, g_1, \ldots, g_n, g, w_1)$ *and* $H = (X_2, Y, h_1, \ldots, h_n, h, w_2)$ *are compatible. Then G and H are ultimately sequence equivalent if and only if*

$$gg_{i_k} \ldots g_{i_1}(w_1) = hh_{i_k} \ldots h_{i_1}(w_2)$$

whenever $card(X_1) + card(X_2) - 1 \leq k \leq 2card(X_1) + 2card(X_2) - 3$ *and* $1 \leq i_1, \ldots, i_k \leq n$.

References

[1] K. Culik II, The ultimate equivalence problem for D0L systems, Acta Inform. 10 (1978) 79-84.

[2] J. Honkala, A short solution for the HDT0L sequence equivalence problem, Theoret. Comput. Sci. 244 (2000) 267-270.

[3] J. Honkala, A note on uniform HDT0L systems, Bulletin of EATCS 75 (2001) 220-223.

[4] G. Rozenberg and A. Salomaa, The Mathematical Theory of L Systems, Academic Press, New York, 1980.

[5] G. Rozenberg and A. Salomaa (Eds.), Handbook of Formal Languages, Vol. 1, Springer, Berlin, 1997.

[6] K. Ruohonen, On some decidability problems for HD0L systems with nonsingular Parikh matrices, Theoret. Comput. Sci. 9 (1979) 377-384.

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

Review of Dines Bjørner's textbooks Software Engineering 1–3

by Hartmut Ehrig
Technische Universität Berlin, Germany

Dines Bjørner is certainly one of the most well-known protagonists of formal methods in software and system engineering. Already in the 70s he was one of the fathers of the Vienna Definition Method and Language advocating formal specifications for software development. More recently he has also been one initiator of the RAISE Method and the RAISE Specification Language RSL. In fact, RSL plays an important role in the 3 volumes Software Engineering 1-3 published as texts in Theoretical Computer Science by Springer in 2006.

Although there is already a large variety of text books on software engineering available, it is certainly a pleasure to read and study the 3 volumes by Dines Bjørner. All the relevant concepts from software engineering are systematically introduced and the role of formal methods is carefully explained. The reviewer shares the point of view of the author that formal techniques apply in all phases, stages and steps of software engineering, and in the development of all kinds of software. The main question, however, is how to present formal methods to people in software engineering, especially to those without solid background in mathematics. The answer given by the author is "Formal Techniques Light". This means basically 3 steps: “1. Start by being systematic. 2. Specify crucial facts formally. 3. Program Code from there.” The first 2 steps are presented carefully for a large variety of concepts and examples covered in the 3 volumes. For step 3, however, there is essentially only a hint that tools can be provided to translate model oriented specifications in RSL into constructs of programming languages like C, $C++$, $C\#$, Java, and Standard ML.

Let us have a closer look at each of the volumes. In volume 1 with subtitle “Abstraction and Modelling” we have in part I a careful introduction into the aims and objectives of the 3 volumes, in part II an informal, but systematic treatment of several areas of discrete mathematics, and in parts III and IV an introduction into RSL. Finally in part IV the concepts of applicative, imperative, and concurrent specification programming are carefully explained and shown how they are supported by RSL. RSL is justified as suitable specification language for this purpose, because it is close to discrete mathematics, allows to express the imperative specification style, can handle expressions of concurrency, has a strong flavour of algebraic specification languages and allows to structure its specifications in a modular fashion. Remarkable in part II is the kind of informal, but systematic treatment of sets, functions, algebra and logic, which almost avoids any mathematical definition. It is coherent with the standard definitions, except of the notion of injective function, which - surprisingly - is not required to be “one-to-one”, but “non-surjective”.

Volume 2 with subtitle SSpecification of Systems and Languages” starts in part I with an RSL primer summarizing all RSL-concepts introduced in volume 1, in part II and III specification facets like hierarchies and composition, denotations and computations, configuration, content and states as well as time and space are carefully introduced. Part IV concerns the linguistic concepts of syntax, semantics and pragmatics, which can be summarized as semiotics. Further specification techniques like modularization, automata and machines are presented in part VI and other specification techniques like Petri nets, message and live sequence charts, statecharts and quantitative models of time in part V. Finally interpreter and compiler definitions are discussed carefully in part VII including different kinds of simple applicative imperative, modular and parallel languages. The chapters on visual specification techniques in part VI are authored by Christian Krog Madsen, a former student of Dines Bjørner. Core concepts of these techniques are introduced on an informal level and then syntax, static and dynamic semantics are given in RSL. This is called “UML”-ising formal techniques.

The main question “What is Software Engineering?” is addressed in volume 3 with subtitle “Domains, Requirements, and Software Design”. The answer to this question is summarized by Dines Bjørner in his Triptych of Software Engineering”. The main idea of this triptych is that software development is an iterative process involving domain engineering, requirements engineering and software design, where all phases should be based on formal techniques. Bjørner claims - and the reviewer agrees in principle - that “developing software without formal techniques is like sailing the high seas without knowing how to compute the current longitude”. According to this triptych paradigm carefully discussed in part I, domain engineering, requirements engineering and computing systems design (including software design) is presented on a conceptual level with a large variety of examples and case studies specified in RSL in parts IV, V, and VI of volume 3 respectively. In the closing part VII it is a pleasure to read how Dines Bjorner is fighting against several myths about formal techniques of software engineering. In summary, the 3 volumes with altogether about 2250 pages - present a most interesting view of software engineering, which is certainly different from most other textbooks, especially from those focusing on implementation, testing and management issues within software engineering. A suitable subtitle for all 3 volumes could be “The role of formal methods in software development”. The choice of RSL as formal specification technique is consistent with the aims of the author, although other protagonists of formal methods in software engineering might prefer other specification or modelling techniques. Moreover, the strong claim of model-oriented software development is widely supported in software engineering these days, where in most cases; however, object-oriented software development based on modelling and metamodelling techniques in the sense of UML and other visual techniques is predominant.

From the formal methods point of view the reviewer agrees with Bjørner that the lack of formal semantics for main parts of UML is still unsatisfactory, but “UML”-ising formal techniques is only one alternative. The reviewer also agrees with the author that the major shortcoming of his 3 volumes is the “all too brief coverage of correctness issues”, where the reader is refered to other literature. Although we regret that the reader is not guided how the show correctness of stepwise refinement form requirements to design, the principle “Formal Techniques Light” mentioned above is a convincing paradigm for the approach in these 3 volumes.

Altogether the 3 volumes can be highly recommended for software engineers in practice and students in software engineering courses in order to learn the basic principles of software development and how they can be supported by formal methods in a systematic way.

The Logic in Computer Science Column

by

Yuri Gurevich

Microsoft Research
One Microsoft Way, Redmond WA 98052, USA
gurevich@microsoft.com

Embedded Finite Models

Leonid Libkin
University of Edinburgh & University of Toronto

Yuri asked me, the author (**A**) to meet his student Quisani (**Q**), who often appears in public just before a new issue of the *Bulletin* comes out, and for whom Yuri arranges meetings with computer science logicians.

As **Q** looked rather tired and suffering from a lack of sleep, I asked him what had caused it. He explained that in a recent meeting with Jan Van den Bussche, which was reported in this *Column* [38], he was given a chapter on embedded finite models from my book [29] as bedtime reading, but didn't find it very easy to start reading a 14-chapter book from chapter 13. So an email to Yuri followed, and a meeting with me was arranged. The following is my transcription of that meeting.

A. At the very least you're now familiar with the main definition of embedded finite models. Let's review it first.

Q. As I recall it, you start with an *infinite* model or structure, something like the real closed field $\mathfrak{R} = \langle \mathbb{R}, +, \cdot, 0, 1, < \rangle$, and then put a *finite* model on it, say, a finite graph whose nodes are real numbers.

A. That's right. Formally speaking, you have two vocabularies, say Ω for an infinite structure, and σ for a finite structure, and you look at (Ω, σ)-structures, where σ-relations are finite.

Bulletin of the EATCS no 90, pp. 134–149, October 2006

Q. So, for example, if I want to work with graphs whose nodes are real numbers, then Ω could be $(+, \cdot, 0, 1, <)$ and σ should have one binary relation $E(\cdot, \cdot)$ for the edges of my graphs.

A. Exactly. And you'll be working with logical formulae over both Ω and σ. So in first-order logic (which we abbreviate as FO), you can write a sentence

$$\exists a \exists b \forall x \forall y \; E(x, y) \rightarrow a \cdot x + b = y$$

saying that the graph lies on a line.

Q. Do you use special names to distinguish the Ω-structure and the σ-structure?

A. Yes, we usually refer to the Ω-structure as the *background* structure, and to finite σ-structures as *embedded finite models*. In our example, graphs are "embedded" into the real field $\Re$.

Q. Ok, I now remember the definition. But can you explain why anyone would study these objects?

A. Certainly. The initial motivation came from the field of database query languages.

Q. Yes, I heard from many people that databases provide much of the motivation for the development of finite model theory, but how do you come up with embedded finite models?

A. Simple. Do you remember what the main theoretical database query language is?

Q. Of course, it's relational calculus, which is just another name for FO.

A. Correct. For example, if you have a graph, you can ask for pairs of nodes (x, y) connected by a path of length 2 using the formula $\exists z \; (E(x, z) \wedge E(z, y))$, or for nodes x from which there is an edge to every other node: $\forall y \; E(x, y)$. And FO provides the basis of the most common real-life query language SQL.

Q. But we only store finite sets in databases, don't we?

A. Wait a minute. Much of database theory (say, as described in [1, 31]) concentrates on languages that operate with uninterpreted objects – in other words, it doesn't matter what those graph nodes are. But in real databases we operate with *interpreted* objects: say, numbers or strings. In fact, for every relation we put in a database, we must write a `create table` statement in SQL that specifies a type for each attribute: real, integer, Boolean, string, and so on.

Q. I think I see it: elements that we store in a database may come from an infinite set.

A. Not only that,but there are also some domain-specific operations, such as arithmetic operations for numerical domains, that we can use in queries.

Q. Can you give me an example?

A. Let's take a ternary relation $R(\cdot,\cdot,\cdot)$, whose tuples are interpreted as two city names and the distance between them. Then the query

$$\exists z, d_1, d_2\ (R(x,z,d_1) \wedge R(z,y,d_2) \wedge d_1 + d_2 < 100)$$

finds pairs of cities x and y so you can travel between them while visiting another city and the total traveled distance is less than 100.

Q. I remember now, Jan Van den Bussche [38] was talking about applications in Geographical Information Systems.

A. Yes, but this is not the only application. One can think of finite strings and various operations and relations on them, such as adding letters at either end of a string, or checking if one string is a prefix of another.

Q. I see. So, the background structure of vocabulary Ω provides information about the domain and operations on it, and the finite σ-structure is a "database" you put on the Ω-structure.

A. Yes. Note also that while Ω may contain function $(+,\cdot)$, relation $(<)$ and constant $(0,1)$ symbols, we assume that σ has only relation symbols in it.

Q. This is a rather natural setting. Didn't database people study it to death during the early days of database theory?

A. Not really – they were not that interested in interpreted operations in query languages (although they are present in all real-life languages). Even more importantly from the relational databases point of view, when one writes queries in logical form, one normally assumes that a database is a finite structure with a finite universe. This suffices for many – but not all – database applications (a notable exception is constraint databases, to be discussed shortly). The formal setting of relational databases, however, assumes an *infinite* domain of possible values, albeit without any operations on it. So technically speaking, relational databases are often defined as finite structures embedded into an infinite structure of the empty vocabulary. So in this case, a logical formalism would be that of an infinite structure with a finite structure embedded into it, rather than just a "stand-alone" finite structure.

Q. And no one was curious whether these two settings are different?

A. Some people did. For example, Paris Kanellakis in his survey of relational databases in the Handbook of TCS [25] mentions this distinction. But by that time, it was known that infinite domains without operations don't add anything to the "everything-is-finite" relational model [2, 24].

Q. How can you state this formally?

A. We'll get to it soon – this is done via *collapse theorems*. But let's first talk about a new direction in database research brought to the fore the issues related to infinite domains and interpreted operations – *constraint databases*. They were introduced in 1990 [26], and a book about them appeared ten years later [28].

Q. Yes, I heard about constraint databases from [38]: they are used to represent infinite sets in databases, right?

A. Right. In fact the model of constraint databases is very similar to embedded finite models: all that changes is the interpretation of σ-relations. Now they are not just finite sets, but sets *definable* (in FO) in the background structure.

Q. And what can we represent in this setting?

A. Let's look at the real field again. An FO formula over $\mathfrak{R} = \langle \mathbb{R}, +, \cdot, 0, 1, < \rangle$ with, say, two free variables $\varphi(x, y)$ defines a subset of the the plane $\mathbb{R}^2$ of points that satisfy the formula. Do you remember what these sets are called?

Q. I think it has something to do with algebra. And somehow the name Tarski also comes to mind.

A. Right, they are *semi-algebraic* sets [13, 39]. And by Tarski's quantifier-elimination for the real field, each FO formula $\varphi(\bar{x})$ over $\mathfrak{R}$ is equivalent to a *quantifier-free* formula, that is, just to a Boolean combination of polynomial inequalities $p(\bar{x}) > 0$.

Q. And I presume you can represent a lot of useful information about, say, geography, using such polynomial constraints.

A. True. So now if your query language is FO over the real field and database predicates – interpreted as semi-algebraic sets – you can ask many queries about your geographical objects, which now are *finitely* represented in your database by means of a set of polynomial constraints. An example would be the "database lies on a line" query, which was our first example.

Q. What types of interesting queries can you write in this language?

A. You can test, for example, if a set is topologically open or closed, if it is bounded; for a trajectory $T(x, y, t)$ you can compute the speed at each time t; you can compute the boundary of a set, compare coordinates of specific points, and so on – many queries one needs to ask in GISs.

This language, by the way, is often called FO + Poly (for first-order with polynomial constraints). In many applications even simpler linear constraints are used [20]; the corresponding query language of firsr-order with linear constraints, and database relations definable with linear constraints, is called FO + Lin.

Q. But now I recall that certain things you can*not* ask in FO + Lin and FO + Poly – and that's why Jan suggested I read the embedded finite models chapter in [29].

A. And do you remember an example of a query that FO + Poly cannot express?

Q. I think it was topological connectivity, wasn't it? But then what does it have to do with finite models?

A. It turns out that many questions about expressiveness of FO + Poly over semi-algebraic sets can be reduced to questions about its expressiveness over finite sets. For example, topological connectivity and graph connectivity are very closely related.

Q. I believe I see why: we can embed any graph into $\mathbb{R}^3$ without self-intersections. So a graph is connected iff its embedding is topologically connected!

A. Exactly. There is one little detail: to reduce non-expressibility of topological connectivity to non-expressibility of graph connectivity you must show that the embedding itself is definable in FO + Poly, but this is easily done.

Q. This is a nice example, but it's quite ad hoc. Is there a general result that describes what problems can be reduced to the finite case?

A. Not really – although it would be nice to have such a result – but there are plenty of examples. For instance, Grumbach and Su [21] showed how inexpressibility of many topological properties in FO + Poly can be reduced to questions about embedded finite models. And many other results about constraint databases are obtained by reduction to the finite case [28]. So one can say that embedded finite models play the same role for constraint databases as usual finite models play in relational database theory.

Q. I think we had quite a detour since I asked you about a general result saying that embedded finite models behave just like the usual finite models.

A. You're absolutely right, let's get back to it. As I said, these results come in the form of *collapse theorems*. But before we state them, we need some notations. Let's use FO($\mathfrak{M}, \sigma$) to denote first-order logic over the background Ω-structure $\mathfrak{M}$ and relational vocabulary σ – remember that now σ-relations are finite. For example, FO + Poly is just another name for FO($\mathfrak{R}, \sigma$). Now what would you call the "standard" finite model-theoretic FO using this notation?

Q. Perhaps FO($\mathfrak{M}_\emptyset, \sigma$) where $\mathfrak{M}_\emptyset$ is a structure with an empty vocabulary?

A. Almost, but not quite. The issue, again, is the underlying domain. Let's say $\mathfrak{M}_\emptyset = \langle U, \emptyset \rangle$. If you write $\exists x \varphi(x)$, what does it mean?

Q. I guess it means that there is a witness a for $\varphi(x)$.

A. Correct, but where does this witness come from?

Q. It must come from the universe of $\mathfrak{M}_\emptyset$, that is, from U.

A. And now we have a little problem. When we work with *finite* models, $\exists x$ means that we can find a witness in the universe of the finite model, that is, somewhere in the σ-structure.

Q. Can you explain why this is a problem?

A. Sure. Let's say σ is the vocabulary $E(\cdot,\cdot)$ of graphs, and we want to say that a graph is reflexive. How would you express this in FO?

Q. I think I see what you want to say. I would like to write $\forall x\, E(x,x)$, but that would mean that $E(a,a)$ is true for all $a \in U$, and hence this sentence is false in all finite graphs embedded in $\mathfrak{M}_\emptyset$.

A. Precisely. So we introduce a new type of quantification that only refers to the σ-structure.

One calls the set of all elements of a finite σ-structure $\mathcal{A}$ its *active domain*, and denotes it by *adom*($\mathcal{A}$). And now we introduce active-domain quantification $\exists x \in$ *adom* $\varphi(x)$ and $\forall x \in$ *adom* $\varphi(x)$ with the meaning that there exists an element (or for all elements) a of *adom*($\mathcal{A}$), the formula $\varphi(a)$ is true.

Q. Does it make a logic more expressive?

A. No, it doesn't, because the active domain itelf is easily expressible in FO: say, for graphs by a formula $\exists y\, (E(x,y) \vee E(y,x))$. But then we can define an interesting fragment of the logic FO($\mathfrak{M},\sigma$), namely its restriction in which all quantification is active-domain, that is, $\exists x \in$ *adom* φ or $\forall x \in$ *adom* φ. We shall denote it by $\mathrm{FO}_{\mathrm{act}}(\mathfrak{M},\sigma)$.

Q. I see – so now $\mathrm{FO}_{\mathrm{act}}(\mathfrak{M}_\emptyset,\sigma)$ is the real finite-model theoretic FO over σ-structures, for which the background structure doesn't matter, and we somehow want to reduce FO($\mathfrak{M},\sigma$) to $\mathrm{FO}_{\mathrm{act}}(\mathfrak{M}_\emptyset,\sigma)$.

A. Almost - but for reasons that will become clear soon, we can't completely eliminate everything from the vocabulary of the background structure, and we need to keep a linear ordering in it. So with each $\mathfrak{M} = \langle U, \Omega\rangle$ we associate $\mathfrak{M}_< = \langle U, <\rangle$, where $<$ is an arbitrary linear ordering (if $\mathfrak{M}$ had it to start with, we'll keep that ordering), and we shall attempt to reduce questions about FO($\mathfrak{M},\sigma$) to $\mathrm{FO}_{\mathrm{act}}(\mathfrak{M}_<,\sigma)$.

Q. But what do we know about $\mathrm{FO}_{\mathrm{act}}(\mathfrak{M}_<,\sigma)$?

A. Plenty, thanks to people working in finite model theory. This is just FO over σ-structures with a linear ordering on them. Many inexpressibility results in this setting are obtained by routine applications of Ehrenfeucht-Fraïssé games, and some heavy tools are available too: for example, the Grohe-Schwentick theorem [19] says that any property expressible in $\mathrm{FO}_{\mathrm{act}}(\mathfrak{M}_<,\sigma)$ that does not depend on a particular linear ordering $<$ is local, i.e., determined by the isomorphism type of a small neighborhood of free variables of a formula, and Shelah's theorem [37], which says that even though $\mathrm{FO}_{\mathrm{act}}(\mathfrak{M}_<,\sigma)$ does not have a 0-1 law, it has a very weak form of it, called the slow oscillation property.

Q. Ok, I am convinced we can use many facts about $\mathrm{FO}_{\mathrm{act}}(\mathfrak{M}_<, \sigma)$ "by citation". But how do we go from $\mathrm{FO}(\mathfrak{M}, \sigma)$ to it?

A. We do it in two steps: first we try to show that $\mathrm{FO}(\mathfrak{M}, \sigma) = \mathrm{FO}_{\mathrm{act}}(\mathfrak{M}, \sigma)$ – and this is called a *natural-active collapse*, because unrestricted quantification over $\mathfrak{M}$ is sometimes referred to as "natural" quantification. As the second step, we try to reduce $\mathrm{FO}_{\mathrm{act}}(\mathfrak{M}, \sigma)$ to $\mathrm{FO}_{\mathrm{act}}(\mathfrak{M}_<, \sigma)$.

Q. What do we do first?

A. Let's start with the second step, it's much easier.

Q. I don't see how it can be true that $\mathrm{FO}_{\mathrm{act}}(\mathfrak{M}, \sigma) = \mathrm{FO}_{\mathrm{act}}(\mathfrak{M}_<, \sigma)$. Say if $\mathfrak{M}$ is the real field and we write something like $\exists a \in adom \exists y \in adom\ E(x, y) \wedge (x + y \neq 1)$. How can we do this if only a linear ordering is available?

A. We cannot. But note that most queries over σ-structures that are of interest to us are queries such as graph connectivity, or cardinality comparisons, and they do not depend on which particular elements of $\mathfrak{M}$ that the active domain of a finite structure consists of. These queries are called *generic*.

Q. Can you define them formally?

A. Of course. Let's do it for Boolean (yes/no) queries. Such a query is just a class C of finite σ-structures $\mathcal{A}$ with $adom(\mathcal{A}) \subset U$. Now suppose $\mathcal{A} \in C$, and let $h : U \to U$ be a 1-1 partial map defined on $adom(\mathcal{A})$. The definition of a generic query Q says that then $h(\mathcal{A})$ must be in C too.

Q. Where $h(\mathcal{A})$ is simply $\mathcal{A}$ in which every $a \in adom(\mathcal{A})$ is replaced by $h(a)$?

A. Of course. Can you give me examples of generic and non-generic queries?

Q. I think I can – graph connectivity, evenness of cardinality are generic, but my earlier example – the existence of an edge (x, y) with $x + y \neq 1$ – is not.

A. Exactly. So our first "reduction" is often called an *active-generic collapse*: it says that every generic query expressible in $\mathrm{FO}_{\mathrm{act}}(\mathfrak{M}, \sigma)$ is also expressible in $\mathrm{FO}_{\mathrm{act}}(\mathfrak{M}_<, \sigma)$. That is, $\mathrm{FO}_{\mathrm{act}}(\mathfrak{M}, \sigma) = \mathrm{FO}_{\mathrm{act}}(\mathfrak{M}_<, \sigma)$ with respect to generic queries.

Q. This sounds like a strong result. And what conditions on $\mathfrak{M}$ do you need for it?

A. Here comes the good news – none whatsoever! This is true for all infinite $\mathfrak{M}$.

Q. That's wonderful! Is this hard to prove?

A. Not really. In fact two very similar proofs appeared almost at the same time [8, 33]. They used very similar ideas based on Ramsey's theorem.

Q. Ramsey's theorem? Isn't this about monochromatic cliques and other strange graph-theoretic constructions?

A. These are finite Ramsey theorems. Here we need the original result by Ramsey: if ordered n-tuples over an infinite set U are partitioned into $\ell \geq 2$ classes, then there is an infinite subset $U_0 \subseteq U$ such that all ordered n-tuples over U_0 belong to the same class of the partition.

So next we use this repeatedly to reduce every subformula involving symbols from Ω to a formula that only involves a linear ordering, and over some infinite subset is equivalent to the original one. For example, for $x + y \neq 1$ we can simply find an infinite set $U_0 \subseteq \mathbb{R}$ such that over it for all pairs (x, y) with $x < y$ we have $x + y \neq 1$. Then over U_0 we simply replace $(x + y \neq 1)$ with $x < y$ – and notice that we introduced an ordering!

Q. I think I see the idea now – you eliminate all symbols from Ω except an ordering and still have a formula equivalent to the original one on some infinite set, but by genericity you can assume that your finite structure comes from that set.

A. Exactly. So as you can see, it's a bit tedious but not hard at all. In fact the easiest proof of the active-generic collapse is simply by induction on the structure of a formula, and it is given in full detail in [10] and in Chap. 13 of my book [29].

Q. So far so good, we have the active-generic collapse for all structures. Is it the same for the natural-active collapse?

A. Far from it. Can you think of a simple counterexample?

Q. I think I can; what if we have an empty σ-structure? Then active-domain quantifiers make no sense and any $\mathrm{FO}_{\mathrm{act}}(\mathfrak{M}, \sigma)$ formula is equivalent to a formula that has no quantifiers at all – but this cannot always be true.

A. Yes. In particular this means that every FO formula over $\mathfrak{M}$ is equivalent to a formula that has no quantifiers at all. Do you remember the name of this property?

Q. Of course, it's called quantifier-elimination. I even remember a few examples: $\langle \mathbb{Q}, < \rangle$, $\langle \mathbb{R}, +, \cdot, 0, 1, < \rangle$, or Presburger arithmetic $\langle \mathbb{N}, +, <, 0, 1 \rangle$ if you add all modulo comparisons $n = m(\mod k)$. So if $\mathfrak{M}$ has the natural-active collapse, it must have quantifier-elimination too.

A. Yes, but actually this is not the biggest problem. After all, quantifier-elimination is easy to achieve.

Q. How?

A. You take a structure $\mathfrak{M}$ and simply add a new k-ary predicate symbol P_φ for every formula $\varphi(x_1, \ldots, x_k)$ whose interpretation is $\{\bar{a} \in U^k \mid \mathfrak{M} \models \varphi(\bar{a})\}$. The new structure $\mathfrak{M}_{qe}$ is no different in terms of FO-definability, and it has quantifier-elimination.

Q. I see. And since the active-generic collapse applies to $\mathfrak{M}_{qe}$, it means that all we need to conclude that some generic queries – such as graph connectivity – are not definable in $\text{FO}(\mathfrak{M}, \sigma)$ is to show that $\text{FO}(\mathfrak{M}_{qe}, \sigma) = \text{FO}_{\text{act}}(\mathfrak{M}_{qe}, \sigma)$.

A. You're absolutely right. In fact, there is even a special name for the statement that $\text{FO}(\mathfrak{M}_{qe}, \sigma) = \text{FO}_{\text{act}}(\mathfrak{M}_{qe}, \sigma)$: it's called a *restricted-quantifier collapse*.

Q. And it isn't true for all structures either?

A. No, and in fact some very familiar structures provide good counterexamples. Here is a hint: replace $\mathbb{R}$ by $\mathbb{N}$.

Q. I guess the best known structure on $\mathbb{N}$ is the standard arithmetic of addition and multiplication: $\mathfrak{N} = \langle \mathbb{N}, +, \cdot \rangle$. Are you saying that the restricted quantifier collapse fails for it, and we can have queries that are in $\text{FO}(\mathfrak{N}, \sigma)$ but not in $\text{FO}_{\text{act}}(\mathfrak{N}_{qe}, \sigma)$?

A. That's right. Let's think of an example. What can you say about $\text{FO}_{\text{act}}(\mathfrak{N}_{qe}, \sigma)$?

Q. We have active-generic collapse for it, so I can't express queries such as 'is the cardinality of a structure even?'. So now I need to express it in $\text{FO}(\mathfrak{N}, \sigma)$...

A. And if you remember some computability theory, you can tell me how.

Q. Of course – in $\mathfrak{N}$ I can code every finite structure by a natural number, and then I can express every computable property of natural numbers in FO. So of course I can say that the cardinality of a finite set is even. Now I see that we need to impose some conditions on the background structure.

A. Yes, and there's been quite a lot of work on identifying conditions that guarantee collapse: natural-active or restricted-quantifier. In fact, this work started with the simplest structure $\mathfrak{M}_\emptyset = \langle U, \emptyset \rangle$ with an empty vocabulary, and it was shown, by Hull and Su [24] to admit the natural-active collapse: $\text{FO}(\mathfrak{M}_\emptyset, \sigma) = \text{FO}_{\text{act}}(\mathfrak{M}_\emptyset, \sigma)$.

Q. How does one prove this?

A. We do it by induction on the formula, and the only case that requires work is that of an unrestricted existential quantifier: $\varphi(\bar{x}) = \exists y\ \psi(\bar{x}, y)$. This is equivalent to

$$\exists y \in adom\ \psi(\bar{x}, y)\ \vee\ \bigvee_{x_i \in \bar{x}} \psi(\bar{x}, x_i)\ \vee\ \exists y \notin adom\ \psi(\bar{x}, y).$$

So we need to take care of the last case. But then notice that since the vocabulary is empty, if there is one witness $y \notin adom$ for ψ, then every $y \notin adom$ is a witness for ψ. We thus modify ψ (which is, by the hypothesis, already an $\text{FO}_{\text{act}}(\mathfrak{M}_\emptyset, \sigma)$) by carefully eliminating the variable y: for example, for each relation S in σ, we can safely replace $S(\ldots, y, \ldots)$ by *false*, since y does not belong to the active domain, and likewise we replace each comparison $y = z$, where z is a quantified variable, by *false* too, since all quantification in ψ is over the active domain. We thus get a formula that does not mention y and is equivalent to $\exists y \notin adom\ \psi(\bar{x}, y)$.

Q. I see. But this proof breaks the moment there is anything at all in the vocabulary.

A. Absolutely. And yet the result is true for the real field. Let's look at one example that we've seen already: all pairs (x, y) in a binary relation S lie on a line. That is, $\exists a \exists b \forall x \forall y\, (S(x, y) \to a \cdot x + b = y)$. There is an easy way to eliminate the unrestricted quantifiers $\exists a \exists b$. Can you try to say what it means for a set of points to lie on a line?

Q. Doesn't this happen iff every three points are collinear?

A. Exactly. So we can state this property as $\forall x_1, x_2, x_3, y_1, y_2, y_3 \in adom\, (\bigwedge_{i=1}^{3} S(x_i, y_i) \to \alpha(\bar{x}, \bar{y}))$, where α states that (x_i, y_i), $i \leq 3$, are collinear. And it is easy to write α as a quantifier-free formula.

Q. This is a cute example but it's very ad hoc. And you're saying that we can do something similar with every FO($\mathfrak{R}, \sigma$) query?

A. Yes. Let me tell you the history of this result. It was conjectured in 1990 [26] that some queries such as evenness and graph connectivity are not expressible in FO($\mathfrak{R}, \sigma$), that is, FO + Poly. The suggested approach was to show the natural-active collapse for the real field $\mathfrak{R}$. This was first achieved in [9] by a non-constructive proof, and a constructive proof appeared in [10]. But a year before the proof of Benedikt and myself [9], Paredaens, Van Gucht and Van den Bussche [34] presented a nice constructive proof of the natural-active collapse for $\langle \mathbb{R}, +, -, 0, 1, < \rangle$ – that is, for the case of linear, rather than polynomial, constraints.

Q. Does multiplication make such a difference?

A. In retrospect, it doesn't. In fact, if you look at the proof of the natural-active collapse for $\mathfrak{R}$ in my book [29], it follows the ideas of Paredaens et al [34]. But the path to that proof wasn't straightforward. In fact, the result of [34] was first generalised quite a bit beyond the real field, as it was proved that every *o-minimal* structure has restricted-quantifier collapse [9, 10]. O-minimality is a central concept of contemporary model theory [35, 39]: it refers to ordered structures $\mathfrak{M} = \langle U, \Omega \rangle$ in which every definable subset of U is a finite union of intervals. Can you tell me why $\mathfrak{R}$ is an example of an o-minimal structure?

Q. I think I can: by Tarski's quantifier-elimination, every formula $\varphi(x)$ is equivalent to a Boolean combination of polynomial inequalities $p_i(x) > 0$, so if r and r' are two roots of polynomials p_i's such that no other root occurs between them, then the signs of all the p_i's on (r, r') don't change and hence the truth value of $\varphi(x)$ doesn't change on (r, r'). Are there other interesting examples of o-minimal structures?

A. There are, and perhaps the most celebrated of them is the "exponential field" – the expansion of $\mathfrak{R}$ with the function e^x. The o-minimality of the exponential field was proved by Wilkie [40].

So [9] proved the result for o-minimal structures, and its constructive version [10]

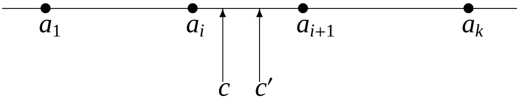


Figure 1: Illustration to the natural-active collapse for the linear case

did so for o-minimal structures again, assuming decidability of their theories. But when the proof was reworked specifically for the case of $\mathfrak{R}$, it looked remarkably similar to the proof for the case of linear constraints.

Q. Will you show this proof to me?

A. I think for this meeting it's better to understand the main idea of the proof for the linear case – after all, it's easier to deal with polynomials that can only have one root, rather than an arbitrary number of roots. So we shall work with $\langle \mathbb{R}, +, -, 0, 1, < \rangle$ as our background structure (and it is well-known to have quantifier-elimination). How do you think the proof will go?

Q. By induction?

A. Of course. So the only case that requires work is elimination of an unrestricted existential quantifier. Let's say we have $\varphi = \exists y \psi(y)$, where $\psi(x)$ is an $\mathrm{FO}_{\mathrm{act}}(\langle \mathbb{R}, +, -, 0, 1, < \rangle, \sigma)$ formula.

Q. Wait a minute, what happened to the free variables? Shouldn't you be looking at $\varphi(\bar{x}) = \exists y \psi(\bar{x}, y)$ to make your induction hypothesis general enough?

A. Of course, but free variables require some extra bookkeeping, and the main ideas can be already seen in the simple case. So let's understand the proof for that case, and you can fill in all the details later.

We assume that $\psi(y)$ is of the form $\exists x_1 \in adom \forall x_2 \in adom \ldots \alpha(\bar{x}, y)$, where α is a Boolean combination of atomic formulae $S(\cdot)$ for $S \in \sigma$ that don't use y (as $S(\cdot, y, \cdot)$ can be replaced by $\exists x' \in adom\, S(\cdot, x', \cdot) \wedge x' = y$), and linear constraints; we also assume that constraints involving y are rewritten as $y \{=, <\} \sum_{i=1}^{m} a_i \cdot x_i + b$.

Let $f_i(\bar{x})$, $1 \le i \le p$, enumerate all the functions that occur as right hand sides of linear constraints whose left-hand side is y, and let $f_0(x_1, \ldots, x_m) = x_1$. Now let $\mathcal{A}$ be a finite σ-structure, and let

$$A = \{ f_i(\bar{a}) \mid i = 1, \ldots, p, \;\; \bar{a} \in adom(\mathcal{A})^m \}.$$

Notice that $adom(\mathcal{A}) \subseteq A$. Assume that $A = \{a_1, \ldots, a_k\}$ with $a_1 < \ldots < a_k$.

Now look at the picture in Fig. 1: if $c \in (a_i, a_{i+1})$ satisfies ψ, then *every* $c' \in (a_i, a_{i+1})$ satisfies ψ because the truth values of $c, c' \{=, <\} f_i(\bar{a})$ are the same for all tuples $\bar{a}$ from the active domain, and all atomic formulae $S(\cdot, c, \cdot)$ and $S(\cdot, c', \cdot)$

are false, since $c, c' \notin adom(\mathcal{A})$.

Q. I see – so if we have a witness for $\psi(y)$ from an interval (a_i, a_{i+1}), the whole interval satisfies ψ. Thus, all we need now is to describe one potential witness from each interval.

A. Yes, and this is easy to do, in a way that is definable with linear constraints: for each interval (a_i, a_{i+1}) we take $(a_i + a_{i+1})/2$ as a witness, for $(-\infty, a_1)$ we take $a_1 - 1$ and for (a_k, ∞) we take $a_k + 1$. Thus, $\exists y \psi(y)$ is now equivalent to:

$$\exists \bar{u}, \bar{v} \in adom \Big(\bigvee_{i=0}^{p} \bigvee_{j=0}^{p} \psi([\frac{f_i(\bar{u}) + f_j(\bar{v})}{2} \;/\; y]) \vee \bigvee_{i=0}^{p} (\psi([(f_i(\bar{u}) - 1) \;/\; y]) \vee \psi([(f_i(\bar{u}) + 1) \;/\; y])) \Big)$$

where $\psi([c/y])$ means that c is substituted for y in ψ. Thus, we replaced $\exists y$ with several active-domain quantifiers ($\exists \bar{u} \in adom \exists \bar{v} \in adom$) and a big disjunction over witnesses from the intervals generated by the set A.

Q. The definition of o-minimality you mentioned also talks about intervals...

A. A very good point. This proof is a special instance of a more general proof for o-minimal structures that uses the same ideas: if there is a witness, then a whole interval is a witness; the number of such intervals is finite; and one can choose specific witnesses from them. O-minimal structures $\mathfrak{M}$ have a remarkable "uniform bounds" property: for each formula $\varphi(x, \bar{y})$ there is a number ℓ such that the set $\{a \mid \mathfrak{M} \models \varphi(a, \bar{c})\}$ is composed of at most ℓ intervals, no matter how we choose $\bar{c}$. This is crucial in the proof as it gives us a finite disjunction of cases to check. In the case of the real field this uniform bounds property follows easily from the fundamental theorem of algebra, but in general this is a very nontrivial property [35, 39].

Q. So o-minimality is the best sufficient condition for collapse?

A. No, there are more conditions known now. They are quite model-theoretic in nature [4, 6, 17], and if you want ot learn about them, there are surveys [30, 7] you can check. And while there is no necessary and sufficient condition for collapse, the property that best describes it is finiteness of the VC (Vapnik-Chervonenkis) dimension.

Q. I remember this notion from computational learning theory [3]! It characterises concepts that are efficiently learnable. What does it have to do with embedded finite models?

A. This notion is used not only in learning, but also in model theory, where it is a very useful concept as was noticed by Shelah 35 years ago [36]. Now let's review the concept of VC dimension, shall we? You said that you know it.

Q. Yes, the VC dimension of a collection C of subsets of a set X is the maximum cardinality of a *shattered* finite set $F \subset X$ – if it exists, and if arbitrarily large sets can be shattered, then the VC dimension is infinite. And F is shattered if $\{F \cap Y \mid Y \in C\}$ is the powerset of X. And what does it mean in the language of an infinite structure $\mathfrak{M}$.

A. We say that $\mathfrak{M}$ has finite VC dimension if every definable family has finite VC dimension. And definable families are given by FO formulae $\varphi(\bar{x}, \bar{y})$ as follows: $\{\{\bar{a} \mid \mathfrak{M} \models \varphi(\bar{a}, \bar{b})\} \mid \bar{b} \in U^{|\bar{b}|}\}$.

Q. Can you give me some examples?

A. Yes: for example, all o-minimal structures [39], but also some unordered structures such as the field of complex numbers $\langle \mathbb{C}, +, \cdot \rangle$ [23].

Q. And in what sense is it close to characterising the collapse?

A. It is known that restricted-quantifier collapse ($\mathrm{FO}(\mathfrak{M}_{qe}, \sigma) = \mathrm{FO}_{\mathrm{act}}(\mathfrak{M}_{qe}, \sigma)$) implies finiteness of VC dimension [11], and finiteness of VC dimension implies that $\mathrm{FO}(\mathfrak{M}_{qe}, \sigma)$ and $\mathrm{FO}_{\mathrm{act}}(\mathfrak{M}_{qe}, \sigma)$, while not necessarily the same, define the same generic queries [4]. In particularly, this very strong result of [4] implies that over every structure of finite VC dimension, the set of generic queries in $\mathrm{FO}(\mathfrak{M}, \sigma)$ is the same as the set of queries definable in $\mathrm{FO}_{\mathrm{act}}(\mathfrak{M}_{<}, \sigma)$.

Q. You never said anything about the complexity of the collapse: how hard is it to convert an $\mathrm{FO}(\mathfrak{M}, \sigma)$ formula into an $\mathrm{FO}_{\mathrm{act}}(\mathfrak{M}, \sigma)$ formula?

A. Unfortunately not much is known about this, and complexity analyses may differ significantly for different structures, as such conversion algorithms need to make calls to quantifier-elimination procedures. One case though that was studied in detail is that of the real field and σ consisting of a single unary predicate. For this case Basu [5] developed special algorithms that also give the best known running time for quantifier-elimination for $\mathfrak{R}$.

Q. I think I have plenty of new information now ... I hadn't realised that there was a whole field within model theory developed when Jan Van den Bussche [38] made a passing remark about collapse theorems. It's quite nice to see this interplay between finite and infinite models.

A. Yes, but I don't want you to leave thinking that this is it for finite/infinite models interaction. There are *plenty* of other directions with very interesting results, techniques, and applications.

Q. Can you give me some examples?

A. Certainly. There are metafinite models of Grädel and Gurevich [18] which are finite models with some functions defined on their elements (or tuples of elements) whose range is in the universe of a fixed infinite structure. In logics over metafinite models, variables typically range over the finite part, so interplay is not as complete as in the case of embedded finite models; however, metafinite models make it easy to extend other logics typically studied in the finite model theory context.

There are various finite representations of infinite structures, like in the case of constraint databases. For example, in recursive structures, all predicate symbols are interpreted as recursive relations that are finitely representable by Turing machines. There are interesting connections between finite model theory and the behaviour of logics over recursive structures; a nice survey of this area was written by Harel [22]. As a special and more manageable case, we can consider structures in which all basic predicates (and thus by closure properties, all definable sets) are given by finite automata. These are automatic structures that have been studied rather actively in recent years [27, 15, 11, 12]. They have decidable theories – in fact, decision procedures use automata-theoretic techniques – and these structures found applications in verification and query languages. In particular, [11] looks at finite models embedded into automatic structures. In constraint satisfaction, logical studies of problems with infinite templates recently appeared [14], and those can be viewed as a special case of embedded finite models. In the field of verification people also have been looking at infinite graphs describing configurations of pushdown automata [32, 16]. These again are finitely represented infinite structures with decidable theories that have applications in software verification. So as you can see, there are many other interesting meetings that Yuri can arrange for you in the future, if you'd like to learn more about connections between the finite and the infinite in CS logic.

Q. I shall certainly think about it. And for now, thanks for your time today.

A. You're welcome.

References

[1] S. Abiteboul, R. Hull and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[2] A.K. Ailamazyan, M.M. Gilula, A.P. Stolboushkin and G.F. Shvarts. Reduction of a relational model with infinite domains to the finite-domain case. *Doklady Akademii Nauk SSSR*, 286 (1) (1986), 308–311. Translation in *Soviet Physics – Doklady*, 31 (1986), 11–13.

[3] M. Anthony and N. Biggs. *Computational Learning Theory.* Cambridge Univ. Press, 1992.

[4] J. Baldwin and M. Benedikt. Stability theory, permutations of indiscernibles, and embedded finite models. *Trans. Amer. Math. Soc.* 352 (2000), 4937–4969.

[5] S. Basu. New results on quantifier elimination over real closed fields and applications to constraint databases. *Journal of the ACM*, 46 (1999), 537–555.

[6] O. Belagradek, A. Stolboushkin, M. Taitslin. Extended order-generic queries. *Ann. Pure and Appl. Logic*, 97 (1999), 85–125.

[7] M. Benedikt. Generalizing finite model theory. In *Lecture Notes in Logic, vol. 24, Logic Colloquium 2003*, ASL, 2006.

[8] M. Benedikt, G. Dong, L. Libkin and L. Wong. Relational expressive power of constraint query languages. *Journal of the ACM* 45 (1998), 1–34.

[9] M. Benedikt and L. Libkin. On the structure of queries in constraint query languages. In *LICS'96*, pages 25–34.

[10] M. Benedikt and L. Libkin. Relational queries over interpreted structures. *Journal of the ACM*, 47 (2000), 644–680.

[11] M. Benedikt, L. Libkin, T. Schwentick, L. Segoufin. Definable relations and first-order query languages over strings. *Journal of the ACM* 50 (2003), 694–751.

[12] A. Blumensath, E. Grädel. Automatic structures. In *LICS 2000*, pages 51–62.

[13] J. Bochnak, M. Coste, M.-F. Roy. *Real Algebraic Geometry*. Springer Verlag, 1998.

[14] M. Bodirsky, J. Nešetřil. Constraint satisfaction with countable homogeneous templates. In *CSL 2003*, pages 44–57.

[15] V. Bruyère, G. Hansel, C. Michaux, R. Villemaire. Logic and *p*-recognizable sets of integers. *Bull. Belg. Math. Soc.* 1 (1994), 191–238.

[16] J. Esparza, A. Kucera, S. Schwoon. Model checking LTL with regular valuations for pushdown systems. *Inf.& Comput.* 186 (2003), 355–376.

[17] J. Flum and M. Ziegler. Pseudo-finite homogeneity and saturation. *J. Symbolic Logic* 64 (1999), 1689–1699.

[18] E. Grädel and Y. Gurevich. Metafinite model theory. *Inf. and Comput.*, 140 (1998), 26–81.

[19] M. Grohe, T. Schwentick. Locality of order-invariant first-order formulas. *ACM Trans. Comput. Log.* 1 (2000), 112–130.

[20] S. Grumbach, P. Rigaux, L. Segoufin. The DEDALE system for complex spatial queries. In *SIGMOD'98*, pages 213–224.

[21] S. Grumbach and J. Su. Queries with arithmetical constraints. *TCS* 173 (1997), 151–181.

[22] D. Harel. Towards a theory of recursive structures. In *MFCS'98*, pages 36–53.

[23] W. Hodges. *Model Theory*. Cambridge, 1993.

[24] R. Hull, J. Su. Domain independence and the relational calculus. *Acta Inform.* 31:513–524, 1994.

[25] P. Kanellakis. Elements of relational database theory. In *Handbook of TCS, Vol. B*, 1990, pages 1073–1156.

[26] P. Kanellakis, G. Kuper, and P. Revesz. Constraint query languages. *JCSS*, 51 (1995), 26–52. Extended abstract in *PODS'90*, pages 299–313.

[27] B. Khoussainov and A. Nerode. Automatic presentations of structures. In *Logic and Computational Complexity*, Springer LNCS vol. 960, 1994, pages 367–392.

[28] G. Kuper, L. Libkin and J. Paredaens, eds. *Constraint Databases*. Springer Verlag, 2000.

[29] L. Libkin. *Elements of Finite Model Theory*. Springer, 2004.

[30] L. Libkin. Embedded finite models and constraint databases. In E. Grädel et al., *Finite Model Theory and its Applications*, Springer, 2007, to appear.

[31] D. Maier. *The Theory of Relational Databases.* Computer Science Press, 1983

[32] D. Muller, P. Schupp. The theory of ends, pushdown automata, and second-order logic. *TCS* 37 (1985), 51–75.

[33] M. Otto and J. Van den Bussche. First-order queries on databases embedded in an infinite structure. *Inform. Proc. Letters* 60 (1996), 37–41.

[34] J. Paredaens, J. Van den Bussche, and D. Van Gucht. First-order queries on finite structures over the reals. *SIAM J. Comput.* 27 (1998), 1747–1763. Extended abstract in *LICS'95*.

[35] A. Pillay, C. Steinhorn. Definable sets in ordered structures. *Bull. of the AMS* 11 (1984), 159–162.

[36] S. Shelah. Stability, the f.c.p., and superstability. *Ann. of Math. Logic* 3 (1971), 271–362.

[37] S. Shelah. On the very weak 0-1 law for random graphs with orders. *J. Log. Comput.* 6 (1996), 137–159.

[38] J. Van den Bussche. First-order topological properties. *Bull. EATCS* 87 (2005), 155–164.

[39] L. van den Dries. *Tame Topology and O-Minimal Structures*. Cambridge, 1998.

[40] A.J. Wilkie. Model completeness results for expansions of the ordered field of real numbers by restricted Pfaffian functions and the exponential function. *J. Amer. Math. Soc.* 9 (1996), 1051–1094.

The Natural Computing Column

by

Grzegorz Rozenberg
Leiden University, Leiden Center for Natural Computing
Niels Bohrweig 1, 2333 CA Leiden, The Netherlands
rozenber@liacs.nl

Z. Pawlak, a Precursor of DNA Computing and of Picture Grammars

Solomon Marcus
Romanian Academy
Calea Victoriei 125, Bucharest, Romania
solomon.marcus@imar.ro

41 years ago, Z. Pawlak has published in Polish language a book aimed perhaps for initiation in the field of mathematical linguistics (Pawlak 1965). Short time after this event, he attended an international Conference in Bucharest and I met him there. He offered me a copy of this book. As a matter of fact, he showed me the book and he said that he is sorry to have it in a language which is not available to me. But I told him that I would like to have the book and I will manage to follow it at least partly. Happy idea! Besides some usual introductory notions concerning the mathematical approach to grammars (the title in Polish “Gramatika i matematika" was clearly “Grammar and mathematics"), a special chapter called my attention, because it was concerned with the grammar of the genetic code. I was already introduced, at that time, in the works of Roman Jakobson and of many other authors concerning the analogy between linguistics and molecular genetics. Pawlak’s approach was mainly presented in symbols, graphs and geometric pictures, while the few words in Polish were in most cases international words like codons, amino acids, nucleotides, proteins.

Bulletin of the EATCS no 90, pp. 150–154, October 2006

It is interesting to recall the period of the sixties of the past century. After a long period in which historical linguistics used ideas and metaphors of Darwinian biology, an important change took place: instead to use biological ideas and metaphors in linguistics, linguistic ideas and metaphors related to phonemic and morphemic segmentation penetrated in the study of nucleic acids, amino acids and proteins.

To this itinerary of opposite sense in respect to the previous one, Pawlak was adding the idea of a generative perspective in the study of heredity. In this aim, he proposed some mechanism operating concomitantly in two directions. On the one hand, in the direction of formal grammars, on the other hand, in the direction of what was called later picture grammars. Let us recall that both formal grammars and picture grammars were at that time at their very beginning. Formal grammars theory had to wait the year 1973 for a first satisfactory rigorous presentation (Salomaa 1973), while picture grammars had to wait the year 1967 for a first systematic attempt (Shaw 1967) and two more years for the monograph by Rosenfeld (1969).

Let us recall the main ideas of Pawlak's approach. Denote by 0, 1, 2, and 3 the four types of nucleotide bases forming the alphabet on which the RNAs are defined. There are 64 modes of arrangements with repetition of them in groups of three elements forming so strings of length three (the constant length of all codons). Codons are for RNAs what morphemes are for well- formes strings in natural languages, while nucleotide bases are for RNAs what phonemes are in natural languages. The starting idea of Pawlak is to associate to each codon an equilateral triangle. Taking into account that a codon is a word of length three on the alphabet 0, 1, 2, 3, the associated triangle will have the respective symbols as labels of its sides. But, as it is well-known, the genetic code establishes a correspondence between codons and amino acids (defining in this way the move from the world of chemistry to the world of biology). There are only 20 types of amino acids relevant for heredity, so Pawlak proposes a way to select exactly 20 types of triangles among the 64 types which are possible from a purely combinatorial point of view. Let us distinguish, for any triangle, the base b, the left side l and the right side r, see Figure 1(a). If the codon is ijk, then we associate i to l, j to b, and k to r. Moreover, Pawlak introduces the restriction $i < j \geq k$, i.e., the symbol associated to l is strictly smaller than the symbol associated to b, which is larger than or equal to the symbol associated to r. It can be seen that the only triangles satisfying this requirements are:

$$a = 010,\ b = 011,\ c = 020,\ d = 021,\ e = 022,\ f = 120,\ g = 121,$$
$$h = 122,\ i = 030,\ j = 031,\ k = 032,\ l = 033,\ m = 130,\ n = 131,$$
$$o = 132,\ p = 133,\ q = 230,\ r = 231,\ s = 232,\ t = 233.$$

In a next step, Pawlak introduces a recursive procedure to define a generative picture grammar, whose basic bricks are the 20 types of labeled triangles. The rule of this procedure are the following:

1. Every triangle from the list $a, b, c, \ldots, r, s, t$ is a well-formed string; they are the only well-formed strings of length one.

2. All well-formed strings are words on the alphabet $\{a, b, c, \ldots, r, s, t\}$. Given a well-formed string x and adding to it a triangle A from the list 1, such that the label of its base is the same as the label of the left or right side of an already existing triangle B in x (in other words, the base of A is the same as the left side or the right side of B), then the new string y of triangles so obtained is again well-formed.

3. The strings obtained by the rules 1 and 2 are the only well-formed strings.

A saturated well-formed string is one from which no other longer well-formed string can be obtained. For instance, the strings of length one 010, 020, 030 are saturated, the strings 011, 010 and 021, 010 are saturated strings of length 2 etc. It is easy to see that there are saturated strings of any length. This fact is a consequence of the existing of some triangles that can be added to themselves. For instance, 011 is such a recursive triangle. We can add it to itself n times, then add 010 to obtained a saturated string. For instance, the saturated string of length 4 obtained in this way is: 011, 011, 011, 010. Other examples of recursive triangles are: 022, 122, 233, 133.

Pawlak calls protein any saturated well-formed string. He defines a kind of dependency grammar, having 20 rules: to each well-formed triangle ijk Pawlak associates the rule $j \rightarrow ik$, where at left we have the label of the base, while at right we have the label of the left side followed by the label of the right side. From this dependency grammar Pawlak moves to a graph representation. The triangle ijk is represented by a vertical line associated to the base labeled with j, while from the inferior extremity of this line we start a segment oriented towards the south-left labeled with i and a segment oriented towards the south-right, labeled with k, see Figure 1(b). In this way, the recursive process induces a tree which can be developed as soon as it is not yet saturated.

We have shown in (Marcus 1974) that a non-deterministic propagating semi-Lindenmayer system can be defined, which is equivalent to the above defined Pawlak mechanism. But we are interested not only in the result of the generative process; we would like to know something about the structure of the language of derivations in the respective semi-Lindenmayer system. This question was left unanswered in (Marcus 1974). In the same paper we have presented a Chomsky type picture grammar which is context-free, but whose language of derivations is not context-free; it is just the Chomskian equivalent of Pawlak's dependency picture grammar.

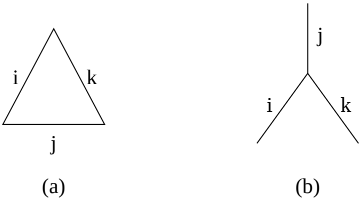


Figure 1: (a) The triangle associated to a codon ijk; (b) A different graphical representation of the triangle in (a).

Some advantages and some shortcomings of Pawlak's mechanism and of the corresponding Chomskian mechanism are discussed in (Marcus 1974). The whole problem deserves to be reconsidered, in the light of the new field of DNA computing, for which we send the reader to (Păun-Rozenberg-Salomaa 1998).

We conclude with some hints about the idea of a semi-Lindenmayer system. It is an ordered pair $S = \langle V, p\rangle$, where V is a finite non-empty set called alphabet, while p is a mapping associating to each element in V a language over V. If for each $v \in V$ the set $p(v)$ contains exactly one finite string over V, then S is said to be deterministic; otherwise, S is said to be non-deterministic. We say that S is propagating, if for each $v \in V$ any string in $p(v)$ is of strictly positive length; otherwise, S is non-propagating. Define now the language $L(S, M)$ generated by a semi-Lindenmayer system S with respect to a language M over V. The string y directly derives from the string x of strictly positive length if there exists a positive integer n such that $x = a(1)a(2)\dots a(n)$, $y = b(1)b(2)\dots b(n)$, where each $a(i)$ $(1 \le i \le n)$ belongs to V and each $b(i)$ $(1 \le i \le n)$ belongs to V^*, with $b(i) \in p(a(i))$ for any $1 \le i \le n$. If p is a homomorphism, we put for any finite string w over V, $w = c(1)c(2)\dots c(n)$, $p(w) = p(c(1))p(c(2))\dots p(c(n))$. We say that the string v derives in S from the string u of strictly positive length if there exists a finite sequence of finite strings $x(1), x(2), \dots, x(q)$ over V, such that $x(1) = u$, $x(q) = v$, and $x(i+1)$ directly derives from $x(i)$ for any $1 \le i \le q-1$. The string y is generated by S with respect to the language M over V if there exists $x \in M$ from which y derives in S. $L(S, M)$ is by definition the set of all strings generated by S with respect to M.

In (Marcus 1974), it is proved that the Pawlak dependency grammar can be expressed as a non-deterministic propagating semi-Lindenmayer system with respect to the language M consisting of four strings of length one: 0, 1, 2, 3. The mapping p is defined by $p(0) = \{0\}$, $p(1) = \{00, 01\}$, $p(2) = \{00, 01, 02, 10, 11, 12\}$ and $p(3) = \{00, 01, 02, 03, 10, 11, 12, 13, 20, 21, 22, 23\}$. The language generated by the considered system is just the set of all saturated well-formed strings, in the sense of Pawlak, i.e., the set of proteins. What about the language of derivations in S?

A basic shortcoming of Pawlak's approach was that he did not take in consideration the Watson-Crick structure of double-helix. Our 1974 approach continuing Pawlak's work had the same shortcoming, as it was clearly mentioned in (Marcus 1974). This missing structure became just the point of departure in Tom Head's pioneering work on DNA computing (Head 1987).

References

[1] T. Head (1987), Formal language theory and DNA: an analysis of the generative capacity of specific recombinant behaviors. Bull. Math. Biology 49 (1987), 737–759.

[2] S. Marcus (1974), Linguistic structures and generative devices in molecular genetics. *Cahiers de Linguistique Théorique et Appliquée*, 11, 2, 77–104.

[3] Gh. Păun, G. Rozenberg, A. Salomaa (1998), *DNA Computing. New Computing Paradigms*. Springer, Berlin.

[4] Z. Pawlak (1965), *Gramatika i matematika*. Publishig House of the Polish Academy of Sciences.

[5] A. Rosenfeld (1969), *Picture Processing by Computer*. Academic Press, New York – London.

[6] A.C. Shaw (1967), *A Proposed Language for the Formal Description of Pictures*. GSG Memo 28, Stanford, California, February 1967, 32 pp.

[7] A. Salomaa (1973), *Formal Languages*. Academic Press, New York – London.

Remark. This article presents in a very convincing way some of Pawlak's important early insights. It also describes interesting details about the landscape of theoretical computer science some 35 years ago. Then the terminology around Lindenmayer systems was still developing. What are called *semi-Lindenmayer systems* in the article are, according to present terminology, *0L systems with a finite axiom set*, or briefly *F0L systems*. It is interesting to note that one of the mathematically most sophisticated results about L systems deals with finite axiom sets: If we add C (coding) and F to the name of the system, then we may also add P (propagation) and, thus, avoid cell death. The details of this construction are given in Chapter 5 of Volume I of *Handbook of Formal Languages*, Springer-Verlag 1997, edited by us.

G. Rozenberg A. Salomaa

THE PROGRAMMING LANGUAGES COLUMN

BY

IAN MACKIE

Department of Computer Science
King's College London, Strand, London, WC2R 2LS, UK
Ian.Mackie@kcl.ac.uk, http://www.dcs.kcl.ac.uk/staff/ian

LANGUAGES FOR CONCURRENCY

Catuscia Palamidessi
INRIA and LIX, École Polytechnique
catuscia@lix.polytechnique.fr

Frank D. Valencia
CNRS and LIX, École Polytechnique
frank.valencia@lix.polytechnique.fr

Abstract

This essay offers an overview of basic aspects and central development in Concurrency Theory based on formal languages. In particular, it focuses on the theory of Process Calculi.

1 Introduction

Concurrency is concerned with the fundamental aspects of systems consisting of multiple computing agents, usually called *processes*, that interact among each other. This covers a vast variety of systems which nowadays, due to technological advances such as the Internet, programmable robotic devices and mobile computing, most people can easily relate to. Some examples are:

Bulletin of the EATCS no 90, pp. 155–171, October 2006
ⓒ *European Association for Theoretical Computer Science*

- *Message-passing* communication based systems: Agents interact by exchanging messages. For instance, e-mail communication on the Internet, or robot point-to-point exchange of messages via infra-red communication.

- *Shared-Variables* communication based systems: Agents communicate by posting and reading information from a central location. For instance, reading and posting information on a server as in an Internet newsgroup. In the context of co-operative robotic devices, there can be a central control, usually a PC, on which the robots can post and read information (e.g., their relative positions).

- *Synchronous* systems: As opposed to *asynchronous* systems, in synchronous systems, agents need to synchronize with one another. In Internet telephony services the caller and the callee's terminal need to synchronize to establish communication. In systems of mobile robotic devices, robots most certainly need to synchronize, e.g., to avoid bumping into each other. An example of asynchrony is SMS communication on mobile phones.

- *Reactive* systems: Involve systems that maintain an ongoing interaction with their environment. For instance, reservation systems and databases on the Internet. Co-operative robotic devices are typically programmed to react to their surroundings, e.g., going backwards whenever a touch sensor is pressed.

- *Timed* systems: Systems in which the agents are constrained by temporal requirements. For example, browser applications are constrained by timer-based exit conditions (i.e., *time-outs*) for the case in which a server cannot be contacted. E-mailer applications can be required to check for messages every k time units. Also, robots can be programmed with time-outs (e.g., to wait for some signal) and with timed instructions (e.g., to go forward for 42 time units).

- *Mobile* systems: Agents can change their communication links. This is the essence of mobile computing devices. For example, portable computers can connect to the Internet from different locations. Robotic devices also exhibit mobility since, as they are on the move they may change their communication configuration. E.g., robots, which could initially communicate with one another, may sometime later be too far away to continue to do so.

- *Secure* systems: Systems in which critical resources of some sort (e.g., secret information) must not be accessed, misused or modified by unwanted agents. Credit card usage on the Internet is now a common practice involving secure systems. To a more physical level, one now hears of robotic security systems [9] which involve mobile devices that are strategically programmed to patrol, detect intruders and respond accordingly.

The above are but a few representatives of systems exhibiting concurrency, often referred to as *concurrent* systems. Furthermore, they can be combined to give rise to very complex concurrent systems; for example the Internet itself.

1.1 Problem: Reasoning about Concurrency

The previous examples illustrate the practical relevance, complexity and ubiquity of concurrent systems. It is therefore crucial to be able to describe, analyze and, in general, reason about concurrent behavior. This reasoning must be precise and reliable. Consequently, it ought to be founded upon mathematical principles in the same way as the reasoning about the behavior of sequential programs is founded upon logic, domain theory and other mathematical disciplines.

Nevertheless, giving mathematical foundations to concurrent computation has become a serious challenge for computer science. Traditional mathematical models of (sequential) computation based on functions from inputs to outputs no longer apply. The crux is that concurrent computation, e.g., in a reactive system, is seldom expected to terminate, it involves constant interaction with the environment, and it is *non-deterministic* owing to unpredictable interactions among agents.

1.2 Solution: Models of Concurrency

Computer science has therefore taken up the task of developing *models*, conceptually different from those of sequential computation, for the precise understanding of the behavior of concurrent systems. Such models, as other scientific models of reality, are expected to satisfy the following criteria:

- They must be *simple*, i.e., based upon few basic principles.
- They must be *expressive*, i.e., capable of capturing interesting real-world situations.
- They must be *formal*, i.e., founded upon mathematical principles.
- They must provide *techniques* to allow reasoning about their particular focus.

In order to develop a model of concurrency one could suggest the following general strategy: Seize upon a few pervasive aspects of concurrency (e.g., synchronous communication), make them the focus of a model, and then submit the model to the above criteria. This strategy can be claimed to have been involved in the development of a mature collection of models for various aspects of concurrency. Some representatives of this collection are mentioned next.

Representative models for synchronous communication. Some of the most mature and well-known models of concurrency are process calculi like Milner's CCS [17], Hoare's CSP [12], and ACP (developed by Bergstra and Klop [5] and also by Baeten [4]). The common focus of these models is synchronous communication.

Process calculi treat processes much like the λ-calculus treats computable functions. They provide a language in which the structure of *terms* represents the structure of processes together with an *operational semantics* to represent computational steps. For example, the term $P \parallel Q$, which is built from P and Q with the *constructor* $\parallel$, represents the process that results from the parallel execution of those represented by P and Q. An operational semantics may dictate that if P can evolve into P' in a computational step then $P \parallel Q$ can also evolve into $P' \parallel Q$ in a computational step.

An appealing feature of process calculi is their *algebraic* treatment of processes. The constructors are viewed as the *operators* of an algebraic theory whose equations and inequalities among terms relate process behavior. For instance, the construct $\parallel$ can be viewed as a commutative operator, hence the equation $P \parallel Q \equiv Q \parallel P$ states that the behavior of the two parallel compositions is the same. Because of this algebraic emphasis, these calculi are often referred to as *process algebras*.

A representative model for true-concurrency. Another important model of concurrency is Petri Nets [23]. The focus of Petri Nets is the simultaneous occurrence of actions (i.e., *true concurrency*). The theory of Petri Nets, which was the first well-established theory of concurrency, is an elegant generalization of classic automata theory in which the concept of concurrently occurring transitions can be expressed.

1.3 Model Extensions

Science has made progress by extending well established theories to capture new and wider phenomena. For instance, computability theory was initially concerned only with functions on the natural numbers but it was later extended to deal with functions on the reals [10]. Also, classical logic was extended to various modal logics to study reasoning involving modalities such as possibility, necessity and temporal progression. Another example of great relevance is automata theory, initially confined to finite sequences, but later generalized to reason about infinite ones as in Büchi automata theory [6].

Similarly, several mature models of concurrency have been extended to treat additional issues. These extensions should not come as a surprise since the field is indeed large and subject to the advents of new technology.

Somce examples of these additional issues are the notions of mobility and security which now pervade the informational world; none of the representative models mentioned above dealt with these notions. It was later found that a CCS extension, the π-calculus [19], could treat mobility in a very satisfactory way. A further extension, the spi-calculus [1], was also designed to model security.

Another prominent example is the notion of *time*. This notion not only is a fundamental concept in concurrency but also in science at large. Just like modal extensions of logic for temporal progression study time in logic reasoning, theories of concurrency were extended to study time in concurrent activity. For instance, neither CCS, CSP nor Petri Nets, in their basic form, were concerned with temporal behavior but they all have been extended to incorporate an explicit notion of time, leading for instance Timed CCS [33], Timed CSP [28], Timed ACP [3] and Timed Petri Nets [34].

2 The Theory of Process Calculi

This section describes some fundamental concepts from process calculi. We do not intend to give an in-depth review of these calculi (the interested reader is referred to [18]), but rather to describe those issues which influenced their development.

There are many different process calculi in the literature mainly agreeing in their emphasis upon algebra. The main representatives are CCS [17] , CSP [12] and the process algebra ACP [5, 4]. The distinctions among these calculi arise from issues such as the process constructions considered (i.e., the language of processes), the methods used for giving meaning to process terms (i.e. the semantics), and the methods to reason about process behavior (e.g., process equivalences or process logics). Some other issues addressed in the theory of these calculi are their expressive power, and analysis of their behavioral equivalences. In what follows we discuss some of these issues briefly.

2.1 The Language of Processes

A common feature of the languages of process calculi is that they pay special attention to economy. That is, there are few operators or combinators, each one with a distinct and fundamental role. Process calculi usually provide the following combinators:

- *Action*, for representing the occurrence of atomic actions.
- *Product*, for expressing the parallel composition.
- *Summation*, for expressing alternate course of computation.
- *Restriction* (or *Hiding*), for delimiting the interaction of processes.
- *Recursion*, for expressing infinite behavior.

A process language. For the purposes of the exposition of the next sections we shall define a basic process language which exemplifies the above.

We presuppose an infinite set $\mathcal{N}$ of *names* $a, b, \ldots$. and then introduce a set of *co-names* $\overline{\mathcal{N}} = \{\overline{a} \mid a \in \mathcal{N}\}$ disjoint from $\mathcal{N}$. The set of *labels*, ranged over by l and l', is $\mathcal{L} = \mathcal{N} \cup \overline{\mathcal{N}}$. The set of *actions Act*, ranged over by the boldface symbols **a** and **b** extends $\mathcal{L}$ with a new symbol τ. The action τ is said to be the *silent* (*internal* or *unobservable*) action. The actions a and $\overline{a}$ are thought of as being *complementary*, so we decree that $\overline{\overline{a}} = a$. The syntax of processes is given by:

$$P, Q, \ldots ::= 0 \mid \mathbf{a}.P \mid P + Q \mid P \parallel Q \mid P\backslash a \mid A\langle b_1, \ldots, b_n\rangle$$

Intuitive Description. The intuitive meaning of the process terms is as follows. The process 0 does nothing. $\mathbf{a}.P$ is the process which performs an atomic action **a** and then behaves as P. The summation $P + Q$ is a process which may behave as either P or Q. $P \parallel Q$ represents the parallel composition of P and Q. Both P and Q can proceed independently but they can also synchronize if they perform complementary actions. The restriction $P\backslash a$ behaves as P except that it cannot perform the actions a or $\overline{a}$. The names a and $\overline{a}$ are said to be *bound* in $P\backslash a$. $A\langle b_1, \ldots, b_n\rangle$ denotes the invocation to a unique recursive definition of the form $A(a_1, \ldots, a_n) \stackrel{\text{def}}{=} P_A$ where all the non-bound names of process P_A are in $\{a_1, \ldots, a_n\}$. Obviously P_A may contain invocations to A. The process $A\langle b_1, \ldots, b_n\rangle$ behaves as $P_A[b_1, \ldots, b_n/a_1, \ldots, a_n]$, i.e., P_A with each a_i replaced by b_i – with renaming of bound names wherever necessary to avoid captures.

2.2 Semantics of Processes

The methods by which process terms are endowed with meaning may involve at least three approaches: *operational*, *denotational* and *algebraic* semantics. Traditionally, CCS and CSP emphasize the use of the operational and denotational method, respectively, whilst the emphasis of ACP is upon the algebraic method.

Operational semantics. The methods was pioneered by Plotkin in his Structural Operational Semantics (SOS) work [24, 25, 26]. An operational semantics interprets a given process term by using transitions (labeled or not) specifying its computational steps. A labeled transition $P \xrightarrow{\mathbf{a}} Q$ specifies that P performs **a** and then behaves as Q. The relations $\xrightarrow{\mathbf{a}}$ are defined to be the smallest which obey the rules in Table 1. In these rules the transition below the line is to be inferred from those above the line.

$$\text{ACT}\ \frac{}{\mathbf{a}.P \xrightarrow{\mathbf{a}} P}$$

$$\text{SUM}_1\ \frac{P \xrightarrow{\mathbf{a}} P'}{P + Q \xrightarrow{\mathbf{a}} P'} \qquad \text{SUM}_2\ \frac{Q \xrightarrow{\mathbf{a}} Q'}{P + Q \xrightarrow{\mathbf{a}} Q'}$$

$$\text{COM}_1\ \frac{P \xrightarrow{\mathbf{a}} P'}{P \parallel Q \xrightarrow{\mathbf{a}} P' \parallel P} \qquad \text{COM}_2\ \frac{Q \xrightarrow{\mathbf{a}} Q'}{P \parallel Q \xrightarrow{\mathbf{a}} P \parallel Q'}$$

$$\text{COM}_3\ \frac{P \xrightarrow{l} P' \quad Q \xrightarrow{\bar{l}} Q'}{P \parallel Q \xrightarrow{\tau} P' \parallel Q'}$$

$$\text{RES}\ \frac{P \xrightarrow{\mathbf{a}} P'}{P \backslash a \xrightarrow{\mathbf{a}} P' \backslash a}\ \text{ if } \mathbf{a} \neq a \text{ and } \mathbf{a} \neq \overline{a}$$

$$\text{REC}\ \frac{P_A[b_1, \ldots, b_n/a_1, \ldots, a_n] \xrightarrow{\mathbf{a}} P'}{A\,\langle b_1, \ldots, b_n \rangle \xrightarrow{\mathbf{a}} P'} \qquad \text{if } A(a_1, \ldots, a_n) \overset{\text{def}}{=} P_A$$

Table 1: An operational semantics for a process calculus.

The rules in Table 1 are easily seen to realize the intuitive description of processes given in the previous section. Let us describe some. The rules SUM$_1$ and SUM$_2$ say that the first action of $P + Q$ determines which alternative is selected, the other is discarded. The rules for composition COM$_1$ and COM$_2$ describe the concurrent performance of P and Q. The rule COM$_3$ describes a synchronizing communication between P and Q. For recursion, the rule REC says that the actions of (an invocation) $A\,\langle b_1, \ldots, b_n \rangle$ are just those that can be inferred by replacing every a_i with b_i in (the definition's body) P_A where $A(a_1, \ldots, a_n) \overset{\mathtt{def}}{=} P_A$.

Behavioral equivalences. Having defined the operational semantics, we can now introduce some typical notions of process equivalence. Here we shall recall *trace*, *failures* and *bisimilarity* equivalences. Although these equivalences can be defined for both CSP and CCS, traditionally the first two are associated with CSP and the last one is associated with CCS.

We need a little notation: The empty sequence is denoted by ϵ. Given a sequence of actions $s = \mathbf{a}_1.\mathbf{a}_2.\ldots \in Act^*$, define $\overset{s}{\Longrightarrow}$ as

$$(\xrightarrow{\tau})^* \xrightarrow{\mathbf{a}_1} (\xrightarrow{\tau})^* \ldots (\xrightarrow{\tau})^* \xrightarrow{\mathbf{a}_n} (\xrightarrow{\tau})^*$$

Notice that $\overset{\epsilon}{\Longrightarrow} = \xrightarrow{\tau}^*$. We use $P \overset{s}{\Longrightarrow}$ to mean that there exists a P' s.t., $P \overset{s}{\Longrightarrow} P'$ and similarly for $P \xrightarrow{s}$.

- *Trace equivalence.* This equivalence is perhaps the simplest of all. Intuitively, two processes are deemed trace equivalent if and only if they can perform exactly the same sequences of non-silent (or observable) actions. Formally, we say that P and Q are *trace equivalent*, written $P =_T Q$, if for every $s \in \mathcal{L}^*$,

  $$P \overset{s}{\Longrightarrow} \text{ iff } Q \overset{s}{\Longrightarrow} .$$

  A drawback of $=_T$ is that it is not sensitive to deadlocks. For example, let $P_1 = a.b.0 + a.0$ and $Q_1 = a.b.0$. Notice that $P_1 =_T Q_1$ but unlike Q_1, after doing a, P_1 can reach a state in which it cannot perform any action, i.e., a *deadlock*.

- *Failures equivalence.* This equivalence is more discriminating (stronger or finer) than trace equivalence. In particular it is sensitive to deadlocks.

  A *failure* is a pair (s, L) where $s \in \mathcal{L}^*$ (called a trace) and L is a set of labels. Intuitively, (s, L) is a failure of P if P can perform a sequence of observable actions s evolving into a P' in which no action from $L \cup \{\tau\}$ can be performed.

  Formally, we say that (s, L) is a *failure of* P if there exists P' such that

  $$(1)\ P \overset{s}{\Longrightarrow} P',\ (2)\ P' \overset{\tau}{\not\longrightarrow} \text{ and } (3) \text{ for all } l \in L,\ P' \overset{l}{\not\longrightarrow}.$$

  We then say that P and Q are *failures-equivalent*, written $P =_F Q$, iff they posses the same failures.

  Notice that $=_F \subseteq =_T$ as a trace is part of a failure. To see the strict inclusion, notice that for the trace equivalent processes P_1 and Q_1 given in the previous point, we have $P_1 \neq_F Q_1$ as P_1 has the failure $(a, \{b\})$ but Q_1 does not. Another interesting example is given by the processes $P_2 = a.(b.0 + c.0)$ and $Q_2 = a.b.0 + a.c.0$. They have the same traces, however $P_2 \neq_F Q_2$ since Q_2 has the failure $(a, \{c\})$ but P_2 does not.

- *Bisimilarity*. Here we first recall the strong version of the equivalence. Intuitively, P and Q are strongly bisimilar if whenever P performs an action **a** evolving into P' then Q can also perform **a** and evolve into a Q' strongly bisimilar to P', and similarly with P and Q interchanged.

  The above intuition can be formalized as follows. A symmetric relation B between process terms is said to be a strong *bisimulation* iff for all $(P, Q) \in B$,

  $$\text{if } P \xrightarrow{\mathbf{a}} P' \text{ then for some } Q',\ Q \xrightarrow{\mathbf{a}} Q' \text{ and } (P', Q') \in B.$$

  We say that P is *strongly bisimilar* to Q, written $P =_{SB} Q$ iff there exists a strong bisimulation containing the pair (P, Q).

  A weaker version of strong bisimilarity, called *weak bisimilarity* or simply *bisimilarity*, abstracts away from silent actions. Bisimilarity can be obtained by replacing the transitions $\xrightarrow{\mathbf{a}}$ above with the (sequences of observable) transitions $\overset{s}{\Longrightarrow}$ where $s \in \mathcal{L}^*$. We shall use $=_B$ to stand for (weak) bisimilarity. Notice that $P =_B \tau.P$ but $P \neq_{SB} \tau.P$.

  Bisimilarity is more discriminating than trace equivalence. It is easy to see that $=_B \subseteq =_T$. The usual example to see the strict inclusion is P_2 and Q_2 as given above. Also, bisimilarity is more discriminating than failures equivalence wrt the *branching* behavior (i.e., nondeterminism); take $P_3 = a.(b.c.0 + b.d.0)$ and $P_3 = a.b.c.0 + a.b.d.0$; they have the same failures but one can verify that $P_3 \neq_B Q_3$. However, failures equivalence is more discriminating than bisimilarity wrt *divergence* (i.e., the execution of infinite sequences of silent actions). Notice that the divergent process **Div**, with $\mathbf{Div} \overset{\text{def}}{=} \tau.\mathbf{Div}$, is bisimilar to the non-divergent $\tau.0$, however $\mathbf{Div} \neq_F \tau.0$ since $\tau.0$ has the failure $(\epsilon, \emptyset)$ but **Div** does not.

Denotational Semantics. The method was pioneered by Strachey and provided with a mathematical foundation by Scott. A denotational semantics interprets processes by using a function $[\![.]\!]$ which maps them into a more abstract mathematical object (typically, a structured set or a category). The map $[\![.]\!]$ is *compositional* in that the meaning of processes is determined from the meaning of its sub-processes.

A strategy for defining denotational semantics advocated in works such as [13] involves the identification of what can be observed of a process; what behavior is deemed relevant (e.g., failures, traces, divergence, deadlocks). A process is then equated with the set of observations that can be made of it. For example, if the observation is the traces of processes, the denotation of the prefix construct $\mathbf{a}.P$ can be defined as

$$[\![\mathbf{a}.P]\!] = \{\epsilon\} \cup \{\mathbf{a}.s \in \mathcal{L}^* \mid s \in [\![P]\!]\}$$

and the denotation of the summation can be defined as

$$[\![P + Q]\!] = [\![P]\!] \cup [\![Q]\!].$$

It easy to see that these denotations realize the operational intuition of traces; any trace of $\mathbf{a}.P$ is either empty or it starts with **a** followed by a trace of P; any trace of $P + Q$ is either a trace of P or one of Q. Note that the compositional nature is illustrated by stating the denotations of $\mathbf{a}.P$ and $P + Q$ in terms of those of P and Q.

Once the denotation has been defined one may ask whether it is in complete agreement with a corresponding operational notion. For example, for the trace denotation one would like the following correspondence wrt the operational notion of trace equivalence,

$$[\![P]\!] = [\![Q]\!] \text{ iff for all contexts } C,\ C[P] =_T C[Q]$$

(A *context* is an expression with a hole [.] such that placing a process in the hole produces a well-formed process term, e.g., if $C = R \parallel [.]$ then $C[P] = R \parallel P$.) If a denotational-operational agreement like the one above can be proven, we say that the denotation is *fully-abstract* [16] wrt the chosen operational notion.

Denotational semantics are more abstract than the operational ones in that they generally distant themselves from any specific implementation. However, the operational semantics approach is, in some informal sense, more elemental in that when developing a denotational semantics one usually has an operational semantics in mind.

Algebraic semantics. This method has been advocated by Baeten and Weijland [4] as well as Bergstra and Klop [5]. An algebraic semantics attempts to give meaning by stating a set of laws (or axioms) equating process terms. The processes and their operations are then interpreted as structures that obey these laws. As remarked by Baeten and Weijland [4], the algebraic approach answers the question “What is a process?” with a seemingly circular answer: “A process is something that obeys a certain set of axioms...for processes”.

As an example consider the following axioms for parallel composition:

$$P \parallel 0 \equiv P, \quad P \parallel Q \equiv Q \parallel P, \quad P \parallel (Q \parallel R) \equiv (P \parallel Q) \parallel R$$

In other words parallel composition is seen as a commutative, associative operator with 0 being the unit. Notice that the above axioms basically equate processes that are the same except for irrelevant syntactic differences, thus one may expect that any reasonable notion of equivalence validates them. But consider the following distribution axiom

$$\mathbf{a}.(P + Q) \equiv \mathbf{a}.P + \mathbf{a}.Q$$

This axiom is valid if we are content with trace equivalence, but not in general (e.g., it does not hold for failures equivalence or bisimilarity).

Given a set of algebraic laws, one may be interested in looking into the correspondence with a denotational semantics or with some operational notion of equivalence. An interesting property is whether the equalities derived from the laws are exactly those which hold for a natural notion of process equivalence. If this property holds, the set of algebraic laws is said to be *complete* wrt the notion of process equivalence under consideration.

In the algebraic approach one can simply *postulate* process equalities while in the operational (or denotational) approach one would need to *prove* them. On the advantages of postulation Russell [29] remarked the following:

> *The method of postulation has many advantages: they are the same as the advantages of theft over honest toil*
> — Bertrand Russell

Algebraic semantics, however, is a convenient framework for the study of process equivalences; postulating a set of laws, and then investigating the consistency of that set and what process equivalence it produces. Some frameworks (e.g., [19]) combine the operational semantics with the algebraic one by, for example, considering processes modulo the equivalence produced by a set of axioms.

2.3 Specification and Process Logics

One often is interested in verifying whether a given process satisfies a property, i.e., a specification. But process terms themselves specify behavior, so they can also be used to express specifications. Then this verification problem can be reduced to establishing whether the process and the specification process are related under some behavioral equivalence (or pre-order).

Hennessy-Milner's modal logic. Another way of expressing process specifications is by using a process logic. One such logic is the Hennessy-Milner's modal logic. The basic syntax of formulae is given by:

$$F \ := \mathtt{true} \mid \mathtt{false} \mid F_1 \wedge F_2 \mid F_1 \vee F_2 \mid \langle K \rangle\, F \mid [K]F$$

where K is a set of actions. Intuitively, the modality $\langle K \rangle\, F$, called *possibility*, asserts (of a given P) that: It is possible for P to do $\mathbf{a} \in K$ and then evolve into a

Q which satisfies F. The modality $[K]P$, called *necessity*, expresses that if P can do $\mathbf{a} \in K$ then it must thereby evolve into a Q which satisfies F.

Formally, the compliance of P with the specification F, written $P \models F$, is recursively given by:

$$\begin{array}{lll}
P \not\models \texttt{false} & & \\
P \models \texttt{true} & & \\
P \models F_1 \wedge F_2 & \text{iff} & P \models F_1 \text{ and } P \models F_2 \\
P \models F_1 \vee F_2 & \text{iff} & P \models F_1 \text{ or } P \models F_2 \\
P \models \langle K \rangle F & \text{iff} & \text{for some } Q,\ P \xrightarrow{\mathbf{a}} Q,\ \mathbf{a} \in K \text{ and } Q \models F \\
P \models [K]F & \text{iff} & \text{if } P \xrightarrow{\mathbf{a}} Q \text{ and } \mathbf{a} \in K \text{ then } Q \models F
\end{array}$$

As an example consider our familiar trace equivalent (but not bisimilar) processes $P_1 = a.(b.0 + c.0)$ and $P_2 = a.b.0 + a.c.0$. Notice that the formula

$$F = \langle\{a\}\rangle\,(\langle\{b\}\rangle\,\texttt{true} \wedge \langle\{c\}\rangle\,\texttt{true})$$

discriminates among them, i.e. $P_1 \models F$ but $P_2 \not\models F$. In fact the discriminating power of this logic wrt a finite processes (i.e., recursion-free processes) coincides with strong bisimilarity (see [32]). That is, two finite processes are strongly bisimilar iff they satisfy the same formulae in the Hennessy-Milner's logic. The result can be extended to image-finite processes by considering infinite disjunctions and conjunctions in the Hennessy-Milner's logic.

Temporal logics. The above logic can express local properties such as "an action must happen next" but it cannot express long-term properties such as "an action eventually happens". This kind of property, which falls into the category of *liveness properties* (expressing that "something good eventually happens"), and also *safety properties* (expressing that "something bad never happens") have been found to be useful for reasoning about concurrent systems. The modal logics attempting to capture properties of the kind above are often referred to as *temporal logics*.

Temporal logics were introduced into computer science by Pnueli [27] and thereafter proven to be a good basis for specification as well as for (automatic and machine-assisted) reasoning about concurrent systems. Temporal logics can be classified into linear and branching time logics. In the *linear* case at each moment there is only one possible future whilst in the *branching* case at each moment time may split into alternative futures.

Below we consider a very simple example of a linear-time temporal logic based on [15]. The syntax of the formulae is given by

$$F := \texttt{true} \mid \texttt{false} \mid L \mid F_1 \vee F_2 \mid F_1 \wedge F_2 \mid \Diamond F \mid \Box F$$

where L is a set of non-silent actions. The formulae of this logic express properties of sequences of non-silent actions; i.e. traces. For the sake of uniformity, we are interested only in infinite traces. Intuitively, the modality $\Diamond F$, pronounced *eventually* F, asserts of a given trace s that at some point in s, F holds. Similarly, $\Box F$, pronounced *always* F, asserts of a given trace s that in every point of s, F holds.

The models of the formulae are taken to be infinite sequence of actions; elements of Act^{ω}. Formally, the infinite sequence of actions $s = \mathbf{a}_1.\mathbf{a}_2 \ldots$ satisfies (or is a model of) F, written $s \models F$, iff $\langle s, 1\rangle \models F$, where

$$\begin{array}{lll}
\langle s,i\rangle \models \texttt{true} & & \\
\langle s,i\rangle \not\models \texttt{false} & & \\
\langle s,i\rangle \models L & \text{iff} & \mathbf{a}_i \in L \cup \tau \\
\langle s,i\rangle \models F_1 \vee F_2 & \text{iff} & \langle s,i\rangle \models F_1 \text{ or } \langle s,i\rangle \models F_2 \\
\langle s,i\rangle \models F_1 \wedge F_2 & \text{iff} & \langle s,i\rangle \models F_1 \text{ and } \langle s,i\rangle \models F_2 \\
\langle s,i\rangle \models \Box F & \text{iff} & \text{for all } j \geq i \;\; \langle s,j\rangle \models F \\
\langle s,i\rangle \models \Diamond F & \text{iff} & \text{there is a } j \geq i \;\text{ s.t. } \langle s,j\rangle \models F
\end{array}$$

Intuitively, P satisfies a linear-temporal specification F, written $P \models F$, iff all of its traces are models of F. Recall, however, that the traces are finite sequences of non-silent actions. But since formulae say nothing about silent actions, we can just interpret a finite trace s as the infinite sequence $\hat{s} = s.(\tau^{\omega})$ which results from s followed by infinitely many silent actions. This leads to the definition: $P \models F$ iff whenever $P \overset{s}{\Longrightarrow}$ then $\hat{s} \models F$.

Let us consider the definitions $A(a,b,c) \overset{\text{def}}{=} a.(b.A\langle a,b,c\rangle + c.A\langle a,b,c\rangle)$ and $B(a,b,c) \overset{\text{def}}{=} a.b.B\langle a,b,c\rangle + a.c.B\langle a,b,c\rangle$. Notice that the trace equivalent processes $A\langle a,b,c\rangle$ and $B\langle a,b,c\rangle$ satisfy the formula $\Box\Diamond(b \vee c)$; i.e. they always eventually do b or c. In general, for every two processes (finite or infinite) if they are trace equivalent then they satisfy exactly the same formulae of this temporal logic. The other direction does not hold in general since the logic is not powerful enough to express, for example, facts about the immediate (or next) future. Take the processes $a.a.0$ and $a.0$; they are not trace equivalent, but they satisfy the same formulae in this simple logic.

2.4 Analyzing Equivalences: Decidability and Congruence Issues

Much work in the theory of process calculi, and concurrency in general, involves the analysis of process equivalences. Let us say that our equivalence under consideration is denoted by $\sim$. Two typical questions that arise are:

1. Is $\sim$ decidable ?

2. Is $\sim$ a congruence ?

The first question refers to the issue as to whether there can be an algorithm that fully determines (or decides) for every P and Q if $P \sim Q$ or $P \nsim Q$. Since most process calculi can model Turing machines most natural equivalences are therefore undecidable. So, the interesting question is rather for what subclasses of processes is the equivalence decidable. For example, bisimilarity is undecidable for full CCS, but decidable for finite state processes (of course) and also for the families of infinite state processes including context-free processes [8], pushdown processes [31] and basic parallel processes [7]. Obviously, the decidability of an equivalence leads to another related issue: the complexity of verifying the equivalence.

The second question refers to the issue as to whether the fact that P and Q are ($\sim$) equivalent implies that they are still ($\sim$) equivalent in any context. The equivalence $\sim$ is a congruence if $P \sim Q$ implies $C[P] \sim C[Q]$ for every context C (as said before, a context C is an expression with a hole [.] such that placing a P in the hole yields a process term). The congruence issue is fundamental for algebraic as well as practical reasons; one may not be content with having $P \sim Q$ equivalent but $R \parallel P \nsim R \parallel Q$.

For example, trace equivalence and strong bisimilarity for the process language here considered is a congruence (see [18]) but weak bisimilarity is not because is not preserved by summation contexts. Notice that we have $b.0 =_B \tau.b.0$, but $a.0 + b.0 \neq_B a.0 + \tau.b.0$. In this case new questions arise: In what restricted sense is the equivalence a congruence? What contexts is the equivalence preserved by? What is the closest congruence to the equivalence? The answer to these questions may lead to a re-formulation of the operators. For instance, the problem with weak bisimilarity can be avoided by using a somewhat less liberal summation called guarded-summation (see [19]).

2.5 Process Calculi Variants

Given a process calculus it makes sense to consider variants of it (e.g., subclasses of processes, new process constructs, etc) to seek for simpler presentations of the calculus or different applications of it. Having these variants one can ask, for example, whether the process equivalences become simpler or harder to analyze (as argued in the previous section) or whether there is loss or gain of *expressive power*.

To compare *expressive power* one has to agree on what it means for a variant to be as expressive as the other. A natural way of doing this is by comparing wrt some process equivalence: If for every process P in one variant there is a Q in the other equivalent to P then way say that the latter variant is as expressive (wrt to the equivalence under consideration) as the former one.

Several studies of variants of CCS and their relative expressive power have been reported in [2]. Also several variants of the π-calculus (itself a generalization of CCS) have been compared wrt weak-bisimilarity (see [30]). An interesting result is that the π calculus construction $!P$ whose behavior is expressed by the law $!P \equiv P \,||!P$ can replace recursion without loss of expressive power. This is rather surprising since the syntax of $!P$ and its description are so simple. Another interesting result is that of Palamidessi [22] showing that under some reasonable assumptions the asynchronous version of the π-calculus is strictly less expressive than the synchronous one.

3 Conclusions

The λ-calculus is a canonical model of sequential computation. Unfortunately, there is no canonical model for concurrent computation at the present time. In spite of promising progress towards canonicity (e.g. [11, 21, 20]) an all-embracing theory of concurrency has yet to emerge. According to Petri [23] such a general model may attain a range of application comparable to that of physics. As argued in [18], however, even after the discovery of it, we shall need to choose different special models for different applications. Here is an analogy from [14]: Newtonian mechanics is not a suitable framework for describing the flow of fluids, for which one needs a theory containing mathematical concepts corresponding to friction and viscosity. Concurrency, as physics, is a field with a myriad of aspects for which we may require different terms of discussion and analysis.

References

[1] M. Abadi and A. D. Gordon. A calculus for cryptographic protocols: The spi calculus. In *Fourth ACM Conference on Computer and Communications Security*, pages 36–47. ACM Press, 1997.

[2] R. Amadio and C. Meyssonnier. On decidability of the control reachability problem in the asynchronous $\pi-$calculus. *Nordic Journal of Computing*, 9(2), 2002.

[3] J. C. M. Baeten and J. A. Bergstra. Real time process algebra. *Formal Aspects of Computing*, 3:142–188, 1991.

[4] J. C. M. Baeten and W. P. Weijland. *Process Algebra*. Cambridge University Press, 1990.

[5] J.A. Bergstra and J.W. Klop. Algebra of communicating processes with abstraction. *Theoretical Computer Science*, 37(1):77–121, 1985.

[6] J. R. Buchi. On a decision method in restricted second order arithmetic. In *Proc. Int. Cong. on Logic, Methodology, and Philosophy of Science*, pages 1–11. Stanford University Press, 1962.

[7] S. Christensen, H. Huttel, and F. Moller. Bisimulation equivalence is decidable for basic parallel processes. In *CONCUR'93*, LNCS 715, pages 143–157. Springer-Verlag, 1993.

[8] S. Christensen, H. Huttel, and C. Stirling. Bisimulation equivalence is decidable for all context-free processes. *Information and Computation*, 96:203–224, 1992.

[9] P. Cory, E. Hobart, and H. Tracy. Radar-based intruder detection for a robotic security system. In *Proc. of SPIE*, volume 3525, 1999.

[10] A. Grzegorczyk. On the definition of computable real continuous functions. *Fund. Math.*, 44:61–71, 1957.

[11] V. Gupta and V. Pratt. Gates accept concurrent behavior. In *Proc. 34th Annual IEEE Symp. on Foundations of Comp. Sci.*, pages 62–71. IEEE, 1993.

[12] C. A. R. Hoare. *Communications Sequential Processes*. Prentice-Hall, Englewood Cliffs (NJ), USA, 1985.

[13] C.A.R. Hoare. Let's make models. In *Proc. of CONCUR '90*, volume 458 of *LNCS*. Springer-Verlag, 1990.

[14] L. Lamport. Answer to Pratt. Concurrency Mailing List Archive, 19, Nov 1990. Availabe via http://www-i2.informatik.rwth-aachen.de/Research/MCS/Mailing_List_archive/.

[15] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems, Specification*. Springer, 1991.

[16] R. Milner. Processes; a mathematical model of computing agents. In *Proc. Logic Colloquium 73, eds.*, pages 257–274, 1973.

[17] R. Milner. *Communication and Concurrency*. International Series in Computer Science. Prentice Hall, 1989. SU Fisher Research 511/24.

[18] R. Milner. *Operational and Algebraic Semantics of Concurrent Processes*, pages 1203–1241. Elsevier, 1990.

[19] R. Milner. *Communicating and Mobile Systems: the π-calculus*. Cambridge University Press, 1999.

[20] R. Milner. Bigraphical reactive systems. In *Proc. of CONCUR '02*, volume 2154 of *LNCS*. Springer-Verlag, 2002.

[21] U. Montanari and F. Rossi. Graph rewriting and constraint solving for modelling distributed systems with synchronization. In *Proc. of COORDINATION '96*, volume 1061 of *LNCS*, pages 12–27. Springer-Verlag, 1996.

[22] C. Palamidessi. Comparing the expressive power of the synchronous and the asynchronous pi-calculus. In ACM Press, editor, *POPL'97*, pages 256–265, 1997.

[23] C.A. Petri. Fundamentals of a theory of asynchronous information flow. In *Proc. IFIP Congress '62*, 1962.

[24] G. Plotkin. A structural approach to operational semantics. Technical Report FN-19, DAIMI, University of Aarhus, 1981.

[25] G. Plotkin. The origins of structural operational semantics. *Journal of Logic and Algebraic Programming*, 60-61:3–15, 2004.

[26] G. Plotkin. A structural approach to operational semantics. *Journal of Logic and Algebraic Programming*, 60-61:17–139, 2004.

[27] A. Pnueli. The temporal logic of programs. In *Proc. of the 18th IEEE Symposium on the Foundations of Computer Science (FOCS-77)*, pages 46–57. IEEE, IEEE Computer Society Press, 1977.

[28] G.M. Reed and A.W. Roscoe. A timed model for communication sequential processes. *Theoretical Computer Science*, 8:249–261, 1988.

[29] B. Russell. Introduction to Mathematical Philosophy, 1919.

[30] D. Sangiorgi and D. Walker. *The $\pi-$calculus: A Theory of Mobile Processes*. Cambridge University Press, 2001.

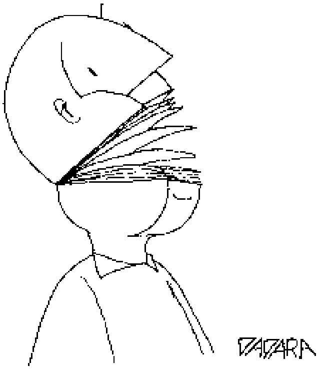
[31] C. Stirling. Decidability of bisimulation equivalence for normed pushdown processes. In *CONCUR'96*, LNCS 1119, pages 217–232. Springer-Verlag, 1996.

[32] C. Stirling. Bisimulation, model checking and other games. Notes for Mathfit Instructural Meeting on Games and Computation, 1998.

[33] W. Yi. *A Calculus for Real Time Systems*. PhD thesis, Chalmers University of Technology, 1991.

[34] W. M. Zuberek. Timed petri nets and preliminary performance evaluation. In *Proc. of the 7th Annual Symposium on Computer Architecture*, pages 88–96. ACM and IEEE, 1980.

Technical Contributions



The Freudenthal problem and its ramifications (Part I)

Axel Born [*] Cor A.J. Hurkens [†] Gerhard J. Woeginger [†]

Abstract

This is the first article (in a series of three) dedicated to the many variants and variations of the so-called Freudenthal problem. The Freudenthal problem is a mathematical puzzle in which the reader deduces two secret integers from several rounds of communication between two persons. One person knows the sum of the two secret integers, while the other person knows the product. The current article surveys some of the most basic variants of the Freudenthal problem.

1 The Freudenthal problem

Hans Freudenthal (1905-1990) studied mathematics at the University of Berlin in the 1920s. He completed his Ph.D. thesis *"Über die Enden topologischer Räume und Gruppen"* under the supervision of Heinz Hopf in 1930. Around that time, he moved to the Netherlands where he worked with Luitzen Brouwer and soon became a lecturer at the University of Amsterdam. As a Jew, Freudenthal survived the period of German occupation unharmed, since he was married to an Arian Dutch woman and since he had lots of luck. In 1946, Freudenthal was offered the chair of pure and applied mathematics at the University of Utrecht. He held this chair until he retired in 1975. Freudenthal's scientific contributions mainly fall into topology, geometry, and the theory of Lie groups. Freudenthal is also remembered and recognized for his numerous contributions to mathematical education and didactics. The institute for innovation and improvement of mathematics education at the University of Utrecht is named after him the *"Freudenthal Institute"*.

In 1969, Hans Freudenthal [2] posed the following puzzle in the problem section of the Dutch mathematics journal *Nieuw Archief voor Wiskunde* (= New

[*] Oberstufen-Realgymnasium Ursulinen, Leonhardstrasse 62, 8010 Graz, Austria.

[†] Email: `{wscor|gwoegi}@win.tue.nl`. Department of Mathematics and Computer Science, TU Eindhoven, P.O. Box 513, 5600 MB Eindhoven, The Netherlands.

Bulletin of the EATCS no 90, pp. 175–191, October 2006

archive for mathematics). The original formulation of the puzzle is in Dutch. Here is our free translation:

> The teacher says to Peter and Sam: I have secretly chosen two integers x and y with $2 \le x < y$ and $x + y \le 100$. I have told the sum $s = x + y$ to Sam (but not to Peter) and the product $p = xy$ to Peter (but not to Sam).
>
> 1. Peter says: I don't know the numbers x and y.
> 2. Sam replies: I already knew you didn't know.
> 3. Peter says: Oh, then I do know the numbers x and y.
> 4. Sam says: Oh, then I also know them.
>
> Determine x and y!

At first sight, the given information just cannot be enough for determining the two numbers... The Freudenthal problem was introduced to the English speaking world in 1976, when David Sprows stated it in the problem section of the Mathematics Magazine [7]. In December 1979, Martin Gardner [4] posed the Freudenthal problem in his mathematical entertainments column in the Scientific American. He writes: *"I call this beautiful problem impossible, because it seems to lack sufficient information for a solution."* And indeed, nowadays the Freudenthal problem sometimes shows up under the name *"The impossible problem"*; see for instance Sallows [6]. Edsger Dijkstra [1] reports that he once solved a variant of the Freudenthal problem during a sleepless night in 1978, when he was jetlagged. He states that it took him almost six hours, and that he solved it in his head, without using paper or pencil.

In this article, we want to discuss some of the most basic Freudenthal variants. We will mainly concentrate on two classes of variants, that are built around the following definitions. Consider two positive integers m and M with $m \le M$, and define the following sets:

> $Z^{\neq}(m, M)$ contains all pairs (x, y) with $m \le x < y$ and $x + y \le M$.
>
> $Z^{=}(m, M)$ contains all pairs (x, y) with $m \le x \le y$ and $x + y \le M$.

In the Freudenthal variant $\textsc{Freudenthal}^{\neq}(m, M)$ the introductory words of the teacher state that the secret number pair (x, y) is taken from $Z^{\neq}(m, M)$. In the Freudenthal variant $\textsc{Freudenthal}^{=}(m, M)$ the introductory words of the teacher state that the secret number pair (x, y) is taken from $Z^{=}(m, M)$. In both variants, the announcement of the teacher is followed by the above four-line conversation between Sam and Peter. Note that the problem originally posed by Hans Freudenthal is $\textsc{Freudenthal}^{\neq}(2, 100)$.

2 The algorithm of Denniston

The *Nieuw Archief voor Wiskunde* [3] lists the names of seventeen readers who submitted correct solutions for the Freudenthal problem; interestingly, two of the names on this list are *J. van Leeuwen* and *J.H. van Lint*. Among other solutions, [3] discusses a simple computational approach by Ralph Hugh Francis Denniston. Although we will only formulate Denniston's approach for problem $\textsc{Freudenthal}^{\neq}(m, M)$, it obviously generalizes to other Freudenthal variants.

Initialization. Introduce a matrix A where the rows p correspond to the products and the columns s correspond to the sums.
Set entry $A[p, s]$ to $+$, if there exist integers x, y with $(x, y) \in Z^{\neq}(m, M)$ that satisfy $x + y = s$ and $xy = p$. Otherwise, set $A[p, s]$ to $-$.

Step 1. Wherever a row p contains just a single $+$ entry, replace this entry by 1. (This product p contradicts statement #1 by Peter.)

Step 2. Wherever a column s contains some 1 entry, replace all $+$ entries in this column by 2. (This sum s contradicts statement #2 by Sam.)

Step 3. Wherever a row p contains two or more $+$ entries, replace them by 3. (This product p contradicts statement #3 by Peter.)

Step 4. Wherever a column s contains two or more $+$ entries, replace them by 4. (This sum s contradicts statement #4 by Sam.)

Output. The remaining $+$ entries specify all sum/product combinations that agree with the full conversation. A $+$ entry in row p_0 and column s_0 means that the values s_0 and p_0 are sum and product of the secret numbers x and y.

If in the end there is a single remaining $+$ entry, then the Freudenthal problem has a unique solution. If there is more than one remaining $+$ entry, then the problem has several possible solutions; Sam and Peter are able to determine x and y from the conversation (and from their private knowledge of s or p), whereas the reader is not. If there are no remaining $+$ entries, then the problem formulation is contradictory.

Some more notation: We say that a sum s and a product p are *compatible* (with respect to some fixed Freudenthal problem that usually is clear from the context), if the initialization step of Denniston's algorithm sets entry $A[p, s]$ to $+$. During an execution of Denniston's algorithm, a row or a column is called *alive* if it contains at least one $+$ entry. We denote by $\mathcal{P}_1$ the set of rows/products p that are alive after Step 1; note that these products are in agreement with statement #1. Similarly, we denote by $\mathcal{S}_2$ the set of columns/sums s that are alive after Step 2 (and that agree with statements #1 and #2), we denote by $\mathcal{P}_3$ the set of rows/products p that are alive after Step 3 (and that agree with statements #1, #2, and #3), and we denote by $\mathcal{S}_4$ the set of columns/sums s that are alive after Step 4 (and that agree with the full conversation).

3 The Freudenthal problem with m=1 and M=11

We now take a closer look at Freudenthal$^{\neq}(1,11)$ and Freudenthal$^{=}(1,11)$, which behave surprisingly different from each other.

Table 1 summarizes Denniston's algorithm for Freudenthal$^{\neq}(1,11)$. This puzzle is contradictory and ill-posed: Statement #1 yields $\mathcal{P}_1 = \{6,8,10,12,18,24\}$, and statement #2 gives $\mathcal{S}_2 = \{7\}$. In statement #3, Peter determines x and y from his product p and from $s = 7$. This makes $\mathcal{P}_3 = \{6,10,12\}$, and leaves us with the three possibilities $(1,6)$, $(2,5)$, and $(3,4)$ for (x,y). Sam cannot make statement #4, as there is no way for him to identify the correct product from $s = 7$ and $\mathcal{P}_3$.

Table 2 demonstrates that problem Freudenthal$^{=}(1,11)$ is well-posed and has a unique solution. Since $x = y$ is legal in this variant, statement #1 now yields $\mathcal{P}_1 = \{4,6,8,9,10,12,16,18,24\}$. Statement #2 restricts the sum to $\mathcal{S}_2 = \{5,7\}$. In statement #3, Peter determines x and y from his product: The product cannot be 6, since then Peter could not distinguish $x = 2, y = 3, s = 5$ from $x = 1, y = 6, s = 7$. Therefore $\mathcal{P}_3 = \{4,10,12\}$. Finally, statement #4 implies $s \neq 7$, since otherwise Sam could not distinguish between $p = 10$ and $p = 12$. Therefore $s = 5$ and $p = 4$, which yields $x = 1$ and $y = 4$.

4 An analysis of the classical Freudenthal problem

We now want to get some understanding how Denniston's algorithm behaves for the classical Freudenthal problem Freudenthal$^{\neq}(2,100)$.

The set $\mathcal{P}_1$ is listed in Table 3; it consists of 574 elements, but has a rather primitive structure: Every element $p \in \mathcal{P}_1$ possesses at least two factorizations $p = xy$ with $(x,y) \in Z^{\neq}(2,100)$. Here are some simple rules for excluding certain products from $\mathcal{P}_1$: First, any product of two prime numbers is not in $\mathcal{P}_1$. Secondly, any p with a prime factor greater than 50 is not in $\mathcal{P}_1$. Next, any number of the form $p = q^3$ with prime q is not in $\mathcal{P}_1$; otherwise, Peter would deduce $x = q$ and $y = q^2$ right at the beginning. Finally, any number of the form $p = 2q^2$ with a prime $q > 10$ is not in $\mathcal{P}_1$; otherwise, Peter could deduce $x = q$ and $y = 2q$.

Next, let us investigate the structure of set $\mathcal{S}_2$. For $s \in \mathcal{S}_2$, all compatible products $x(s-x)$ must lie in $\mathcal{P}_1$. Hence, the following values of s are not contained in $\mathcal{S}_2$:

$p\backslash s$	2	3	4	5	6	7	8	9	10	11
1	–	–	–	–	–	–	–	–	–	–
2	–	1	–	–	–	–	–	–	–	–
3	–	–	1	–	–	–	–	–	–	–
4	–	–	–	1	–	–	–	–	–	–
5	–	–	–	–	1	–	–	–	–	–
6	–	–	–	2	–	4	–	–	–	–
7	–	–	–	–	–	–	1	–	–	–
8	–	–	–	–	2	–	–	2	–	–
9	–	–	–	–	–	–	–	–	1	–
10	–	–	–	–	–	4	–	–	–	2
11	–	–	–	–	–	–	–	–	–	–
12	–	–	–	–	–	4	2	–	–	–
13	–	–	–	–	–	–	–	–	–	–
14	–	–	–	–	–	–	–	1	–	–
15	–	–	–	–	–	–	1	–	–	–
16	–	–	–	–	–	–	–	–	1	–
17	–	–	–	–	–	–	–	–	–	–
18	–	–	–	–	–	–	–	2	–	2
19	–	–	–	–	–	–	–	–	–	–
20	–	–	–	–	–	–	–	1	–	–
21	–	–	–	–	–	–	–	–	1	–
22	–	–	–	–	–	–	–	–	–	–
23	–	–	–	–	–	–	–	–	–	–
24	–	–	–	–	–	–	–	–	2	2
25	–	–	–	–	–	–	–	–	–	–
26	–	–	–	–	–	–	–	–	–	–
27	–	–	–	–	–	–	–	–	–	–
28	–	–	–	–	–	–	–	–	–	1
29	–	–	–	–	–	–	–	–	–	–
30	–	–	–	–	–	–	–	–	–	1

Table 1: The outcome of Denniston's algorithm for $\textsc{Freudenthal}^{\neq}(1, 11)$.

$p\backslash s$	2	3	4	5	6	7	8	9	10	11
1	1	–	–	–	–	–	–	–	–	–
2	–	1	–	–	–	–	–	–	–	–
3	–	–	1	–	–	–	–	–	–	–
4	–	–	2	+	–	–	–	–	–	–
5	–	–	–	–	1	–	–	–	–	–
6	–	–	–	3	–	3	–	–	–	–
7	–	–	–	–	–	–	1	–	–	–
8	–	–	–	–	2	–	–	2	–	–
9	–	–	–	–	2	–	–	–	2	–
10	–	–	–	–	–	4	–	–	–	2
11	–	–	–	–	–	–	–	–	–	–
12	–	–	–	–	–	4	2	–	–	–
13	–	–	–	–	–	–	–	–	–	–
14	–	–	–	–	–	–	–	1	–	–
15	–	–	–	–	–	–	1	–	–	–
16	–	–	–	–	–	–	2	–	2	–
17	–	–	–	–	–	–	–	–	–	–
18	–	–	–	–	–	–	–	2	–	2
19	–	–	–	–	–	–	–	–	–	–
20	–	–	–	–	–	–	–	1	–	–
21	–	–	–	–	–	–	–	–	1	–
22	–	–	–	–	–	–	–	–	–	–
23	–	–	–	–	–	–	–	–	–	–
24	–	–	–	–	–	–	–	–	2	2
25	–	–	–	–	–	–	–	–	1	–
26	–	–	–	–	–	–	–	–	–	–
27	–	–	–	–	–	–	–	–	–	–
28	–	–	–	–	–	–	–	–	–	1
29	–	–	–	–	–	–	–	–	–	–
30	–	–	–	–	–	–	–	–	–	1

Table 2: The outcome of Denniston's algorithm for FREUDENTHAL$^{=}(1, 11)$.

12,	18,	20,	24,	28,	30,	32,	36,	40,	42,
44,	45,	48,	50,	52,	54,	56,	60,	63,	64,
66,	68,	70,	72,	75,	76,	78,	80,	84,	88,
90,	92,	96,	98,	99,	100,	102,	104,	105,	108,
110,	112,	114,	116,	117,	120,	124,	126,	128,	130,
132,	135,	136,	138,	140,	144,	147,	148,	150,	152,
153,	154,	156,	160,	162,	164,	165,	168,	170,	171,
172,	174,	175,	176,	180,	182,	184,	186,	188,	189,
190,	192,	195,	196,	198,	200,	204,	207,	208,	210,
216,	220,	222,	224,	225,	228,	230,	231,	232,	234,
238,	240,	243,	245,	246,	248,	250,	252,	255,	256,
258,	260,	261,	264,	266,	270,	272,	273,	275,	276,
279,	280,	282,	285,	286,	288,	290,	294,	296,	297,
300,	304,	306,	308,	310,	312,	315,	320,	322,	324,
325,	328,	330,	336,	340,	342,	344,	345,	348,	350,
351,	352,	357,	360,	364,	368,	370,	372,	374,	375,
376,	378,	380,	384,	385,	390,	392,	396,	399,	400,
405,	406,	408,	410,	414,	416,	418,	420,	425,	429,
430,	432,	434,	435,	440,	441,	442,	444,	448,	450,
455,	456,	459,	460,	462,	464,	465,	468,	470,	475,
476,	480,	483,	486,	490,	492,	494,	495,	496,	500,
504,	506,	510,	512,	513,	516,	518,	520,	522,	525,
528,	532,	539,	540,	544,	546,	550,	552,	558,	560,
561,	564,	567,	570,	572,	574,	576,	580,	585,	588,
592,	594,	595,	598,	600,	602,	608,	609,	612,	616,
620,	621,	624,	627,	630,	637,	638,	640,	644,	646,
648,	650,	651,	656,	660,	663,	666,	672,	675,	680,
682,	684,	688,	690,	693,	696,	700,	702,	704,	714,
715,	720,	726,	728,	735,	736,	738,	740,	741,	744,
748,	750,	754,	756,	759,	760,	765,	768,	770,	774,
780,	782,	783,	784,	792,	798,	800,	806,	810,	812,
814,	816,	819,	820,	825,	828,	832,	836,	840,	850,
855,	858,	860,	864,	868,	870,	874,	880,	882,	884,
888,	891,	896,	897,	900,	902,	910,	912,	918,	920,
924,	928,	930,	935,	936,	945,	946,	950,	952,	957,
960,	962,	966,	968,	969,	972,	975,	980,	984,	986,
988,	990,	992,	1000,	1008,	1012,	1014,	1020,	1026,	1032,
1035,	1036,	1040,	1044,	1050,	1053,	1054,	1056,	1064,	1066,
1071,	1078,	1080,	1088,	1092,	1100,	1102,	1104,	1105,	1110,
1116,	1118,	1120,	1122,	1125,	1131,	1134,	1140,	1144,	1148,
1150,	1152,	1155,	1160,	1170,	1173,	1176,	1178,	1184,	1188,
1190,	1196,	1197,	1200,	1204,	1215,	1216,	1218,	1224,	1230,
1232,	1240,	1242,	1248,	1254,	1258,	1260,	1275,	1276,	1280,
1288,	1292,	1296,	1300,	1302,	1311,	1312,	1320,	1323,	1326,
1330,	1332,	1334,	1344,	1350,	1360,	1364,	1365,	1368,	1377,
1380,	1386,	1392,	1394,	1400,	1404,	1406,	1408,	1425,	1426,
1428,	1430,	1440,	1449,	1450,	1452,	1456,	1458,	1470,	1472,
1476,	1480,	1482,	1485,	1488,	1496,	1500,	1508,	1512,	1518,
1520,	1530,	1536,	1539,	1540,	1550,	1554,	1560,	1564,	1566,
1568,	1575,	1584,	1596,	1600,	1610,	1612,	1617,	1620,	1624,
1628,	1632,	1638,	1650,	1656,	1664,	1672,	1674,	1680,	1700,
1702,	1710,	1716,	1725,	1728,	1736,	1740,	1748,	1750,	1755,
1760,	1764,	1768,	1776,	1782,	1792,	1794,	1798,	1800,	1820,
1824,	1836,	1848,	1850,	1856,	1860,	1872,	1890,	1904,	1914,
1920,	1924,	1932,	1938,	1944,	1950,	1960,	1972,	1980,	1984,
2016,	2030,	2040,	2046,	2052,	2070,	2080,	2100,	2108,	2112,
2142,	2145,	2160,	2176,	2184,	2200,	2205,	2240,	2244,	2268,
2280,	2340,	2352,	2400.						

Table 3: The set $\mathcal{P}_1$ for FREUDENTHAL$^{\neq}(2, 100)$.

- $55 \leq s \leq 100$: For $x = 53$ and $y = s - 53$, the product xy is not in $\mathcal{P}_1$.
- $s = 6$: The product of $x = 2$ and $y = 4$ is not in $\mathcal{P}_1$.
- $s = 51$: The product of $x = 17$ and $y = 34$ equals $2 \cdot 17^2$, and is not in $\mathcal{P}_1$.
- $8 \leq s \leq 54$, and s even: Then s can be written as the sum of two distinct, odd primes x and y; hence the corresponding product xy is not in $\mathcal{P}_1$.
- $5 \leq s \leq 53$, and $s = q + 2$ for a prime q: The product of $x = 2$ and $y = q$ is not in $\mathcal{P}_1$.

This leaves us with the ten numbers $11, 17, 23, 27, 29, 35, 37, 41, 47, 53$ as candidates for $\mathcal{S}_2$. The following lines enumerate the compatible products for every candidate:

s=11: 18, 24, 28, 30.

s=17: 30, 42, 52, 60, 66, 70, 72.

s=23: 42, 60, 76, 90, 102, 112, 120, 126, 130, 132.

s=27: 50, 72, 92, 110, 126, 140, 152, 162, 170, 176, 180, 182.

s=29: 54, 78, 100, 120, 138, 154, 168, 180, 190, 198, 204, 208, 210.

s=35: 66, 96, 124, 150, 174, 196, 216, 234, 250, 264, 276, 286, 294, 300, 304, 306.

s=37: 70, 102, 132, 160, 186, 210, 232, 252, 270, 286, 300, 312, 322, 330, 336, 340, 342.

s=41: 78, 114, 148, 180, 210, 238, 264, 288, 310, 330, 348, 364, 378, 390, 400, 408, 414, 418, 420.

s=47: 90, 132, 172, 210, 246, 280, 312, 342, 370, 396, 420, 442, 462, 480, 496, 510, 522, 532, 540, 546, 550, 552.

s=53: 102, 150, 196, 240, 282, 322, 360, 396, 430, 462, 492, 520, 546, 570, 592, 612, 630, 646, 660, 672, 682, 690, 696, 700, 702.

Since all listed products are in $\mathcal{P}_1$, we conclude that set $\mathcal{S}_2$ consists of 11, 17, 23, 27, 29, 35, 37, 41, 47, 53.

We turn to set $\mathcal{P}_3$. A product p is in $\mathcal{P}_3$, if and only if it is compatible with precisely one of the sums in $\mathcal{S}_2$; this means that p shows up in exactly one of the ten enumerations listed above. For instance, the three products 18, 24, 28 only show up for $s = 11$, and hence are contained in $\mathcal{P}_3$. The product 30 shows up once for $s = 11$ and once for $s = 17$, and hence is not in $\mathcal{P}_3$. Here is a cleaned-up version of the above enumerations, that only lists the values in $\mathcal{P}_3$:

s=11: 18, 24, 28.

s=17: 52.

s=23: 76, 112, 130.

s=27: 50, 92, 110, 140, 152, 162, 170, 176, 182.

s=29: 54, 100, 138, 154, 168, 190, 198, 204, 208.

s=35: 96, 124, 174, 216, 234, 250, 276, 294, 304, 306.

s=37: 160, 186, 232, 252, 270, 336, 340.

s=41: 114, 148, 238, 288, 310, 348, 364, 378, 390, 400, 408, 414, 418.

s=47: 172, 246, 280, 370, 442, 480, 496, 510, 522, 532, 540, 550, 552.

s=53: 240, 282, 360, 430, 492, 520, 570, 592, 612, 630, 646, 660, 672, 682, 690, 696, 700, 702.

Finally, we derive $\mathcal{S}_4 = \{17\}$, since the set $\mathcal{P}_3$ contains two or more compatible products for each sum $s \in \mathcal{S}_2$, except for $s = 17$. Hence, $s = 17$ and $p = 52$ with $x = 4$ and $y = 13$ form the unique solution to the classical Freudenthal problem.

5 Stable solutions and phantom solutions for m=2

Martin Gardner [4] attempted to simplify the classical Freudenthal problem for his Scientific American column: He reduced the feasible region to the smaller region $2 \leq x, y \leq 20$, which is easier to handle but still safely contains the numbers 4 and 13 of the supposed solution. This simplification turned out to be fatal, and hundreds of readers pointed out that Gardner's modified problem has no solution at all. In this section, we will discuss problem $\textsc{Freudenthal}^{\neq}(2, M)$ under varying feasible regions, when the bound M grows and tends to infinity.

We will write $\mathcal{P}_1(M)$, $\mathcal{S}_2(M)$, $\mathcal{P}_3(M)$, $\mathcal{S}_4(M)$ to stress that these concepts now depend on M (whereas $m = 2$ is fixed). For a product p, we denote by $\mathcal{M}_1(p)$ respectively $\mathcal{M}_3(p)$ the set of all bounds M with $p \in \mathcal{P}_1(M)$ respectively $p \in \mathcal{P}_3(M)$. Similarly, for a sum s, we denote by $\mathcal{M}_2(s)$ respectively $\mathcal{M}_4(s)$ the set of all bounds M with $s \in \mathcal{S}_2(M)$ respectively $s \in \mathcal{S}_4(M)$. An *interval* $[\ell, r]$ consists of all integers M with $\ell \leq M \leq r$, and a *half-line* $[\ell, \infty]$ of all M with $\ell \leq M$.

Theorem 1. *For any sum s and any product p, the following holds true.*

(a) $\mathcal{M}_1(p)$ is either empty or a half-line.

(b) $\mathcal{M}_2(s)$ is either empty or a half-line.

(c) $\mathcal{M}_3(p)$ is either empty or a half-line or an interval.

(d) $\mathcal{M}_4(s)$ is either empty or a half-line or an interval.

Proof. Throughout we will ignore the trivial cases where the considered set is empty.

(a) A product p is in $\mathcal{P}_1(M)$, if and only if it has at least two distinct factorizations under the bound M. The claim now follows from $Z^{\neq}(2, M) \subseteq Z^{\neq}(2, M+1)$.

(b) A sum s lies in $\mathcal{S}_2(M)$, if and only if all compatible products $x(s-x)$ are in $\mathcal{P}_1(M)$. By (a), this is the case if and only if M lies in the intersection of the corresponding half-lines $\mathcal{M}_1(x(s-x))$. This intersection is again a half-line.

(c) A product p lies in $\mathcal{P}_3(M)$, if and only if exactly one of its compatible sums $x + p/x$ lies in $\mathcal{S}_2(M)$. By (b), this is the case if and only if M lies in exactly one of the half-lines $\mathcal{M}_2(x + p/x)$, say in the half-line corresponding to sum $s(p)$. If there are no other half-lines involved, then $\mathcal{M}_3(p)$ coincides with the half-line $\mathcal{M}_2(s(p))$. If there are other half-lines involved, then $\mathcal{M}_3(p)$ is the interval that goes from the endpoint of $\mathcal{M}_2(s(p))$ to the leftmost endpoint of the remaining half-lines. Note that in either case the left endpoint of $\mathcal{M}_3(p)$ coincides with the left endpoint of $\mathcal{M}_2(s(p))$.

(d) Let s be an arbitrary sum. Assume that the products p_a and p_b both are compatible with s, and that there exist two values $M_a, M_b \in \mathcal{M}_2(s)$ such that $p_a \in \mathcal{P}_3(M_a)$ and $p_b \in \mathcal{P}_3(M_b)$. Then the discussion under (c) yields that $s(p_a) = s(p_b) = s$, and that furthermore the left endpoints of $\mathcal{M}_3(p_a)$ and $\mathcal{M}_3(p_b)$ both coincide with the left endpoint of $\mathcal{M}_2(s)$.

A sum s is in $\mathcal{S}_4(M)$, if and only if exactly one of its compatible products $x(s-x)$ lies in $\mathcal{P}_3(M)$. By (c), this is the case if and only if M lies in exactly one of the corresponding half-lines or intervals. By the above discussion, the left endpoints of all these half-lines and intervals coincide with the left endpoint of $\mathcal{M}_2(s)$. Then $\mathcal{M}_4(s)$ is the region covered by exactly one of these half-lines and intervals, and is again a half-line or an interval (or is empty). ■

For a pair (x, y), we denote by $\mathcal{M}(x, y)$ the set of all integers M for which (x, y) is a solution to FREUDENTHAL$^{\neq}(2, M)$. Theorem 1 yields that $\mathcal{M}(x, y)$ is either a half-line or an interval. We call (x, y) a *stable* solution, if $\mathcal{M}(x, y)$ is a half-line, and we call it a *phantom* solution, if $\mathcal{M}(x, y)$ is an interval. For instance, the pair $(67, 82)$ is a phantom solution that is only active for the range $4.721 \leq M \leq 5.485$.

Theorem 2. *The pair* $(4, 13)$ *forms a stable solution for* FREUDENTHAL$^{\neq}(2, *)$. *The set* $\mathcal{M}(4, 13)$ *consists of all* $M \geq 65$.

Proof. First, we discuss the cases with $M \geq 65$. It is easily verified that the six sums 11, 17, 23, 27, 35, 37 are contained in $\mathcal{S}_2(65)$. Theorem 1.(b) implies that these six sums are also contained in all sets $\mathcal{S}_2(M)$ with $M \geq 65$. As a consequence, the set $\mathcal{P}_3(M)$ does not contain any of the following six products: $30 = 5 \cdot 6 = 2 \cdot 15$; $42 = 2 \cdot 21 = 3 \cdot 14$; $60 = 3 \cdot 20 = 4 \cdot 15$; $66 = 2 \cdot 33 = 6 \cdot 11$; $70 = 2 \cdot 35 = 7 \cdot 10$; and $72 = 3 \cdot 24 = 8 \cdot 9$. On the other hand the product $52 = 4 \cdot 13 = 2 \cdot 26$ lies in $\mathcal{P}_3(M)$, since $17 \in \mathcal{S}_2(M)$ and $28 \notin \mathcal{S}_2(M)$. We now derive $17 \in \mathcal{S}_4(M)$ from this: The sum 17 can be written as 2 + 15, 3 + 14, $4+13$, $5+12$, $6+11$, $7+10$, and $8+9$ with corresponding products 30, 42, 52, 60, 66, 70, and 72. Since exactly one of these products lies in $\mathcal{P}_3(M)$, the pair $(4, 13)$ indeed forms a solution for $M \geq 65$. Next, we discuss cases $M \leq 64$. We claim that neither 19 nor 37 is in $\mathcal{S}_2(M)$:

- $2 \cdot 17 \notin \mathcal{P}_1(M)$ implies $19 = 2 + 17 \notin \mathcal{S}_2(M)$.
- $186 \notin \mathcal{P}_1(M)$, since only $186 = 6 \cdot 31$ can be a legal factorization for $M \leq 64$. (In particular the factorization $186 = 3 \cdot 62$ with sum $3 + 62 > M$ is not legal.) Then $186 = 6 \cdot 31 \notin \mathcal{P}_1(M)$ implies $6 + 31 = 37 \notin \mathcal{S}_2(M)$.

Now suppose for the sake of contradiction that the pair $(4, 13)$ forms a solution. Then $17 \in \mathcal{S}_2(M)$ and $52 \in \mathcal{P}_3(M)$. Since the factorizations of 70 are $2 \cdot 35$, $5 \cdot 14$, and $7 \cdot 10$, and since exactly one of the corresponding sums 37, 19, 17 lies in $\mathcal{S}_2(M)$, we get $70 \in \mathcal{P}_3(M)$. Since $\mathcal{P}_3(M)$ contains two products $52 = 4 \cdot 13$ and $70 = 7 \cdot 10$ compatible with the sum 17, we get $17 \notin \mathcal{S}_4(M)$. Hence, the pair $(4, 13)$ cannot be a solution for $M \leq 64$. ■

The pair $(4, 13)$ is actually the *unique* solution of Freudenthal$^{\neq}(2, M)$ for $65 \leq M \leq 1.684$. For $M \leq 64$ there are no solutions, and for $M \geq 1.685$ the pair $(4, 61)$ forms a second solution. Martin Gardner conjectured in private correspondence with John Kiltinen and Peter Young (mentioned in the introduction of [5]) that the number of solution pairs for Freudenthal$^{\neq}(2, *)$ should be infinite. To the best of our knowledge, this conjecture is still open. We propose the following slight strengthening.

Conjecture 3. Freudenthal$^{\neq}(2, *)$ *has infinitely many stable solutions.*

Many stable solutions for Freudenthal$^{\neq}(2, *)$ contain a power of 2, but not all of them do: The pair $(201, 556)$ is a stable solution that is active for all $M \geq 966.293$. Section 8 provides additional information on stable solutions for Freudenthal$^{\neq}(2, *)$.

6 A meta-variant of Freudenthal

In September 2000, Clive Tooth created a kind of Meta-Freudenthal problem, and posed it to the readers of the newsgroup `sci.math` on the Usenet. We present it in a slightly modified form that is built around the solutions of problem Freudenthal$^{=}(2, 5.000)$.

> The teacher says to Peter and Sam: I have secretly chosen two integers x and y with $2 \leq x \leq y$ and $x+y \leq 5.000$. I have told their sum $s = x+y$ only to Sam and their product $p = xy$ only to Peter.

1. Peter says: I don't know the numbers x and y.
2. Sam replies: I already knew you didn't know.
3. Peter says: Oh, then I do know the numbers x and y.
4. Sam says: Oh, then I also know them.

Up to this point, John has listened quietly to the conversation.

5. John complains: But I still don't know the numbers x and y.
6. Sam replies: But if we told you the value x, then you could determine y.
7. John says: Oh, then I do know the numbers x and y.

Determine x and y!

Denniston's algorithm for $\textsc{Freudenthal}^=(2, 5.000)$ yields ten possible solution pairs that agree with the first four statements of the conversation; these ten pairs are listed in Table 4. Since the values $x = 4$, $x = 16$, $x = 32$, and $x = 64$ do not uniquely determine the corresponding y, we conclude (together with John) that the answer must be the (phantom) solution $x = 67$ and $y = 82$.

	1	2	3	4	5	6	7	8	9	10
x	4	4	4	16	16	32	32	64	64	67
y	13	61	229	73	111	131	311	73	309	82
s	17	65	233	89	127	163	343	137	373	149
p	52	244	916	1.168	1.776	4.192	10.976	4.672	19.776	5.494

Table 4: Ten intermediate solutions for the Meta-Freudenthal problem.

If the meta-variant is built around problem $\textsc{Freudenthal}^{\neq}(2, 5.000)$ instead of $\textsc{Freudenthal}^=(2, 5.000)$, then $x = 67$ and $y = 82$ remains the unique answer. However, the line of argument changes slightly, since $\textsc{Freudenthal}^{\neq}(2, 5.000)$ only possesses five feasible solutions, which are the first, second, fourth, fifth, and tenth solution in Table 4.

7 A Mediterranean variant of Freudenthal

The *Mediterranean Mathematical Olympiad* (MedMO) is an annual mathematical competition for high-school students from all countries which either have a Mediterranean coast or are adjacent to a country with a Mediterranean coast. Here is a slightly adapted version of the first problem posed at MedMO'2005:

> The teacher says to Peter and Sam: I have secretly chosen two positive integers x and y with $x \leq y$. I have told their sum $s = x + y$ to Sam and their product $p = xy$ to Peter.
>
> 1. Sam says: You are not able to determine s.
> 2. Peter says: Aha. But now I know that $s = 136$.
>
> Determine x and y!

Note that Sam's statement #1 summarizes and contracts the first and second statement in the Freudenthal problem $\text{Freudenthal}^{=}(1, *)$: Peter is not able to work out the numbers x and y from the product p, and Sam is aware of this fact. We will demonstrate below that at the end of the conversation, Peter and Sam both know the numbers x and y. Hence, the above conversation is equivalent to the standard Freudenthal problem, except that Peter explicitly reveals the sum $s = 136$ to the reader.

If we replace the value 136 in the conversation by an arbitrary positive integer z, then we arrive at the Mediterranean Freudenthal problem $\text{Med}(z)$. In this section, we will fully analyze and understand all these Mediterranean problems. Some standard definitions: A divisor d of a positive integer z is *proper*, if $1 < d < z$. An integer $z \geq 2$ is *composite*, if it has some proper divisor. Our analysis is structured into three observations.

First: Consider the moment just before statement #1. If the product p is prime, then Peter would already know at that moment that $x = 1$ and that $y = p$. If the product p is composite, then Peter cannot distinguish between the case where $x = 1$ and the case where x is the smallest proper divisor of p. This yields that p must be composite.

Second: We conclude that statement #1 is equivalent to the following: For all positive integers x and y with $x + y = s$, the product xy is composite. And it is not hard to see that this statement simply boils down to: The number $s - 1$ is composite.

Third: Let $1 = d_1 < d_2 < \cdots < d_k$ be an enumeration of the divisors of p that are less or equal to $\sqrt{p}$. Then at the time point just before statement #2, the values $s_i := d_i + p/d_i$ $(i = 1, \ldots, k)$ are Peter's current candidate values for the sum s. The Mediterranean problem has a solution, if and only if Peter can exclude all these candidates except one. And Peter can exclude the candidate s_i, if and only if $s_i - 1$ is prime. Consequently, with a single exception all the values $s_i - 1$ must be prime. And we already have identified this single exception: Since $d_1 = 1$, the value $s_1 - 1 = d_1 + p/d_1 - 1$ equals p, and we observed above that p is composite. Hence, $p = s - 1$, $x = 1$, and $y = p$.

We summarize the above observations in the following theorem.

Theorem 4. *Let $z \geq 2$ be an integer. The Mediterranean problem* MED(z) *is well-posed and possesses a unique solution, if and only if z is a so-called Mediterranean number, that is, a number that satisfies the following properties:*

- $z - 1$ *is composite*
- $d + (z - 1)/d - 1$ *is prime, for any proper divisor d of $z - 1$*

Furthermore, the unique solution in this case is $x = 1$ and $y = z - 1$, and the corresponding product is $p = z - 1$.

Let us quickly verify this theorem for problem MED(136), the well-posed problem from MedMO'2005: Clearly 135 is composite. The factorizations of 135 into two proper factors are $3 \cdot 45$, $5 \cdot 27$, and $9 \cdot 15$. And indeed, the three corresponding candidate sums $3 + 45 - 1 = 47$, $5 + 27 - 1 = 31$, and $9 + 15 - 1 = 23$ all are prime. Therefore 135 is a Mediterranean number, and the unique answer for MED(136) is $x = 1$ and $y = 135$.

The reader may want to check that 5, 9, 10, 16, 28, 33, 34, 36, 46, and 50 are the first ten Mediterranean numbers. Also 666 (the number of the beast) is a Mediterranean number. Altogether, 39.821 of the integers below 1.000.000 are Mediterranean numbers. We leave the following challenge to the reader: Is there a polynomial time algorithm for deciding whether a given number z is Mediterranean?

8 More stable solutions and phantom solutions

This section continues the discussion in Section 5. We investigate the solution sets for FREUDENTHAL$^{\neq}(m, M)$ and FREUDENTHAL$^{=}(m, M)$ as M grows while m is fixed. Theorem 1 easily generalizes to $m \geq 1$, and thus yields the classification into stable solutions and phantom solutions for any fixed $m \geq 1$.

Let us start with the case $m = 1$. The stable solutions for problem FREUDENTHAL$^{=}(1, *)$ are easy to describe, since they are closely related to Theorem 4: A pair forms a stable solution, if and only if it is of the form $(1, z-1)$ where z is a Mediterranean number. The stable solutions for problem FREUDENTHAL$^{\neq}(1, *)$ can be characterized in a similar fashion: A pair is a stable solution, if and only if it is of the form $(1, z-1)$ where z satisfies the following two almost-Mediterranean properties:

- $z - 1$ is neither prime, nor the square of a prime
- $d + (z - 1)/d - 1$ is prime or the square of a prime, for any proper divisor d of $z - 1$ with $d^2 \neq z - 1$

		Freudenthal$^=(2, M)$		Freudenthal$^{\neq}(2, M)$	
x	y	$x + y \in \mathcal{S}_2$	solution	$x + y \in \mathcal{S}_2$	solution
4	13	$28 \leq M$	$65 \leq M$	$28 \leq M$	$65 \leq M$
4	61	$124 \leq M$	$869 \leq M$	$173 \leq M$	$1.685 \leq M$
32	131	$317 \leq M$	$1.505 \leq M$	$317 \leq M$	$9.413 \leq M$
16	73	$169 \leq M$	$1.970 \leq M$	$169 \leq M$	$1.970 \leq M$
16	111	$233 \leq M$	$2.522 \leq M$	$233 \leq M$	$2.522 \leq M$
32	311	$677 \leq M$	$3.832 \leq M$	$677 \leq M$	$6.245 \leq M$
64	73	$265 \leq M$	$4.037 \leq M$	$265 \leq M$	$6.245 \leq M$
4	229	$460 \leq M$	$4.628 \leq M$	$460 \leq M$	$6.893 \leq M$
8	239	$485 \leq M$	$7.787 \leq M$	$485 \leq M$	$72.365 \leq M$
4	181	$364 \leq M$	$7.898 \leq M$	$1.373 \leq M$	$237.173 \leq M$
16	163	$349 \leq M$	$7.940 \leq M$	$349 \leq M$	$7.940 \leq M$
64	127	$367 \leq M$	$9.104 \leq M$	$367 \leq M$	$9.104 \leq M$

		Freudenthal$^=(3, M)$		Freudenthal$^{\neq}(3, M)$	
x	y	$x + y \in \mathcal{S}_2$	solution	$x + y \in \mathcal{S}_2$	solution
13	16	$49 \leq M$	$98 \leq M$	$49 \leq M$	$125 \leq M$
16	73	$169 \leq M$	$961 \leq M$	$169 \leq M$	$9.413 \leq M$
64	127	$367 \leq M$	$1.783 \leq M$	$367 \leq M$	$5.045 \leq M$
16	133	$283 \leq M$	$2.767 \leq M$	$283 \leq M$	$6.893 \leq M$
16	163	$349 \leq M$	$5.300 \leq M$	$349 \leq M$	$5.300 \leq M$
16	223	$469 \leq M$	$5.761 \leq M$	$469 \leq M$	$332.933 \leq M$
64	367	$847 \leq M$	$5.821 \leq M$	$847 \leq M$	$18.773 \leq M$
16	193	$403 \leq M$	$7.229 \leq M$	$403 \leq M$	$7.229 \leq M$
64	457	$1.024 \leq M$	$9.349 \leq M$	$1.024 \leq M$	$36.485 \leq M$

Table 5: Some stable solutions for $m = 2$ and $m = 3$.

Since the arguments are similar to those in Section 7, we leave all details to the reader. The smallest stable solution for Freudenthal$^=(1, *)$ is $(1, 4)$, which is active for all $M \geq 11$. The smallest stable solution for Freudenthal$^{\neq}(1, *)$ is $(1, 6)$, which is active for all $M \geq 23$. There are plenty of phantom solutions for Freudenthal$^=(1, *)$ and Freudenthal$^{\neq}(1, *)$, and they do not seem to have interesting properties. We only mention that the phantom solution $(3, 4)$ for Freudenthal$^=(1, *)$ is particularly simple and can be verified by hand; it is active for the range $16 \leq M \leq 22$.

Now let us turn to $m = 2$ and $m = 3$. The left half of Table 5 lists all stable so-

		Freudenthal$^=(2, M)$	Freudenthal$^{\neq}(2, M)$
x	y	active in the interval	active in the interval
64	309	$4.625 \leq M \leq 13.168$	$187.493 \leq M \leq 1.739.764$
67	82	$4.721 \leq M \leq 5.485$	$4.721 \leq M \leq 5.485$
139	192	$10.975 \leq M \leq 17.788$	——
149	188	$12.353 \leq M \leq 14.004$	——
83	248	——	$17.789 \leq M \leq 19.324$
96	241	——	$16.133 \leq M \leq 22.804$

Table 6: Some phantom solutions for $m = 2$.

lutions for Freudenthal$^=(2, *)$ and Freudenthal$^=(3, *)$ that enter the scene before $M = 10.000$. The right half of the table lists the corresponding data for problems Freudenthal$^{\neq}(2, *)$ and Freudenthal$^{\neq}(3, *)$. Note that the stable solutions in both halves of the table are the same. This is not just a lucky coincidence:

Theorem 5. *Assume that the following modification of Goldbach's conjecture holds true: Every even number $s \geq 8$ can be written as the sum of two distinct primes. Then for $m = 2$ and $m = 3$, the stable solutions of* Freudenthal$^=(m, *)$ *coincide with the stable solutions of* Freudenthal$^{\neq}(m, *)$.

Proof. Since we deal with stable solutions, the upper bounds M do not play any role and will be ignored throughout. We observe that $\mathcal{S}_2$ only contains odd numbers: The sums $s = 4$ and $s = 6$ obviously cannot be in $\mathcal{S}_2$. Modified Goldbach yields that every even sum $s \geq 8$ is compatible with the product of two distinct primes, and hence not in $\mathcal{S}_2$.

Now consider a product of the form q^4, where q is prime. This product has at most one factorization $q^4 = xy$ with $(x, y) \in Z^{\neq}(m, *)$, but may have two distinct factorizations with $(x, y) \in Z^=(m, *)$. The main difference between the two variants (without upper bound M) is that these products q^4 will show up in the set $\mathcal{P}_1$ for one variant, but not in $\mathcal{P}_1$ for the other variant. However this will not affect the sets $\mathcal{S}_2$, since $\mathcal{S}_2$ only contains odd numbers, whereas the factorizations of q^4 concern the even numbers $q + q^3$ and $2q^2$. ■

We have checked all pairs (x, y) with $x + y \leq 50.000$ with the help of a computer program. Among these pairs there are 1.796 stable solutions and 689 phantom solutions for Freudenthal$^=(2, *)$ and Freudenthal$^{\neq}(2, *)$, and there are 804 stable solutions and 288 phantom solutions for Freudenthal$^=(3, *)$ and Freudenthal$^{\neq}(3, *)$. Some of the phantom solutions for $m = 2$ are listed in Table 6. The smallest phantom solution for $m = 3$ is $(123, 128)$; it is active in the interval $[2.870, 10.480]$ for Freudenthal$^=(3, *)$ and active in the interval $[6.893, 10.480]$ for Freudenthal$^{\neq}(3, *)$.

The behavior of the cases with $m \geq 4$ is not understood. We are not aware of *any* solution for *any* of these problems. In particular, we have not found any solutions (x, y) with $x + y \leq 50.000$.

Conjecture 6. *For $m \geq 4$, problems* FREUDENTHAL$^{=}(m, *)$ *and* FREUDENTHAL$^{\neq}(m, *)$ *do not have any solutions.*

9 Yet another Freudenthal problem

All the Freudenthal problems discussed in this article contained a statement of the type *"I already knew that you didn't know"*, which in some sense is their common theme. Here is a final puzzle of this type:

> The teacher says to Peter and Sam: I have secretly chosen two integers x and y with $1 \leq x \leq y$. I have told their sum $s = x + y$ only to Sam and their product $p = xy$ only to Peter.
>
> 1. Peter says: I don't know the numbers x and y.
> 2. Sam says: I already knew that. The sum is less than 14.
> 3. Peter says: I already knew that. But now I know the numbers x and y.
> 4. Sam says: Oh, then I also know them.
>
> Determine x and y!

It is not difficult to show that $x = 2$ and $y = 9$, and we leave this to the reader.

References

[1] E.W. DIJKSTRA (1978). A problem solved in my head. *Manuscript EWD666.*

[2] H. FREUDENTHAL (1969). Problem No. 223. *Nieuw Archief voor Wiskunde 17,* 152.

[3] H. FREUDENTHAL (1970). Solution to Problem No. 223. *Nieuw Archief voor Wiskunde 18,* 102–106.

[4] M. GARDNER (1979). Mathematical Games. *Scientific American 241(6),* 20–24.

[5] J.O. KILTINEN AND P.B. YOUNG (1985). Goldbach, Lemoine, and a know/don't know problem. *Mathematics Magazine 58,* 195–203.

[6] L. SALLOWS (1995). The impossible problem. *The Mathematical Intelligencer 17,* 27–33.

[7] D.J. SPROWS (1976). Problem No. 977. *Mathematics Magazine 49,* 96.

SUBPROBLEMS AND NP-COMPLETENESS THEORY[*]

Li Sek Su[†]

Abstract

Subproblems have become an important object of NP-completeness theory since its beginning. In this paper, we show some undesirable consequences for subproblems deduced by the standard foundation of the theory, which are different from the practical viewpoint of computation. By the consequences, we clarify further the range in which the standard foundation is applicable to decision problems.

Keywords: NP-completeness theory, complexity, subproblems, decision problems

1 Introduction

NP- completeness theory has been widely used to prove the computational complexity of various problems in mathematics, computer science, cryptography, etc.

Subproblems are obtained from the original problems by giving some restrictions to the allowed instances. The 3-CNF satisfiability problem, the planar graph 3 colorability problem and so on are well-known subproblems. Sometimes the analysis of computational characteristics of subproblems are of very importance in the practical application as well as the theoretical viewpoint. From the fact that subproblems are also a kind of problems and their important role in both theoretical and practical aspects, they have become an object of NP-completeness theory since its beginning [3, 13]. Now we have an amount of subproblems whose complexity has been described by the terms of NP-completeness theory [8, 2, 5, 12].

However, we do not feel free when we apply the standard NP-completeness theory to subproblems. In [9, 10], it has been mentioned that for a subproblem with the set of instances not recognizable in polynomial time, its complexity is not necessarily preserved by transforming into a language recognition according to the standard foundation. As such an example, the problem of deciding whether

[*] This work was done in part during the author's stay in Department für Informartik, Universität Oldenburg by the fellowship from the Alexander von Humboldt Foundation.

[†] Department of Computer Science, University of Science, Pyongyang, DPR Korea

a Hamiltonian graph is 3-colorable has been considered. In [14], it has been considered that we can not apply the standard foundation to such cases as the problem of deciding whether a 3-colorable graph is planar.

In this paper, we present some undesirable consequences, including the results in [14], for subproblems different from the viewpoint of practical computation but deduced from the definitions of standard NP-completeness theory. By the consequences, we clarify further the range in which the standard foundation is applicable to decision problems.

2 Standard foundation of NP-completeness theory

In this section, we describe the fundamental concepts of the standard theory of NP-completeness, referring mainly to [8, 11].

A decision problem Π is a general question to require either "yes" or "no" as its answer. Π consists of 2 kinds of sets D_Π and Y_Π of instances, i.e. $\Pi = (D_\Pi, Y_\Pi)$, where D_Π is the set of all instances and Y_Π the subset of D_Π consisting of all instances with "yes" answer. $L \subseteq \Sigma^*$ is said to be a language over a finite alphabet Σ, where Σ^* is the set of all strings consisting of symbols in Σ with finite lengths.

The correspondence between a decision problem and a language is made by an encoding scheme $e : D_\Pi \to \Sigma^*$. A problem Π and its encoding scheme e separate Σ^* into 3 classes of strings: the class of strings which represent the instances with "yes" answer, the class of strings with "no" answer, and the class of strings which do not represent even any instance. The problem Π is formalized by the first class of strings under the encoding scheme e, and this class is denoted by $L[\Pi, e]$.

Definition 2.1. P is the class of all the languages recognizable by deterministic Turing machines in polynomial time. A decision problem Π is said to be in the class P if the language $L[\Pi, e]$ is in P under a reasonable encoding scheme e.

Definition 2.2. NP is the class of all the languages recognizable by nondeterministic Turing machines in polynomial time. A decision problem Π is said to be in the class NP if the language $L[\Pi, e]$ is in NP under a reasonable encoding scheme e.

Definition 2.3. A language $L_1 \subseteq \Sigma_1^*$ is polynomially reducible to a language $L_2 \subseteq \Sigma_2^*$, denoted by $L_1 \propto L_2$, if there is a mapping $f : \Sigma_1^* \to \Sigma_2^*$ as follows:

1. f is computable by a deterministic Turing machine in polynomial time,

2. for all $x \in \Sigma_1^*$, $x \in L_1$ if and only if $f(x) \in L_2$. Let Π_1, Π_2 be decision problems and e_1, e_2 their reasonable encoding schemes, respectively. If $L[\Pi_1, e_1] \propto L[\Pi_2, e_2]$, then we say that Π_1 is polynomially reducible to Π_2 and denote as $\Pi_1 \propto \Pi_2$.

In the problem level, a polynomial reduction of a decision problem Π_1 to a decision problem Π_2 means the existence of a mapping $f : D_{\Pi_1} \rightarrow D_{\Pi_2}$ as follows:

1. f is computable by a polynomial algorithm,
2. for all $I \in D_{\Pi_1}$, $I \in Y_{\Pi_1}$ if and only if $f(I) \in Y_{\Pi_2}$.

Definition 2.4. A language L is said to be NP-hard if $L' \propto L$ for each $L' \in$ NP.

Definition 2.5. A language L is said to be NP-complete if L ∈NP and L is NP-hard. A decision problem Π is said to be NP-complete(or hard) if the language $L[\Pi, e]$ is NP-complete(or hard) under a reasonable encoding scheme e.

Similarly to the above way, the other classes such as PSPACE, EXPTIME, EXPSPACE etc. and their hardness, completeness are defined.

3 Subproblems and standard foundation

As seen in the above section, the standard foundation of NP-completeness theory was established on the basis of the transformability of (solving) a decision problem into (recognizing) a language and the convenience for the development of complexity theory by the well-formed language theoretical approach [1, 8, 11].

A subproblem of a problem $\Pi = (D_\Pi, Y_\Pi)$ is a problem $\Pi' = (D'_\Pi, Y'_\Pi)$ such that $D'_\Pi \subseteq D_\Pi$ and $Y'_\Pi = Y_\Pi \cap D'_\Pi$ [8]. Because subproblems are a kind of problems, it is certain that they have become an object of the application of NP-completeness theory [3, 13, 8, 2, 5]. Whenever we face a subproblem likely to be important in the practical or theoretical view, we tend to analyze its computational complexity in terms of the NP-completeness theory if it is decidable. However, as we mentioned in the section of Introduction, the standard foundation of NP-completeness theory can cause some undesirability with subproblems.

In this section, we present such undesirability caused with subproblems, which is hardly accepted in the viewpoint of practical computation. The first to mention is that the different subproblems in the problem level can be appeared in the language level as if they are the same problem.

For example, consider the 3-colorability problem for planar graphs (in other words, planar graph 3-colorability) and the planarity problem for 3-colorable graphs. To transform these 2 subproblems into languages, we use the following reasonable encoding scheme over $\Sigma = \{0, 1, *\}$ for graphs which was given in Cook [3]: a graph G is represented by the string consisting of the successive rows of its adjacency matrix, separated by $*$s. Then since the sets of yes instances of the 2 problems under consideration are the same as the set of 3-colorable planar graphs, the 2 languages corresponding to them also become the same language over Σ.

Similarly, consider the deadlock(ability) problem for 1-safe free choice Petri nets and the 1-safeness problem for deadlockable free choice Petri nets [2, 15]. For these 2 subproblems, we can use the following reasonable encoding scheme for marked free choice Petri nets over an alphabet $\Sigma = \{0, 1, *\}$ which is analogous to the one of [3, 5]: a marked free choice net (C, μ^0) is represented by 2 substrings, concatenated by double $*$s, for C and μ^0, respectively. The substring for C consists of the successive rows of its incidence matrix, separated by $*$s. The one for μ^0 consists of the successive binary numbers of its elements, separated by $*$s. Then since the sets of yes instances of the 2 subproblems are both the set of 1- safe deadlockable marked free choice nets, the 2 languages over Σ corresponding to the problems are the same.

In general, if the sets of properties involved in describing different (sub)problems are the same, then the problems can be transformed into a same language under a reasonable encoding scheme irrespective of whether the properties are in Givens or Questions. In such cases, the different problems are regarded in the formal level as if they were the same problem, since their formal models are languages. However, this is never accepted by computer scientists.

The second to say with the standard foundation is that the reduction in the language level does not become just the one in the problem level. For example, consider the reduction of $\Pi = (D_\Pi, Y_\Pi)$ to $\Pi_1 = (D_{\Pi_1}, Y_{\Pi_1})$ or $\Pi_2 = (D_{\Pi_2}, Y_{\Pi_2})$ such that $Y_{\Pi_1} \subseteq Y_{\Pi_2}$ and $(D_{\Pi_1} \setminus Y_{\Pi_1}) \cap (D_{\Pi_2} \setminus Y_{\Pi_2}) = \emptyset$. Let e be the reasonable encoding scheme for Π. Let e_{12} be the reasonable encoding scheme over an alphabet Σ for Π_1 and Π_2. Then the reduction of $L[\Pi, e]$ to $L[\Pi_1, e_{12}] \subseteq \Sigma^*$ also becomes the one of $L[\Pi, e]$ to $L[\Pi_2, e_{12}] \subseteq \Sigma^*$, and hence Π is reducible to Π_2 according to the standard foundation. But such a reduction can never become a reduction of Π to Π_2 in the problem level, which can be called a *pseudo-reduction*.

The third to say is that there really exist such subproblems of Π in P (or NP) which are not in P (or NP). For example, consider the planarity problem for 3-colorable graphs. Since the planarity problem for general graphs is in P[8], it is certain that the one for 3-colorable graphs is solvable in polynomial time. But according to the standard definition of NP-completeness theory, we can deduce the following proposition.

Proposition 3.1. *The planarity problem for 3-colorable graphs is NP-complete.*

Proof. As we considered in the above, under a reasonable encoding scheme the language corresponding to this problem is the same as the language to the 3-colorability problem for planar graphs. Let L be the language. Since the 3-colorability problem for planar graphs is NP-complete by Theorem 4.2 in [8], L is NP-complete. Therefore the problem under consideration to which L is corresponding is NP-complete. Q.E.D.

With the conjecture P⊂NP the above problem can not be said to be in P despite the problem is solvable in polynomial time. As a similar example, consider the halting problem for free (program) schemas. For any free schema it is not a halting one if and only if there is a LOOP statement or a cycle in it [16]. Since the existence problem of a cycle is equivalent to that of a strongly connected component in directed graphs and the latter is solvable in polynomial time [1], the halting problem for free schemas and its complement are both solvable in polynomial time. But from the standard foundation of NP-completeness theory the following proposition is obtained.

Proposition 3.2. *At least one of the halting problem for free schemas and its complement is not solvable in finite time as well as in polynomial time.*

Proof. Let Π be the halting problem for free schemas and Π^c its complement problem. Let e be a reasonable encoding scheme for program schemas. Since the freeness problem for schemas is not even partially decidable (see property 3 at page 270 in [16]), then $e(D_\Pi)$ is not recursively enumerable. Then, since $e(D_\Pi) = e(Y_\Pi) \cup e(Y_{\Pi^c})$, at least one of $e(Y_\Pi)$ and $e(Y_{\Pi^c})$ is not recursively enumerable. On the other hand, since $e(Y_\Pi) = L[\Pi, e]$ and $e(Y_{\Pi^c}) = L[\Pi^c, e]$, at least one of $L[\Pi, e]$ and $L[\Pi^c, e]$ is not recursively enumerable, i.e. for at least one of these there exists no deterministic Turing machine recognizable it. Therefore at least one of Π and Π^c is not solvable in finite time as well as in polynomial time. Q.E.D.

As another example, consider the deadlock problem for 1-safe free choice Petri nets. Since the deadlock problem for free choice Petri nets is NP-complete (see Theorem 14 in [2]), it should be apparent that each of its subproblems including the one under consideration might be either NP-complete or polynomial solvable (see page 80 in [8]). But we can deduce the following proposition according to the standard foundation of NP-completeness theory.

Proposition 3.3. *The deadlock problem for 1-safe free choice Petri nets is PSPACE-hard.*

Proof. As we considered in the first part of this section, under a reasonable encoding scheme the language corresponding to this problem is the same as the one to the 1-safeness problem for deadlock(able) free choice Petri nets. Let L be the language. Since the 1-safeness problem for deadlock(able) free choice Petri nets is PSPACE-hard by Theorem 14 in [15], L is PSPACE-hard. Therefore the problem under consideration to which L is corresponding is PSPACE-hard. Q.E.D.

As long as NP ⊆ PSPACE, the above problem can not be said to be in NP despite the problem is solvable in nondeterministically polynomial time. As considered in the above, some undesirable consequences which are hard to be accepted in the viewpoint of practical computation can be obtained from the standard foundation of NP-completeness theory with subproblems. Especially, from the propositions above we can be aware of that if the complexity of recognizing the instances of a problem is so stronger than the complexity of solving the problem itself that we can not preserve the complexity class of the problem then we may come not to apply the standard foundation to the problem. However, we can not say that the above statement holds for all such cases. For example, consider the liveness problem for bounded free choice Petri nets. This problem can be solvable in polynomial time [6], while the recognition of its instances can not be said to be in polynomial time since it is PSPACE-hard by Corollary 17.2 in [15]. And its language under a reasonable encoding scheme is in P. The reason is as follows. Since the liveness and boundedness problem for free choice Petri nets is in P by Corollary 6.18 in [4] and the set of yes instances of this problem is the same as the one of the problem under consideration, their languages are the same; therefore our problem is also in P as we considered in this section.

4 Conclusion

In this paper, we presented some undesirable consequences deduced with subproblems from the standard foundation of NP-completeness theory, which are hardly accepted in the viewpoint of practical computation. Based on these consequences, we considered about that in which cases we can apply the standard foundation of NP-completeness theory and in which cases we can not.

In the cases that we could not apply the standard foundation, it would be indispensable for us to use the promise problem extension of the standard theory, which has been originated by Even-Selman-Yacobi [7] and systemized by Goldreich [9, 10]. In my opinion, promise problems are a kind of yes/no problems [16], which can be defined in the same way as decision problems described in this paper. If we extend slightly the concept of languages L over Σ into the concept of domain languages (L, D_Σ) over Σ and transform a problem $\Pi = (D_\Pi, Y_\Pi)$ into a domain language $(e(Y_\Pi), e(D_\Pi))$ under a reasonable encoding scheme e similarly to [14], then we can get the same variant as the promise problem extension.

Acknowledgements. I would like to thank Eike Best for his several advices that enabled to consider about the contents of the paper in various aspects. I am grateful to Harro Wimmel for his discussions and opinions. I also wish to thank Javier Esparza and Jacobo Toran for answering my questions, pointing out the relation with promise problems, and making additional helpful suggestions.

References

[1] A. Aho J. Hopcroft, D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, 1974.

[2] A. Cheng, J. Esparza, J.Palsberg, Complexity results for 1-safe nets, Theoret. Comput. Sci.147 (1995) 117-136.

[3] S. Cook, The Complexity of Theorem-Proving Procedures, in: Proc. 3rd Ann. ACM Symp. on Theory of Computing, ACM, New York, 1971, pp.151-158.

[4] J. Desel, J. Esparza, Free Choice Petri Nets, Cambridge University Press, 1995.

[5] J. Esparza, Decidability and Complexity of Petri net Problems-An Introduction, Lectures on Petri Nets I: Basic Models, Advances in Petri Nets, Lecture Notes in Computer Science, Vol.1491 (Springer Verlag, 1998) 374-428.

[6] J. Esparza, M. Silva, A Polynomial-time algorithm to decide Liveness of Bounded Free Choice nets, Theoret. Comput. Sci.102 (1992) 185-205.

[7] S. Even, A. Selman, Y. Yacobi, The Complexity of Promise Problems with Applications to Public-Key Cryptography, Inform. and Contr. 61 (1984) 159-173.

[8] M. Garey, D. Johnson, Computers and Intractability, A Guide to the Theory of NP-Completeness, Bell Laboratories, 1979.

[9] O. Goldreich, On Promise Problems: A Survey, in: Theoretical Computer Science:Essays in Memory of Shimon Even, Lecture Notes in Computer Science, Vol.3895 (Springer Verlag, 2006) 254-290.

[10] O. Goldreich, Computational Complexity: Conceptual Perspective, Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel, Working draft, 2006.

[11] J. Hopcroft, D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, 1979.

[12] N. Jones, L. Landweber, Y. Lien, Complexity of some problems in Petri Nets, Theoret. Comput. Sci. 4 (1977) 277-299.

[13] R. Karp, Reducibility among Combinatorial Problems, in: Complexity of Computer Computations Proc. Symp., 85-103, 1972.

[14] Li Sek Su, New Foundation of NP-Completeness Theory, in: Proc of Collaboration Reseach, USTC Press, P.R. of China, 1994, pp.156-161.

[15] Li Sek Su, Full-Output Siphons and Deadlock-Freeness for Free Choice Petri Nets, Technical Report 2/06, University of Oldenburg, 2006.

[16] Z. Manna, Mathematical Theory of Computation, McGraw-Hill, 1974.

Joseph Goguen

Kokichi Futatsugi
Japan Advanced Institute of Sciences and Technology,
Ishikawa, Japan

Jean-Pierre Jouannaud
Université Paris Sud, and
École Polytechnique, France

José Meseguer
University of Illinois at Urbana-Champaign,
USA

This contribution is unusual in content. It is about the life and death of Joseph Goguen, a colleague who devoted his life to science. Joseph Goguen was born on June 28, 1941, and died on the early hours of July 3, 2006, just a few days after the Festschrift Symposium that we organized in his honor around his 65th birthday at the University of California at San Diego.

Joseph had been diagnosed with cancer last February. Prior to that, the three of us had been organizing the Festschrift volume and Symposium in his honor. Joseph wanted us to go ahead with this symposium in spite of his failing health.

The amazing and wonderful thing is that Joseph was able to be present at all the main events of the symposium : at the opening and first session; at a piano recital by his wife Ryoko; at a banquet on his 65th birthday; and at the closing. And he said some words at each of these events. One theme that he insisted on several times was his encouragement for all of us to collaborate and help each other.

Bulletin of the EATCS no 90, pp. 199–201, October 2006

The symposium itself was a wonderful event, both scientifically, and because of the great warmth that everybody showed to Joseph. And we know that it meant a lot to him; so much so that he gathered super-human strength to be present in it, even though his life was quickly ebbing away. And his humor, while being fully aware that he was dying, had a special spark and joyfulness to it; a joyfulness and serenity that he communicated to all of us who had the fortune of seeing him in those last days. Something worth mentioning, because it is a wonderful example of true friendship, is that Rod Burstall, emeritus professor at the University of Edinburgh and a long-time close friend and colleague of Joseph's, moved to San Diego for the last two months of Joseph's life to spend time with him and to help out in all kinds of ways.

The great impact and influence that Joseph's scientific ideas have had on computer science, and in particular on semantics and formal methods is briefly explained in the preface we wrote for the Festschrift [1], which is partly reproduced below. The Festschrift [1] also contains a full bibliography of Joseph Goguen's published work. One amazing thing is the breadth of it all; in particular, his systematic and sustained effort to connect the humanities, specially in areas relevant to computing and its social impact, with mathematical models and with computer science. We think that the papers in the volume speak for themselves and provide a wonderful overview of Joseph Goguen's enormously influential ideas in one of the best ways possible, namely, by reflecting on how they have become and are part of a vast scientific dialogue. In the preface of [1] we wrote:

> Joseph Goguen is one of the most prominent computer scientists worldwide. His numerous research contributions span many topics and have changed the way we think about many concepts. Our views about data types, programming languages, software specification and verification, computational behavior, logics in computer science, semiotics, interface design, multimedia, and consciousness, to mention just some of the areas, have all been enriched in fundamental ways by his ideas.
>
> Considering just one strand of his work, namely the area of Algebraic Specifications, his ideas have been enormously influential. The concept of initiality (or co-initiality) that he introduced is now a fundamental concept in theoretical computer science applied in many subfields. The Clear formal specification language was the first language with general theory composition operations based on categorical algebra. Such generality inspired Goguen and Burstall to propose institutions as a meta-logical theory of logics, so that Clear-like languages could be defined for many logics. The OBJ language, one of the earliest and most influential executable algebraic specification languages, also incorporated the Clear ideas. Categorically-based module composition operations had an enormous influence not only

> in formal specification, but also in software methodology: his parameterized programming methodology predates by about two decades more recent work on generic programming. These ideas, and many others that he has pioneered, reverberate through the pages of the Festschrift, in which entire chapters are devoted to some of them. Furthermore, there are several regular scientific meetings of an international scope, including the CALCO and AMAST conferences and the WADT Workshop, dedicated to ideas either initiated or directly influenced by Joseph Goguen. There are also a number of important languages that have been influenced by his CLEAR and OBJ algebraic specification languages, including: ACT1, ML, CASL, Maude, CafeOBJ, and ELAN.
>
> A common thread in his work is the use of abstract algebra, particularly of categorical algebra, to get at the core of each problem and formulate concepts in the most general and useful way possible. Algebraic and logical methods are then deployed to provide a rigorous account of meaning, both in computational systems and in semiotic systems. Furthermore, in areas in which social aspects are involved, a humanistic perspective is combined with mathematical and computational perspectives to do justice in a non-reductionist and critical way to a wide range of human phenomena, including phenomena arising from the use or misuse of computer systems in concrete social situations.

We feel privileged to have edited the Festschrift, and for having been able to be with him at the symposium shortly before his death. For us it has been a way of expressing our admiration, our gratitude, and our friendship to Joseph Goguen. The four of us worked closely together at SRI's Computer Science Laboratory designing and implementing the OBJ2 language during the 1983–4 academic year. The scientific enthusiasm, camaraderie, and friendship of that relatively short but very influential period have grown over the years and have had a great impact on our lives.

His death is a great loss for many of us, not only as a brilliant and most influential colleague, but above all as a dear friend. And we share in the sadness of his family, and of many other friends who were touched by his life.

References

[1] Algebra, Meaning, and Computation: A Festschrift in Honor of Joseph Goguen. K. Futatsugi, J.-P. Jouannaud and J. Meseguer editors. Lecture Notes in Computer Science 4060. Springer Verlag, 2006.

Zdzisław Pawlak
Life and Work

1926–2006

Professor Zdzisław Pawlak, Member of the Polish Academy of Sciences, will be remembered as a great human being with exceptional humility, wit and kindness as well as an extraordinarily innovative researcher with exceptional stature. His research contributions have had far-reaching implications with many of them playing a fundamental role in establishing new perspectives for scientific research in a wide spectrum of fields.

Zdzisław Pawlak was born on 10 November 1926 in Łódź, 130 km southwest from Warsaw, Poland[1]. In 1947, Pawlak began his studies in the Faculty of Electrical Engineering at Łódź University of Technology, and then from 1949 continued his studies in the Telecommunication Faculty at Warsaw University of Technology. In 1950, he presented the first project of a computer in Poland, called GAM 1. He completed his M.Sc. in Telecommunication Engineering in 1951. His publication in 1956 on a new method for random number generation was the first publication abroad in informatics by a researcher from Poland. In 1958, Pawlak completed his doctoral degree from the Institute of Fundamental Technological Research at the Polish Academy of Science with a Thesis on Applications of Graph Theory to Decoder Synthesis. In 1961, Pawlak was also a member of a

[1] Wikipedia summary of the life and work of Z. Pawlak:
`http://pl.wikipedia.org/wiki/Zdzislaw_Pawlak`

research team that constructed one of the first computers in Poland called UMC 1. Pawlak received his habilitation from the Institute of Mathematics at the Polish Academy of Sciences in 1963. In his habilitation "Organization of Address-Less Machines", he proposed and investigated parenthesis-free languages, a generalization of polish notation introduced by Jan Łukasiewicz.

During succeeding years, Pawlak worked at the Institute of Mathematics at Warsaw University and, in 1965, introduced the foundations for modeling DNA and what has come to be known as molecular computing. In 1968, he proposed a new formal model of a computing machine known as the *Pawlak machine* which was based on the addressing structure of contemporary computers. During the 1970s, Pawlak introduced knowledge representation systems as a result of his broader research on the mathematical foundations of information retrieval. This led to his most widely recognized contribution, namely, his brilliant approach to classifying objects with their attributes (features) and his introduction of approximation spaces, which establish the foundations of granular computing and provide frameworks for perception and knowledge discovery in many areas.

During the early 1980s, he worked at the Institute of Computer Science of the Polish Academy of Sciences, where he introduced rough sets and the idea of classifying objects by means of their attributes[2]. Rough set theory has its roots in Pawlak's research on knowledge representation systems. Rather than attempt exact classification of objects with attributes (features), Pawlak considered an approach to solving the object classification problem in a number of novel ways. First, in 1973, he introduced knowledge representation systems. Then, in 1981, he introduced approximate descriptions of sets of objects and considered knowledge representation systems in the context of upper and lower classification of objects relative to their attribute values. During the succeeding years, Pawlak refined and amplified the foundations of rough sets and their applications and nurtured worldwide research in rough sets that has led to over 4000 publications[3]. The consequences of this approach to the classification of objects relative to their feature values have been quite remarkable and far-reaching. The work on knowledge representation systems and the notion of elementary sets have profound implications when one considers the problem of approximate reasoning and concept approximation. Also, during the 1980s, Pawlak invented a new approach to conflict analysis.

[2] Z. Pawlak, Rough Sets. Research Report PAS 431, Institute of Computer Science, Polish Academy of Sciences (1981); Z. Pawlak, Classification of Objects by Means of Attributes. Research Report PAS 429, Institute of Computer Science, Polish Academy of Sciences, ISSN 138-0648, January (1981); Z. Pawlak, Rough sets. International J. Comp. Inform. Science 11 (1982) 341-356; Z. Pawlak, Rough Sets – Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, Dordrecht, 1991.

[3] See, e.g., Rough Set Database System, `http://rsds.wsiz.rzeszow.pl/`.

Professor Zdzisław Pawlak was with us only for a short time and, yet, when we look back at his accomplishments, we realize how greatly he has influenced us with his generous spirit and creative work in many areas such as approximate reasoning, intelligent systems research, computing models, mathematics (especially, rough set theory), molecular computing, pattern recognition, philosophy, art, and poetry. As many can readily testify, Pawlak gave generously his time and energy to help others. His spirit and insights have influenced many researchers worldwide. During his life, he manifested an extraordinary talent for inspiring his students and colleagues as well as many others outside his immediate circle.

Andrzej Ehrenfeucht, James F. Peters, Grzegorz Rozenberg, Andrzej Skowron

AUTOMATA: FROM MATHEMATICS TO APPLICATIONS AUTOMATHA

AN ESF STANDING COMMITTEE FOR PHYSICAL AND ENGINEERING SCIENCES (PESC) PROGRAMME

MAY 2005–MAY 2010

Abstract

AutoMathA (Automata: from Mathematics to Applications) is an international research programme of the European Science Foundation (ESF). As it lies at the crossroad of mathematics, theoretical computer science and applications, it is expected to catalyse progress in both theoretical and practical directions. Main activities of the programme are to provide a full financial support for visits/exchanges among the programme participants (usually for short periods, typically two weeks), to organize workshops and schools for programme participants and to sponsor conferences in the area of AutoMathA. All applications should be submitted via the online application forms available on AutoMathA webpage (`www.esf.org/automatha`).

Automata theory (AT) is one of the longest established areas in computer science. Over the past few years AT has not only developed in many different directions but has also evolved in an exciting way at several levels: the exploration of specific new models and applications has at the same time stimulated a variety of deep mathematical theories. Standard applications of AT include pattern matching, syntax analysis and software verification. In recent years, novel applications of automata-theoretic concepts have emerged from biology, physics, cognitive sciences, neurosciences, control, tomography, linguistics, mathematics, etc., while developments in information technology have increased the need for formally based design and verification methods to cope with emerging technical needs such as network security, mobile intelligent devices and high performance computing. At the same time, the mathematical foundations of AT rely on more and more advanced areas of mathematics. While in the early 1960s only elementary graph theory and combinatorics were required, new tools from non-commutative algebra (semigroups, semirings and formal power series), logic, probability theory and symbolic dynamics have been successively introduced and the latest developments borrow ideas from topology and geometry. Both trends have enhanced the role of fundamental research in AT and the importance of closer interaction between theoretical and applied scientists. This multidisciplinary programme lies at the crossroads of mathematics, theoretical computer science and applications. By setting up a framework where new applications of AT and theoretical insights can be communicated and shared by an open and qualified group of European scientists, this programme will catalyse progress in both directions.

Activities

The programme includes the following planned activities.

- Short-term visit/exchanges among the programme participants. Eligibility for exchange grants are:
  1. Undertake work applicable to the programme, that is, related to Automata theory or applications.
  2. Apply to stay in a country other than the country of origin.
  3. Return to the institute of origin upon termination, so that the applicant's institution may also benefit from the broadened knowledge of the scientist.
  4. Acknowledge ESF in publications resulting from the grantee's work in relation with the grant.
- Organisation of workshops for programme participants, to allow the dissemination of early research results and experiences.
- Sponsoring of conferences in the area of AutoMathA.
- Organisation of schools on the subjects covered by the programme.
- Organisation of open workshops in the area of Auto-MathA.
- Setting up a comprehensive Internet research dissemination channel and publication activities.

Priority will be given to applications where the institutions involved are in countries that financially support the programme.

Every year, Automatha plans to support 30 short visits, and in 2005-2006, Automatha has already supported the following events: Workshop on Semigroup and Automata, Workshop on Weighted Automata Theory and Application (WATA 2006), Workshop on Advances on Two-dimensional Language Theory, Workshop on Tree Automata, Mons Days of Theoretical Computer Science (JM 2006), Workshop on Algebraic Theory of Automata to Applications, International Conference of Formal Modeling and Analysis of Timed Systems (FORMATS 2006).

For more information please visit `www.esf.org/automatha` or send an email to `automatha@liafa.jussieu.fr`.

Funding

ESF scientific programmes are principally financed by the Foundation's Member Organisations on an à la carte basis. AutoMathA is supported by: Fonds zur Förderung der Wissenschaftlichen Forschung, Austria; Fonds National de la Recherche Scientifique, Belgium; Akademie věd České republiky, Czech Science Foundation, Czech Republic; Forskningsrådet for Natur og Univers, Denmark; Academy of Finland, Finland; Centre National de la Recherche Scientifique, France; Deutsche Forschungsgemeinschaft, Germany; Hungarian Academy of Sciences, Hungarian Scientific Research Fund, Hungary; The Israeli Academy of Sciences and Humanities, Israel; CNR: Ist. di Elettronica e di Ingegneria dell'Informazione e delle Telecomunicazioni, Torino, Ist. di Cibernetica "E. Caianiello", Pozzuoli ; Ist. di Scienza e Tecnologia dell'Informazione "A. Faedo", Pisa, Italy; Polska Akademia Nauk, Poland; Fundação para e Ciência e a Tecnologia, Portugal; Slovak Academy of Sciences, Slovak Republic; The Swiss National Science Foundation for the promotion of scientific research, Switzerland.

Steering committee

- Jean-Eric Pin (Paris, France), chair
- Jorge Almeida (Porto, Portugal)
- Véronique Bruyère (Mons, Belgium)
- Stefano Crespi-Reghizzi (Milano, Italy)
- Jacques Duparc (Lausanne, Switzerland)
- Søren Eilers (Copenhagen, Denmark)
- Zoltan Esik (Szeged, Hungary)
- Jozef Gruska (Brno, Czech Republic)
- Tatiana Jajcayova (Bratislava, Slovak Republic)
- Juhani Karhumaki (Turku, Finland)
- Andrzej Kisielewicz (Wroclaw, Poland)
- Werner Kuich (Wien, Austria)
- Stuart W. Margolis (Ramat Gan, Israel)
- Wolfgang Thomas (Aachen, Germany)

Reports from Conferences

Report on ICALP 2006 / PPDP 2006 / LOPSTR 2006

33rd Intl Colloquium on Algorithms, Languages and Programming
and
8th Symposium on Principles and Practice of Declarative Programming
and
Symposium on Logic-based Program Synthesis and Transformation

9–16 July, Venice, Italy

Manfred Kudlek

ICALP 2006, the 33rd in this series of conferences on theoretical computer science, took place from July 10-14, 2006, together with the workshops from July 9-16, 2006, at Venezia, the fourth time in Italy. It was co-located with PPDP 2006, held from July 10-12, 2006, and LOPSTR 2006 (International Symposium on Logic-based Program Synthesis and Transformation) which took place from July 12-14, 2006. There were also 9 satellite workshops. Conference site was on *San Servolo*, a small island about 2 km southeast from the centre San Marco. Now a conference and meeting place, a former Benedictine monastery and later a hospital run by nuns, monks, and priests.

ICALP 2006 was organized by Dipartimento di Informatica, Università Ca' Foscari di Venezia. The Organizing Committee consisted of Michele Bugliesi (ICALP and chair), Annalisa Bossi (PPDP), Sabina Rossi (LOPSTR), Andrea Pietracaprina and Francesco Ranzato (satellite events), Arianna Caldon and Nora Hoggui (Key Congress), and Paolo Baldan, Martha Cooper, Francesco di Nes, Raffaele Facchin, Marco Franceschin, Marco Guinti, Davide Lappon, Francesco Levorato, Damiano Macedonio, Matteo Maffei, Giuseppe Pirrotta, Alessandra Raffaetà, Giulio Ranzanetto, Marco Scavazzon and Enrico Ramo.

ICALP 2006 was sponsored by Dipartimento di Informatica, Università Ca' Foscari, Venice International University, Microsoft Research, IBM Italia, VENIS S.P.A. (venezia informatica e sistemi), CVR (Consorzio Venezia Ricerche), and EATCS.

The conferences and workshops were attended by at least 432 participants (data from July 11), with some details given in the following table:

ICALP only	245	ICALP+workshops	110
PPDP only	50	ICALP+PPDP+LOPSTR	12
LOPSTR only	30	workshops total	195

ICALP 2006 covered the following fields in the three tracks :

A	**B**
Algorithms	Automata
Approximation Algorithms	Equations
Complexity	Games
Data Structures	Logics
and Linear Algebra	Models
Fixed Parameter Complexity	Semantics
Formal Languages	**C**
Game Theory	Bounded Storage
Graph Algorithms	and Quantum Models
Graph Theory	Cryptographic Primitives
Graphs	Cryptographic Protocols
Networks, Circuits	Foundations
Regular Expressions	Multi-party Protocols
Quantum Computing	Secrecy and Protocol Analysis
Randomness	Zero Knowledge and Signatures

The scientific program of ICALP 2006 consisted of 4 invited lectures, 4 special lectures, and 109 contributions, selected from 407 submissions, exactly the same number as in 2005. They came from 43 countries, the highest number so far. 6 papers were withdrawn, 5 in track B, 1 in C. Details on number of authors and distribution by countries for all tracks are given below. The program was divided into 24 sessions (14 in track A, 3 in B, 7 in C), sometimes 3 in parallel, 4 plenary sessions (1 joint with PPDP, 1 joint with LOPSTR), as well as 2 special sessions. The program can be found at `http://icalp06.dsi.unive.it`.

The following table gives the statistics by number of authors (S submitted, A accepted) in all tracks and total :

	A		B		C		Σ	
	S	A	S	A	S	A	S	A
1	53	13	25	8	10	4	88	25
2	81	18	35	6	38	11	154	35
3	66	19	17	6	18	6	101	31
4	17	4	11	4	8	2	36	10
5	8	4	1		4		13	4
6	4	3	1		1	1	6	4
7			1		1		2	
8	1						1	
	230	61	91	24	80	24	401	109

The next table presents the statistics by countries.

C	I	AS	AA	BS	BA	CS	CA	ΣS	ΣA
AT		$\frac{1}{3}$	$\frac{1}{3}$					$\frac{1}{3}$	$\frac{1}{3}$
AU				1		$3\frac{13}{20}$	$\frac{2}{3}$	$4\frac{13}{20}$	$\frac{2}{3}$
BE		1		$\frac{1}{7}$		$1\frac{1}{3}$	$\frac{1}{6}$	$2\frac{10}{21}$	$\frac{1}{6}$
CA	1	$7\frac{5}{12}$	$1\frac{1}{2}$	$2\frac{1}{4}$		2		$11\frac{2}{3}$	$1\frac{1}{2}$
CH	1	$7\frac{5}{12}$	$\frac{5}{12}$			$1\frac{1}{6}$	$\frac{1}{2}$	$8\frac{7}{12}$	$\frac{11}{12}$
CL		$\frac{2}{3}$	$\frac{2}{3}$					$\frac{2}{3}$	$\frac{2}{3}$
CN		$9\frac{2}{3}$	$\frac{1}{2}$	$10\frac{1}{2}$	1	$3\frac{3}{5}$		$23\frac{23}{30}$	$1\frac{1}{2}$
CO		$\frac{1}{4}$						$\frac{1}{4}$	
CY		$\frac{3}{5}$						$\frac{3}{5}$	
CZ		1	1	$1\frac{1}{12}$	$\frac{1}{3}$			$2\frac{1}{12}$	$1\frac{1}{3}$
DE		$26\frac{1}{6}$	$9\frac{5}{12}$	$9\frac{5}{6}$	$2\frac{1}{4}$	$4\frac{7}{12}$	2	$40\frac{7}{12}$	$13\frac{2}{3}$
DK		$5\frac{5}{12}$	2	$\frac{1}{3}$				$5\frac{3}{4}$	2
DZ		1						1	
EE		2				1		3	
ES		$4\frac{1}{2}$		$2\frac{1}{2}$				7	
FI		3		$1\frac{2}{3}$	$\frac{2}{3}$			$4\frac{2}{3}$	$\frac{2}{3}$
FR		$14\frac{2}{3}$	$\frac{3}{4}$	$13\frac{5}{12}$	$4\frac{1}{2}$	$10\frac{53}{420}$	$5\frac{1}{3}$	$38\frac{22}{105}$	$10\frac{7}{12}$
GR		$5\frac{7}{15}$	$2\frac{3}{5}$			1		$6\frac{7}{15}$	$2\frac{3}{5}$
HK		$1\frac{1}{2}$	$\frac{1}{2}$			$\frac{2}{3}$		$2\frac{1}{6}$	$\frac{1}{2}$
HU		1		2				3	
IE		1	1					1	1
IL	1	$20\frac{2}{3}$	$4\frac{1}{2}$	3		$3\frac{4}{5}$	3	$27\frac{7}{15}$	$7\frac{1}{2}$
IN	1	$9\frac{1}{2}$	2			5		$14\frac{1}{2}$	2
IS		$\frac{1}{4}$		1	1			$1\frac{1}{4}$	1
IT		$11\frac{3}{20}$	$3\frac{11}{15}$	$8\frac{8}{21}$	$2\frac{1}{2}$	3	2	$22\frac{223}{420}$	$8\frac{7}{30}$
JO		1						1	
JP		$6\frac{5}{6}$	$1\frac{1}{3}$			$7\frac{17}{28}$	2	$14\frac{37}{84}$	$3\frac{1}{3}$
KR		2				7		9	
NL		$2\frac{1}{3}$	1	$1\frac{1}{7}$	1	$5\frac{1}{2}$	1	$8\frac{41}{42}$	3
NO		1						1	
PL		$2\frac{3}{4}$	$1\frac{1}{4}$	$4\frac{1}{2}$	3	1		$8\frac{1}{4}$	$4\frac{1}{4}$
PT				1		$\frac{1}{2}$	$\frac{1}{2}$	$1\frac{1}{2}$	$\frac{1}{2}$
RO				1				1	
RU		$2\frac{1}{3}$	1	1				$3\frac{1}{3}$	1
SE		$3\frac{1}{2}$	1					$3\frac{1}{2}$	1
SG		1						1	
SI		$\frac{1}{2}$						$\frac{1}{2}$	

C	I	AS	AA	BS	BA	CS	CA	ΣS	ΣA
SK		$1\frac{1}{2}$	$\frac{1}{2}$					$1\frac{1}{2}$	$\frac{1}{2}$
TN				$\frac{3}{4}$				$\frac{3}{4}$	
TW		1	1					1	1
UA		$\frac{2}{3}$						$\frac{2}{3}$	
UK	2	$6\frac{3}{5}$	$1\frac{3}{4}$	$13\frac{1}{6}$	$3\frac{1}{2}$	$2\frac{5}{6}$	$1\frac{5}{6}$	$22\frac{3}{5}$	$7\frac{1}{12}$
US	1	$61\frac{17}{20}$	$21\frac{1}{4}$	$15\frac{1}{14}$	$4\frac{1}{4}$	13	5	$89\frac{129}{140}$	$30\frac{1}{2}$
Σ	7	230	61	91	24	80	24	401	109

ICALP 2006 was accompanied by the following 9 workshops :

CHR 2006	Third Workshop on Constraint Handling Rules
FCC 2006	Formal and Computational Cryptography
MeCBIC 2006	Membrane Computing and Biologically Inspired Process Calculi
ALGOSENSORS 2006	International Workshop on Algorithmic Aspects of Wireless Sensor Networks
CL&c	Classical Logic and Computation
SecReT 2006	1st International Workshop on Security and Rewriting Techniques
DCM 2006	2nd International Workshop on Developments in Computational Models
iETA	Improving Exponential Time Algorithms
WCAN 2006	2nd Workshop on Cryptography for Ad Hoc Networks

The next table presents the date, number of invited lectures and contributions.

CHR	7.09	1	9	SecReT	7.15	1	8
FCC	7.09		9	DCM	7.16	1	9
MeCBIC	7.09	2	15	iETA	7.16	2	9
ALGOSENSORS	7.15	1	20	WCAN	7.16	1	10
CL&c	7.15	2	7				

ICALP 2006 was opened on Monday morning by Michele Bugliesi, introducing Venezia, San Servolo, and thanking the program committee, Mogens Nielsen, Jan van Leeuwen, the organizers, invited speakers, and sponsors.

In the excellent first invited lecture *'Additive Approximation for Edge-Deletion Problems'* (Giuseppe Italiano had to chair since Ingo Wegener was lost in Venice) Noga Alon (co-authors Asaf Shapira, Benny Sudakov), mentioning football WC the penalty shooting the day before, presented new results on algorithms for monotone graph properties, using results from extremal graph theory and spectral techniques by Erdős, Szemerédi, and Turán.

Cynthia Dwork, in the very good and interesting second one on *'Differential Privacy'*, presented an approach to formalize privacy. She started with Kinsey's *'Let's talk about sex'* and Dalenius' definition of privacy *'Anything to be learned from statistical data bases can be learned without access to data bases'*. Then she presented a survey on a general impossibility result, differential privacy, how to achieve it, and a substantiating impossibility result, finishing with *'No Holy Grail!'*

The third one (joint with PPDP) *'Composable Memory Transactions'*, given by Simon Peyton Jones on his 20th wedding anniversary, was an excellent, very vivid, survey on the history of concurrency (*'How to program these beasts'*), races, deadlock, lost wakeups, diabolical error recovery), significant recent progress (*'bricks and mortar instead of bananas*), and 3 primitives (atomic, retry, orElse). Then he gave an introduction on realising STM in Haskell (*'Haskell programmers are brutally trained from birth to use memory effects sparsingly'*). He concluded with *'It's like using a high-level language instead of assembly code. Not a silver bullet'*. Unfortunately, it is not in the proceedings, but information can be found in *'Atomic Blocks and Transactional Memory'* (co-authors Tim Harris, Simon Marlow, at `http://research.microsoft.com/~simonpj`.

The fourth one (joint with LOPSTR), *'The One Way to Quantum Computation'* by Prakash Panangaden (co-authors Vincent Damos, Elham Kashefi) was an excellent survey on the topic, the new model from physics (quantum mechanics and computation, entaglement and teleportation), measurement based computation and calculus, standardization, the Bell Ineaquality, and also quantum process algebra. He was introduced by Vladimiro Sassone who also mentioned his fantastic jokes.

In the IBM session on Monday late afternoon Birgit Pfitzmann gave an interesting talk on *'Security and Privacy Challenges in Industrial Research'*, starting with *'This is a rather industrial talk'*. She presented a survey on present IBM, todays security challenges, impact of new IT paradigms, problems in complexity reduction, construction of sound foundations, model-driven security design paradigms, risk and compliance, legal problems, and the market. She finished with *'Security remains a critical research topic. How to cope with complexity ? Theory would help in many ways'*.

The award session on Thursday afternoon was opened by Mogens Nielsen. The first event was the presentation of the *Gödel Prize*. It started with the interesting special talk *'Kurt Gödel, Some Dates'* by Pierre-Louis Curien. In it he gave a survey on Gödel' scientific life, the incompleteness theorem, the Princeton lectures, and the system T in *Dialectica*, as well as the development of computability in the 30s of last century. After that he informed us on the Gödel price and its winners. Finally he explained the decision of the committee (he, Volker Diekert, Christos Papadimitriou, John Reif, Jeff Ullman, Paul Vitányi) to present the price to Manindra Agrawal, Neeraj Kayal, Nitin Saxena for their paper *'Primes are in P'*. (in Annals of Mathematics 160(2), pp 781-793, 2004).

After this Manindra Agrawal explained in a very nice talk *'A Short History of 'Primes is in P''* the research history from August 1998 until August 2002, conjectures, failures, experiments, and the final breakthrough to get the result, also mentioning his professor Somenath Biswas.

The next event was the presentation of the EATCS distinguished award to Michael S. Paterson for his scientific work. The decision was explained by Mariangiola Dezani-Ciancaglini. After that Mike Paterson gave a very interesting talk on *'Secrets of my Success'*, starting with *'Thank you !'* and lifting the secret of the initial S., namely Stewart. First : *'Start early'* (1964/7 in his case), meeting right people as Miss D. M. Tibenham (also present at ICALP as his wife), and *'Get lucky'*, e.g on technology in conferences (he gave a talk on ICALP'78 in Udine how to use slides). Second : *'Hang around with smart people'* (as Mike Fischer, Uri Zwick, Leslie Goldberg). Third : *'Enjoy what you do !'*. Some photos illustrated his life. He finished with *'Thank you again !'*.

To mention are also the excellent presentations of the best papers. In track A by Michael Paterson (co-authors Martin Dyer, Leslie Ann Goldberg) on *'On Counting Homomorhisms to Directed Acyclic Graphs'*, in track B by Filip Murlak on *'The Wadge Hierarchy of Deterministic Tree Languages'*, and in track C by 'Danny Harnik (co-authors Iftach Haitner, Omer Reingold) on *'Efficient Pseudorandom Generators from Exponentially Hard One-way Functions'*, as well as of the best student paper by Qiqi Yan on *'Lower Bounds for Complementation of ω-Automata via the Full Automata Technique'*.

Excellent and interesting presentations were given by Magnus Bordewich on stopping time analysis and approximate counting, by Turlough Neary on the P-completeness of *Wolfram*'s cellular automaton 110, by John Hitchcock on comparison of reductions to NP-complete sets, by Juhani Karhumäki on the power of rewriting and communication of 2 stacks, and by Markus Lohrey on decision problems for theories of HNN extensions and amalgamated products.

To mention are also the good and interesting talks by Oleg Verbitsky, referring to a follow-up paper, on testing the graph isomorphism problem, by Amin Coja-Oghlan on a spectral gap of certain random graphs, by Douglas E. Carroll on the embedding of bounded bandwidth graphs, by Ronald de Wolf, starting with *'Not really QC'*, on lower bounds on matrix rigidity, applying quantum arguments, and finishing with *'To be or not to be quantum'*, by Jaikumar Radhakrishnan on gap amplification in PCP's, and by Michal Kunc on algebraic characterization of the finite power property.

Other good and interesting presentations were given by Christos Kapoutsis on non-closure under complement of small sweeping NFA's, by Matthias Samuelides on the power of pebble automata, by Yevgeniy Dodis on the impossibility to extract classical randomness by quantum computers, by Tomoyuki Yamakami, showing Buddha as oracle, on quantum hardcore functions, finishing with *'QC is listening by phone'*, by Sébastien Zimmer onhardness of distinguishing MSB and LSB of secret keys, and by Mihalis Yannakakis on recursive concurrent stochastic games.

Not to forget are the also interesting and good talks by Colin Stirling on a game-theoretic metod to decide higher-order matching, by Esfandiar Haghverdi on typed geometry of interaction for exponentials, by Corin Pitcher on security languages using λ-calculus, by Wong Karianto, advertising STACS 2007, on intersection problems for polynomially generated sets, by Aniello Murano, having the proper name for ICALP 2006, on complexity of enriched μ-calculi, and by Blaise Genest on exponential-size deterministic Zielonka-automata.

Kohei Honda started with *'First, it's useful'*, and Bas Luttik with *'You ended up in an equation session. There will be many more in this talk'*.

The EATCS General Assembly was held on Tuesday late afternoon. On it there is a separate report. Mike Paterson received the best paper award for track A by Ingo Wegener, and Qiqi Yan the best student paper award by Vladimiro Sassone. The winners in tracks B and C received their prices later. Burkhard Monien got an EATCS button by the author of this report for having reached more than 5 full papers on ICALP's. Each of the four editors of the proceedings received a button, too. The current state of busy contributors to ICALP's is given in the table above.

The proceedings, edited by Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, have been published for the first time in two volumes as Springer LNCS 4051 (track A), and 4052 (tracks B, C). They contain all contributions and invited lectures, except for that by Simon Peyton Jones, that of Noga Alon only as abstract. There is also a CD-ROM with the two volumes.

The social program started on Sunday late afternoon, watching the Football World Championship final in the *Auditorio*. On Monday evening the welcome reception took place in the church yard of *San Servolo*, immediately after the IBM session. It lasted well until 21h. On Tuesday evening, fater the EATCS general assembly, there was a reception in the patio. On Wednesday afternoon we had a guided excursion on two boats to other islands in the Lagoon. The first stop was on *Murano* where we visited the glass factory *Ferro-Lazzarini*. Master *Giorgio* demonstrated us the traditional glass blowing, producing a glass horse within a few minutes. After that we could visit the show and shopping room, or have a walk through a part of the island. The next stop was on *Torcello* with a short walk to *Basilica Santa Maria Assunta*, from 639 AD the oldest church in the region, and *Attila*'s chair. The last stop was on *Burano*, famous for lace making, and a lop-sided church tower. There we had one hour time for walking around

Kurt Mehlhorn	11	*ICALP Contributors*	
Jean-Eric Pin	$10\frac{1}{2}$		
Juhani Karhumäki	$9\frac{7}{60}$	Dominique Perrin	$4\frac{5}{6}$
Zvi Galil	8	Zohar Manna	$4\frac{5}{6}$
Amir Pnueli	$7\frac{1}{2}$	Thomas Henzinger	$4\frac{3}{4}$
Mihalis Yannakakis	$7\frac{5}{12}$	Juraj Hromkovič	$4\frac{7}{10}$
Philippe Flajolet	$7\frac{1}{4}$	Denis Thérien	$4\frac{7}{12}$
Grzegorz Rozenberg	7	Moshe Vardi	$4\frac{7}{12}$
Paul Vitányi	$6\frac{11}{12}$	Manfred Droste	$4\frac{1}{2}$
Claus-Peter Schnorr	$6\frac{1}{2}$	Robin Milner	$4\frac{1}{2}$
Torben Hagerup	$6\frac{1}{2}$	David Peleg	$4\frac{9}{20}$
Géraud Sénizergues	$6\frac{1}{2}$	Ming Li	$4\frac{5}{12}$
Christos Papadimitriou	$5\frac{5}{6}$	Maurice Nivat	$4\frac{1}{4}$
Karel Čulik II	6	Moti Yung	$4\frac{1}{4}$
John Reif	$5\frac{3}{4}$	Volker Diekert	$4\frac{1}{6}$
Walter Vogler	$5\frac{1}{2}$	Piotr Berman	$4\frac{1}{6}$
Joost Engelfriet	$5\frac{1}{2}$	Marek Karpiński	$4\frac{1}{6}$
Matthew Hennessy	$5\frac{1}{2}$	Thomas Wilke	$4\frac{1}{6}$
Arto Salomaa	$5\frac{1}{2}$	Christophe Reutenauer	4
Juris Hartmanis	$5\frac{1}{3}$	Marcel Paul Schützenberger	4
Andrzej Lingas	$5\frac{1}{3}$	Davide Sangiorgi	4
Burkhard Monien	$5\frac{19}{60}$	Leslie Valiant	4
Ronald Book	$5\frac{1}{4}$	Colin Stirling	4
Christian Choffrut	5		
Michael Rabin	5		
Arnold Schönhage	5		

see small canals and houses painted in many colours. By 19h we returned to San Servolo, but most participants had left at San Marco already. The ICALP banquet on Thursday evening took place in *Monaco et Gran Canal* at *Canale Grande*, about 200m from San Marco Cathedral. It ended by 23h.

PPDP 2006 (Eigth ACM SIGPLAN Symposium on Principles and Practice of declarative Programming), taking place from July 10-12, 2006, was organized by Università Ca' Foscari di Venezia and ACM SIGPLAN, and supported by the same sponsors as ICALP 2006. The organizing committee consisted of Annalisa Bossi, Michael Maher, and Agostino Cortesi.

The scientific program of PPDP 2006 consisted of 3 invited lectures and 22 contributions. The invited talks were given by Vladimiro Sassone (co-author Mikkel Bundgaard) on *'Typed Polyadic Pi-calculus in Bigraphs'*, by Thom Frühwirth on *'Constraint Handling Rules – The Story so Far'*, and (joint with ICALP) by Simon Peyton Jones on *'Composable Memory Transactions'*. The program can be found at `http://www.dsi.unive.it/ppdp2006`.

The proceedings, edited by Michael Maher, containing all contributions and invited lectures, except for that by Simon Peyton Jones and that by Tom Frühwirth only as extended abstract, have been published asa report by ACM Press, order number 550060.

LOPSTR 2006, taking place from July 12-14, 2006, was organized by Universidad Politécnica de Madrid and Università Ca' Foscari di Venezia, and supported by the same sponsors as ICALP 2006. The organizing committee consisted of Annalisa Bossi, Michele Bugliesi, Germán Puebla, and Sabina Rossi.

The scientific Program of LOPSTR 2006 consisted of 3 invited lectures and 17 contributions. The invited lectures were given by Massimo Marchiori on *'How to Talk to a Human : the Semantic Web and the Clash of the Titans'*, an interesting title, by Prakash Panangaden (joint with ICALP) on *'The One Way to Quantum Computation'*, and by Shaz Qadeer (co-author Madan Musuvathi) on *'CHESS : Systematic Stress Testing of Concurrent Software'*. The program can be found at `http://www.dsi.unive.it/lopstr2006`.

The proceeedings, edited by Germán Puebla, containing all contributions and invited lectures (as abstracts only), except for that by Prakash Panangaden, have been published as *Rapporto di Ricerca CS-2006-5* by Dipartimento di Informatica, Università Ca' Foscari di Venezia.

In the breaks coffee, tea, juice, mineral water (*San Benedetto*!), and cakes were served in the patio. Lunch was served in the cafeteria of *San Servolo*. Internet was available on 21 PC's from 10–18h, but one had to leave a deposit for a card. There was also the traditional book exhibition by Springer, as well a one by Elsevier.

Most participants stayed in the lodgings on *San Servolo* or in *Junghans* lodgings on *Giudecca*, another island. Weather was warm and humid, nearly without clouds, and highest temperatures above 30°*C*.

Thus ICALP 2006 (as well as PPDP 2006 and LOPSTR 2006) was a successful conference again, on high scientific level and well organized. And with Vladimiro Sassone at the end *'Excellent audience to excellent ICALP'*. The next ICALP will be in Wrocław, Poland from July 9–13, 2007, co-located with LICS and Logic Colloquium 2007.

Ciao Venezia and Witamy w Wrocławe.

Report on ICE-TCS

Icelandic Centre of Excellence in Theoretical Computer Science (ICE-TCS) Second Symposium on Theoretical Computer Science and Public Lecture by Moshe Vardi

Luca Aceto, Magnús Már Halldórsson and Anna Ingolfsdottir

The Icelandic Centre of Excellence in Theoretical Computer Science (ICE-TCS) is a research centre devoted to research in Theoretical Computer Science that started its activities in April 2005. It is the result of a collaboration between the Division of Computer Science, Engineering Research Institute, University of Iceland, and the Department of Computer Science, School of Science and Engineering, Reykjavík University, and is based at both institutions.

One of the yearly activities of the centre is to organize an "Icelandic Theory Day". The second event in this series was held on Wednesday, 31 May 2006, at the University of Iceland. The aim of these "theory days" is to give the Icelandic computer science community a bird's eye view of the area of Theoretical Computer Science, with emphasis on the research fields of the members of the centre.

The second symposium was graced by the presence of three outstanding invited speakers from outside Iceland, namely Wan Fokkink, Jan Kratochvil and Moshe Vardi. The presentations by the invited speakers were complemented by 25 minute talks delivered by Luca Aceto, Ragnar K. Karlsson, Magnús Már Halldórsson, and Anders Claesson/Sergey Kitaev (who shared the closing slot).

The morning session was devoted to "Volume B" talks. Moshe Vardi gave the meeting the best of starts with a talk describing the design of the ForSpec Temporal Logic, the new temporal logic of ForSpec, Intel's new formal property-specification language, which is today part of Synopsis OpenVera hardware verification language. The focus of Moshe Vardi's talk was on design rationale, rather than a detailed language description, and during the presentation he offered a very accessible discussion of the field of model checking, of linear and branching time temporal logics, and of the automata-theoretic approach to LTL model checking. Moshe Vardi also told the audience that his analysis of the relative merits of LTL and CTL has been referred to as "character assassination" by some of our colleagues!

Moshe Vardi's talk was immediately followed by the second keynote address, which was delivered by Wan Fokkink. Wan Fokkink's talk presented joint work with Bard Bloom, Rob van Glabbeek and Paulien de Wind devoted to the systematic derivation of congruence formats for various behavioural equivalences in the linear-time/branching-time spectrum from decomposition results for the modal logics that characterize them. This work offers yet another beautiful example of the usefulness of modal logics in concurrency theory.

Luca Aceto closed the morning session with a talk presenting joint work with Taolue Chen, Wan Fokkink and Anna Ingolfsdottir on the axiomatizability of bisimulation equivalence over the language BCCSP extended with the priority operator.

The afternoon session was devoted to “Volume A” talks. The first talk in that session was delivered by Jan Kratochvil, who presented a survey of work on graph homomorphisms, viz. edge-preserving vertex mappings between two graphs. He showed how the use of graph homomorphisms unifies previously defined and independently studied notions such as graph covers, role assignments, and distance constrained graph labellings. Jan Kratochvil surveyed recent results and open problems related to these notions, with special emphasis on the computational complexity issues. He also mentioned connections to the Constraint Satisfaction Problem and the Dichotomy Conjecture.

Graphs featured also in the two 25 minute talks that followed Jan Kratochvil’s address. The first talk was delivered by Ragnar Karlsson, who had defended his MSc. thesis the day before. Ragnar Karlsson presented the main results in his MSc. thesis related to so-called strip graphs. These graphs are formed by an interval graph together with an equivalence relation on the vertices, and can be used to model the classic Job Interval Selection Problem on one machine. In this problem, the input is a set of jobs, each of which is a set of intervals, and the object is to select at most one interval from each job such that no two chosen intervals intersect. This corresponds to being given multiple possible run-times for each job and trying to schedule as many jobs as possible. This problem is known to be NP-complete. However, strip graphs provide a very nice way to model the input of this problem and, by using structural observations of the input, Ragnar Karlsson was able to find a fairly efficient exponential algorithm to solve this problem.

Magnús Már Halldórsson presented the next 25 minute talk, reporting on joint work with Takeshi Tokuyama and Alexander Wolff. The scientific director of ICE-TCS considered the problem of computing a non-crossing spanning tree of a graph that has been embedded in the plane. This problem is known to be NP-hard. During his talk, Magnús Már Halldórsson considered the complexity of the problem in terms of an input parameter k: the number of pairs of edges that cross. He gave an algorithm with a dependence on k being $k^{\sqrt{k}}$, improving on recent work by Knauer et al. who gave a simple algorithm that runs in linear time, for fixed values of k; the dependence on k was 2^k.

The meeting was brought to a fitting close by Anders Claesson and Sergey Kitaev, two of the members of the ICE-TCS combinatorics group, who presented an accessible survey of work within the area of permutation patterns. A (permutation) pattern is a permutation of a totally ordered set. An occurrence of a pattern P in a permutation p is a subsequence of letters of p whose relative order is the same as that of the letters in P. As an example, the permutation 461352 has three occurrences of the pattern 321, namely the subsequences 432, 632 and 652. In 1969 Don Knuth pioneered this work by showing that the stack sortable permutations are exactly the 231-avoiding permutations. Anders and Sergey gave a brief introduction to the field, starting with a presentation of Don Knuth's result.

The symposium was attended by about 25 participants, was held in a relaxed workshop atmosphere, and was scientifically stimulating. It was pleasing to see that several MSc. students in Computer Science and Mathematics showed intellectual curiosity and attended the whole event. This bodes well for the future of research in (Theoretical) Computer Science in Iceland.

As part of its "theory week activities", ICE-TCS also hosted a public lecture by Moshe Vardi on Thursday, 1 June 2006 at Reykjavík University. This public lecture, entitled "And Logic Begat Computer Science: When Giants Roamed the Earth", was probably the event with the highest profile hosted by ICE-TCS so far, was heavily advertised and was attended by over one hundred people. (Chairs had to be brought in the room to accommodate the audience, and people were sitting along the aisles of the lecture theatre.)

During the talk, Moshe Vardi provided an overview of the unusual effectiveness of logic in computer science by surveying the history of logic in computer science, going back all the way to Aristotle and Euclid, and showing how logic actually gave rise to computer science. This was an erudite and witty lecture, full of memorable one liners. For example, Moshe Vardi told his audience that

> *Aristotle was the most influential intellectual of all times, whose wisdom stood unchallenged for over 2000 years. Now we hope for two years!*

Above all, we believe that each person in the audience learned something new about the history of thought that led to the development of computer science as we know it, and about the lives of the people involved. We certainly did and loved every minute of it.

Moshe Vardi's public talk was taped, and will soon be available on the web.

Information on the ICE-TCS centre may be found at `http://www.ru.is/icetcs/`. Future events organized under the auspices of the centre will be advertised there and will be reported on in the pages of this bulletin.

We look forward to hosting further events that will increase the visibility of Theoretical Computer Science in Iceland, and to seeing many of you in Reykjavík!

Report on Bertinoro International Center for Informatics

The "Leonardo Melandri" Programme at BICI
Bertinoro International Center for Informatics

Luca Aceto

BICI (Bertinoro International Center for Informatics) is an association whose mission is to foster cutting-edge research and advanced education (PhD and postdoctoral level) in Computer Science. BICI-sponsored events take place at the University Residential Center of the University of Bologna. The center is a wonderfully renovated XII century castle in Bertinoro, a charming old medieval town situated amidst beautiful countryside, near the byzantine-art treasures of Ravenna and not far from Bologna.

Typical events sponsored or organized directly by BICI include thematic research workshops, strategic meetings charting new research agenda and advanced schools.

In spite of the young history of BICI, from a scientific point of view its events rank on a par with those of older and more established institutions such as DIMACS, Schloss Dagstuhl and Mathematisches Forschungsinstitut Oberwolfach. BICI thus introduces a new exciting possibility for high quality scientific meetings at the international level, in wonderful surroundings.

A look at the list of coming and past events that have been held in Bertinoro shows that BICI is fulfilling its mission very well. Seventeen BICI events, attended by about 500 participants, have been held in Bertinoro in 2005, and about twenty events are expected to be held in 2006. Bookings for 2007 are already well under way. BICI has also hosted high profile international conferences like COLT 2005, the Eighteenth Annual Conference on Learning Theory.

In the past, BICI events have been sponsored by

- institutions like UNESCO, DIMACS (Center for Discrete Mathematics and Theoretical Computer Science, Rutgers, New Jersey), INDAM (Istituto Nazionale di Alta Matematica "Francesco Severi"), ETH (Swiss Federal Institute of Technology, Zurich), and NSF (National Science Foundation, USA);

- firms working in Computer Science and Information Technology like Eurotech Group, IBM Research, Microsoft Research and Yahoo!; and

- local firms like Romagna Acque.

On June 10, 2006, BICI held a press conference at the University Residential Center in Bertinoro announcing the establishment of a new form of sponsorship

Bulletin of the EATCS no 90, pp. 223–224, October 2006

for BICI events, namely the *"Leonardo Melandri" Programme*. This is a three-year sponsorship programme in favour of BICI. The sponsors are three banks or, more precisely, the foundations associated with them—namely, the Foundations of the Cassa di Risparmio di Bologna, Cesena and Forlì. The programme is named after the late Leonardo Melandri, a senator of the Italian Republic and former president of SerInAr, who was the prime mover behind the establishment of the initial funding for BICI.

With the money provided by the programme, BICI will institute a fellowship programme to allow scholars in difficult financial conditions and PhD students to take part in Bertinoro events. The funding will be used to provide junior and senior fellowships, funding for invited speakers and for event sponsorships.

The level of funding is still very far from that enjoyed by similar institutions in other countries, but it is nevertheless substantial. This is remarkable given how difficult it is to find funding for initiatives of this kind, and science in general, in Italy. As an Italian abroad, I consider this a great omen for the future.

Proposals for events under the auspices of BICI can be submitted to any member of the Executive Committee. I myself have organized two workshops in Bertinoro, and will try to do so again in the future. The location and the facilities are truly excellent, and so is the local support. I strongly encourage the members of the Theoretical Computer Science community to consider Bertinoro as a location for hosting high quality scientific events. If my experience is anything to go by, these events will be well attended, will be successful both scientifically and socially, and you will feel like visiting Bertinoro again.

REPORT ON WG 2006

The 32nd International Workshop on Graph Theoretic Concepts in Computer Science 22–24 June 2006, Sotra, Norway

Dieter Kratsch

From June 22–24, 2006, the **32nd International Workshop on Graph-Theoretic Concepts in Computer Science**, WG 2006, was held at the Norlandia Marsteinen Hotell at Sotra, near to the city of Bergen (Norway). The hotel is located on an island within a few metres from the coast. The surrounding landscape is beautiful and worth visiting the remote place.

This was the first WG outside continental europe. The 91 submitted papers (of which one was withdrawn) had authors from 29 different countries. 30 papers were accepted for presentation. In addition to these 30 regular lectures, there were two invited lectures. There were over 70 participants, from all over the world.

From the two excellent invited lectures, the first one entitled *Treewidth: characterizations, applications and computations* was given by Hans Bodlaender at Thursday morning. The talk, starting from different definitions and characterizations of the treewidth of a graph and applications, as e.g., in probabilistic and electrical networks, presented various algorithms to compute lower bounds, upper bounds or the exact treewidth of graphs; and it reported on experimental results. The second invited lecture was given Friday morning, by Tandy Warnow: *Algorithmic issues in inferring the "Tree of Life"*. This talk was also very inspiring. It reviewed several techniques in reconstructing the tree of life from DNA sequences, in particular polynomial time distance methods and algorithms to solve NP-hard problems like Maximum Parsimony. The succesful use of chordal graphs in designing the corresponding TNT software was of great interest for the WG community.

Each of the 30 accepted papers was presented at the meeting by one of the authors. The talks showed a variety of topics concerning graphs, with many of their aspects in relation to computer science. Many talks gave new or better algorithms for graph problems, some with a theoretical formulation, and some with an application. The overall quality of the presented results and the presentations was high. These talks and the two invited lectures made that WG 2006 had an excellent scientific program.

The social program consisted of a welcome reception with dinner on wednesday evening, an excursion on friday afternoon and a conference dinner on friday evening. The excursion was a 4 hour boat trip providing amazing views and

Bulletin of the EATCS no 90, pp. 225–226, October 2006

wonderful impressions of the norwegian coast region and its famous fjords. The conference dinner gave the participants an excellent taste of norwegian cuisine.

The nice location, the excellent organisation, the very interesting scientific program, and the pleasant atmosphere among the participants made this a very enjoyable meeting.

WG 2007 will be held near Jena in Germany, and this is a meeting to look forward to. The scientific program of WG 2006 is available at `http://www.ii.uib.no/wg06/program.shtml`.

Report on DLT 2006

Tenth International Conference on Developments in Language Theory June 26–29, 2006, Santa Barbara, CA, USA

Mark Daley

The Tenth International Conference on Developments in Langauge Theory (DLT 2006) was held at The University of California at Santa Barbara this summer, shortly following DCFS 2006. This marked the first time that a conference in the DLT series had been held in North America and Santa Barbara, boardered on one side by the Pacific Ocean and the Santa Ynez Mountains on the other, provided a venue rich in natural beauty and historical significance.

The conference was organized by Oscar Ibarra and Omer Egecioglu (co-chairs) with the assistance of the organizing committee of Bee Jay Estalilla, Cagdas Gerede, Jan Holtzclaw, Matthew Shayefar, Jianwen Su, Shelly Vizzolini, Sara Woodworth, and Fang Yu.

The 59 participants at DLT 2006 came from 20 countries, with the exact distribution by country listed in the table below.

DEU	11	CAN	10	USA	9	ITA	7
FRA	4	HUN	2	ZAF	2	RUS	2
ESP	1	KOR	1	JPN	1	POL	1
GRC	1	CHE	1	CZE	1	FIN	1
ROU	1	GBR	1	MLD	1	TWN	1

Table 1: DLT 2006 participant numbers by country of employment

The technical programme consisted of 4 invited lectures and 35 contributed lectures on a wide variety of language- and automata-theoretic topics encompassing both purely theoretical and more application driven work. Each day typically began with an invited lecture and continued in a single-track format for the contributed talks.

The first invited lecture was the exception to this rule with the speaker presenting at the end of the day on Monday due to air travel difficulties. Rajeev Alur (co-author P. Madhusudan) introduced the notion of *nested words* as a model of structures which have both a natural linear sequencing as well as nested hierarchical interdependencies. The study of nested words was motivated by the desire to explore the expressiveness of specification languages used in model-checking and program analysis tools. The theory regular languages of nested words was shown

Bulletin of the EATCS no 90, pp. 227–229, October 2006

to be a reformulation of the theory of visibly pushdown languages, providing a direct connection to existing work.

On Tuesday morning, Grzegorz Rozenberg's (co-author A. Ehrenfeucht) invited lecture provided insight into a new, discrete, model for the abstract description of chemical reactions. These *reaction systems* are based on two fundamental mechanisms of chemistry: facilitation/acceleration and inhibition/retardation. In contrast to traditional computational models in which states are modified by transformations, reactions are viewed as first-class citizens while structures are are secondary; in particular, reactions create states rather than transforming them. The nature of reaction systems was shown to be quite different from other existing formalizations of concurrent systems and examples relevant to both biochemistry and computer science were provided.

Wednesday's invited lecture saw a dynamic presentation by Gheorghe Păun providing both a general introduction to the rapidly-growing field of membrane computing as well as some specific details on the newly emerging theory of Spiking Neural P Systems. The Spiking Neural P Systems are inspired by ideas from neurobiology which describe the nature of inter-neuronal communication in terms of electrical pulses. The notion of a network of neurons, interconnected by various synapses was shown to be quite naturally captured within the formalism developed during the study of P systems. The interaction and mutual contribution between the distinct, yet related, fields of formal language theory and membrane computing was highlighted throughout.

The final invited lecture came in the form of a question posed by Yuri Gurevich (co-author Charles Wallace): "Can abstract state machines be useful in language theory?" A brief introduction to the theory of Abstract State Machines (ASMs) was given, with an emphasis on the often-overlooked, but critical, role of abstraction in the description of computational processes. ASMs were proposed as a "richer notion of universality" compared to, e.g., Turing Machines which, while computationally universal, are bound to a permanently fixed level of abstraction. Tools for working with directly with ASMs on a computer were also discussed and briefly demonstrated.

The programme of the contributed talks may be found at:

`http://dlt2006.cs.ucsb.edu/program.html`.

Thanks to the efforts of the organizing and programme committees, the final proceedings, published as number 4036 in Springer's *Lecture Notes in Computer Science*, edited by Oscar H. Ibarra and Zhe Dang, were available immediately at the conference.

The social program of the conference began with a reception on Monday evening, following the first day of talks. The reception was held outdoors at a facility overlooking the Pacific Ocean, allowing participants the opportunity to spot both marine mammals (opinion remains divided on whether they were dolphins or porpoises) and several local avian species.

The primary social events took place on Wednesday beginning with a trolley tour of Santa Barbara in the afternoon. The tour visited several sites of local historical interest and culminated in a visit to what has been called “the most beautiful government building in America”: The Santa Barbara County Courthouse. The courthouse was built in the late 1920’s, following the destruction of the original courthouse in a 1925 earthquake. Designed by William Mooser III, the architecture and artwork draw inspiration from a diverse collection of traditions including Spanish, Moorish and Classical Roman. The excursion was followed by a lovely dinner at the UCSB Faculty Club which showcased local Californian wines and concluded with a surprise visit from The Great Bolgani for subsequent intimate magic shows.

A beautiful collection of photos from both the conference sessions and excursion is available on the conference website at:

`http://dlt2006.cs.ucsb.edu/Photos/index.html`.

The skillful photography is courtesy of Alexander Okhotin who also provided the participants of DLT 2006 with a compelling preview of DLT 2007 to be held in Turku, Finland.

DLT 2006 was very well organized and extremely successful with both intruiging and diverse technical programs and social events. The natural beauty of Santa Barbara provided a perfect backdrop for the mathematical beauty of the invited and contributed presentations.

Report on DCM 2006

2nd Int Workshop on Development of Computational Models
16 July 2006, Venice, Italy

Manfred Kudlek

DCM 2006 (2nd International Workshop on Development of Computational Models) was held as a satellite workshop of ICALP 2006 in *Venezia* on July 16, 2006. Conference site was on the island *San Servolo*. DCM 2006 was organized by Maribel Fernández and Ian Mackie. The workshop was attended by about 30 participants. The scientific program consisted of an invited lecture and 9 contributions. It can be found at `www.dcm-workshop.org.uk/2006`.

The invited talk *'Every Computable Function is Linear (in a Sense)'* by Maribel Fernández (joint work with Sandra Álvez, Luis Damas, Mário Florido, Ian Mackie), had a curious title for people from complexity theory. It was a good and interesting survey linear λ-calculus (every arguent used only once), showing also that linear recursive functions are Turing complete, and that the system T is linear.

A nice and interesting presentation was also given by Rajagopal Nagarajan (co-authors Simon Gay, Nikolaos Papanikolaou) on quantum computing, quantum information science, and quantum cryptography. Mircea-Dan Hernest presented a demo of a program system, with the motto *From Gödel's Dialectica to Light Dialectica* (like Coca Cola Light?). A nice and interesting talk was also given by Robert K. Meyer on *BBL, the Better Bubbling Lemma* (not chewing gum!), dealing with ternary relation semantics.

The pre-proceedings, edited by Jean-Pierre Jouannaud and Ian Mackie, containing all contributions, can be found as a pdf file at the following web site `http://www.dcm-workshop.org.uk`. It will also be published electronically in ENTCS *(Electronic Notes in Theoretical Computer Science).*

DCM 2006 was a successful workshop of high scientific level.

Bulletin of the EATCS no 90, pp. 230–230, October 2006

Report on MeCBIC2006

Membrane Computing and Biologically inspired Process Calculi
9 July 2006, Venice, Italy

Manfred Kudlek

MeCBIC2006 (Workshop on Membrane Computing and Biologically Inspired Process Calculi) was held as a satellite workshop of ICALP 2006 in *Venezia* on July 9, 2006. Conference site was on the island *San Servolo*. MeCBic2006 was organized by Nadia Busi and Claudio Zandron. The workshop was supported by Università di Bologna, Università di Milano-Bicocca, and the PRIN project SYBILLA (System Biology: Modelling, Languages and Analysis). The workshop was attended by about 30 participants.

The scientific program consisted of 2 invited lectures and 15 contributions. It can be found at www.bio.disco.unimib.it/MeCBIC. The first invited talk *'Bitonal Membrane Systems'* by Luca Cardelli was a very good and interesting intoduction to a formal calculus on algebraic and topological operations for globality and locality. It was very well presented with many nice illustrations, including a short movie. Gheorghe Păun in the second invited lecture *'Membrane Computing and Brane Calculi (Some Personal Notes)'* gave an excellent presentation on the history and research in these two fields, in particular on the similarity and dissimilarity between them, showing that they are essentially similar.

Nice and interesting presentations were also given by Rudolf Freund (co-author Marion Oswald) on special *tissue P systems*, and by Maurice Margenstern (co-author Giuditta Franco) on *universal computing by floating strings*. A new form of presentation was shown by Gabriel Ciobanu (see picture page).

The pre-proceedings, edited by Nadia Busi and Claudio Zandron, containing all contributions except for the invited lecture by Luca Cardelli, have been published as a report, and will also be published electronically in *ENTCS (Electronic Notes in Theoretical Computer Science)* at the following web site www.elsevier.nl/locate/entcs.

MeCBIC2006 was a successful workshop of high scientific level.

REPORT ON GI THEORIETAG 2005

Formal languages: 15th German meeting
September 2005, Lauterbad near Freudenstadt, Germany

Henning Fernau

This annual meeting started with a one-day workshop entitled Theoretical Aspects of Grammar Induction (TAGI), followed by a couple of tutorials on Grammar Induction (directed towards formal language specialists) on the second day, and finally the mentioned workshop on Formal Languages itself, featuring nearly twenty contributed talks. The detailed program can be found on the web page of the workshop `www-fs.informatik.uni-tuebingen.de/~fernau/`. For each of the three days, a separate Technical Report containing abstracts of the talks is available from University of Tübingen.

Within the workshop, we asked the participants to write down their favorite open problems. In the following, we list these problems in alphabetical order w.r.t. the contributors. We also include the names of the contributors of the open questions, so that this hopefully initiates collaboration on the open questions.

Artiom Alhazov, Rudolf Freund, Marion Oswald: Restricting ourselves to P systems with symport / antiport rules and only one symbol, are we able to generate any recursively enumerable set of natural numbers?
A *P system* (of degree $m \geq 1$) *with symport/antiport rules* (e.g., see [2], or `http://psystems.disco.unimib.it`) is a tuple $\Pi = (O, \mu, w_1, \cdots, w_m, R_1, \cdots, R_m)$, where O is the alphabet of *objects*, μ is the *membrane structure* (it is assumed that we have m membranes, labelled with $1, 2, \ldots, m$, the skin membrane usually being labelled with 1), w_i, $1 \leq i \leq m$, are strings over O representing the *initial* multiset of *objects* present in the membranes of the system, R_i, $1 \leq i \leq m$, are finite sets of *symport/antiport rules* of the form x/y, for some $x, y \in O^*$, associated with membrane i (if $|x|$ or $|y|$ equals 0 then we speak of a symport rule, otherwise we call it an antiport rule).

An antiport rule of the form $x/y \in R_i$ means moving the objects specified by x from membrane i to the surrounding membrane j (to the environment, if $i = 1$), at the same time moving the objects specified by y in the opposite direction; the rules with one of x, y being empty are called symport rules, but in the following we do not explicitly consider this distinction. We assume the environment to contain all objects in an unbounded number. The computation starts with the multisets specified by $w_1, \ldots, w_m$ in the m membranes; in each time unit, the rules assigned to each membrane are used in a maximally parallel way, i.e., we choose a multiset of rules at each membrane in such a way that, after identifying objects inside and outside the corresponding membranes to be affected by the selected multiset of rules, no objects remain to be subject to any additional rule at any membrane. The computation is successful iff it halts.

The main results established in [1] are summarized in the following table; the class of P systems indicated by A generates exactly $NFIN$, by B generates at least $NREG$ and by d can simulate any d-register machine. A box around a number indicates a known computational completeness bound, (U) indicates a known unpredictability bound.

	Membranes					
$\lvert O\rvert$	1	2	3	4	5	m
1	A	B	B	B	B	B
2	B	1	2 (U)	$\boxed{3}$	4	$m-1$
3	1	**2** (U)	$\boxed{4}$	6	8	$2m-2$
4	2 (U)	$\boxed{4}$	**6**	9	12	$3m-3$
5	$\boxed{3}$	6	9	**12**	16	$4m-4$
6	4	8	12	16	**20**	$5m-5$
s	$s-2$	$2s-4$	$3s-6$	$4s-8$	$5s-10$	$\max\{m(s-2),$ $(m-1)(s-1)\}$

[1] A. Alhazov, R. Freund, M. Oswald: Symbol/ Membrane complexity of P systems with symport/antiport rules. *Pre-Proceedings WMC6*, Vienna (2005), 123–146.

[2] Gh. Păun: *Computing with Membranes: An Introduction*. Springer, 2002.

Henning Bordihn: The language

$$L_1 = \{\, uxvu'xv' \mid x \in \{a,b\},\ u,v,u',v' \in \{a,b\}^*,\ |u| = |u'|,\ |v| = |v'|\,\}$$

(with at least one correct copy) is known to be context-free. Is the language

$$\begin{aligned} L_2 \ &= \{\, uxvywu'xv'yw' \mid \ \ x,y \in \{a,b\},\ u,v,w,u',v',w' \in \{a,b\}^*, \\ &\qquad\qquad\qquad\qquad |u| = |u'|,\ |v| = |v'|,\ |w| = |w'|\,\} \end{aligned}$$

(with at least two correct copies) context-free?

The problem becomes easy if languages over alphabets with at least three letters or over one-letter alphabets are considered.

Henning Fernau: Is there a "unified theory" of regular language learning?

There exist lots of papers dealing with learning regular string languages, some papers on learning regular tree languages, or ω-languages, or picture languages. The proofs and algorithms are often quite similar. Is there are logical or algebraic "reason" behind? For example, the L^* algorithm of Angluin [1] has a very algebraic flavour and can thus be adapted from learning regular string languages to learning regular tree languages.

[1] D. Angluin. Learning regular sets from queries and counterexamples. *Inform. & Comput.*, 75:87–106, 1987.

Wong Karianto: Find a lower bound (with respect to the number of states) for the transformation of a nondeterministic Büchi automaton to a deterministic Muller automaton (on ω-words).

Andreas Malcher: What kinds of nondeterministic versions of DFA can be *efficiently* minimized ?

Known results: Minimization of DFA for finite sets is doable in time $O(n)$ [4] and for infinite sets in $O(n \log(n))$ [1]. However, minimization of NFA is PSPACE-complete [2], but NP-complete for restricted models: UFA [2], NFA with finite branching [3], and DFA with multiple initial states [3].

[1] Hopcroft, J.E.: An *n* log *n* algorithm for minimizing states in a finite automaton. In: Kohavi, Z. (ed.): Theory of Machines and Computations. Academic Press, New York (1971) 189–196.

[2] Jiang, T., Ravikumar, B.: Minimal NFA problems are hard. *SIAM J. Comput.* 22:6 (1993) 1117–1141.

[3] Malcher, A.: Minimizing finite automata is computionally hard. *Theor. Comp. Sci.* 327:3 (2004) 375–390.

[4] Revuz, D.: Minimisation of acyclic deterministic automata in linear time. *Theor. Comp. Sci.* 92:1 (1992) 181–189.

František Mráz: Does $\mathcal{L}(\mathsf{RWW}) = \mathcal{L}(\mathsf{RRWW})$ hold?

RRWW and RWW denote two classes of restarting automata. An RRWW-automaton [1] has a finite control with a read/write window of a fixed size. The window moves on a flexible tape with sentinels. Such an automaton works in cycles. In each cycle, it starts in its initial state with the left sentinel and the beginning of the tape in its read/write window. It moves the read/write window to the right one cell at a time until it decides (nondeterministically) to rewrite the part of the tape content in the read/write window by a shorter string. In the rewriting, also some non-input symbols can be used. Then, it can continue to move right until it *restarts*, that is, it reenters the initial state and places the read/write window over the left end of the tape. The next cycle starts on the shortened tape. The automaton halts either by performing an accept operation, in which case it accepts the input word, or by entering a configuration for which it has no instruction, in which case it rejects. An RWW-automaton is an RRWW-automaton which must restart immediately after performing a rewrite operation.

[1] P. Jančar, F. Mráz, M. Plátek, J. Vogel. Different types of monotonicity for restarting automata. In *FST&TCS'98*, LNCS 1530, Springer, 1998, 343–354.

Friedrich Otto: Gilman's Conjecture [1] A group given through a finite monoid presentation involving a confluent monadic string-rewriting system is the free product of finitely many finite groups and a free group of finite rank.

A finite *monoid presentation* is a pair $(\Sigma; R)$, where Σ is a finite alphabet, and R is a finite string-rewriting system on Σ. The system R is *monadic* if $|u| > |v|$ and $|v| \leq 1$ hold for all rules $(u \to v)$ of R. It is *confluent* if the induced reduction relation $\to_R$ is confluent. For more information, see [2].

[1] R.H. Gilman. Computations with rational subsets of confluent groups. In: J. Fitch (ed.), *EUROSAM 84, Proc.*, LNCS 174 (1984) 207–212.

[2] K. Madlener, F. Otto. About the descriptive power of certain classes of finite string-rewriting systems. *Theor. Comp. Sci.* 67 (1989) 143–172.

Daniel Reidenbach: The Equivalence Problem for E-pattern Languages

Definitions: Let Σ be a finite alphabet of *terminal symbols* – e. g. $\mathtt{a}, \mathtt{b}, \mathtt{c} \in \Sigma$ – and $X = \{x_1, x_2, \ldots\}$ an infinite set of *variables* with $\Sigma \cap X = \emptyset$. A *pattern* is a word in $(\Sigma \cup X)^+$. A morphism $\phi : (\Sigma \cup X)^* \longrightarrow (\Sigma \cup X)^*$ is *terminal-preserving* if, for every $\mathtt{A} \in \Sigma$, $\phi(\mathtt{A}) = \mathtt{A}$. A terminal-preserving morphism σ is a *substitution* if, for every $i \in \mathbf{N}$, $\sigma(x_i) \in \Sigma^*$. The *E-pattern language* $L_\Sigma(\alpha)$ of a pattern α is the set of all images of α under arbitrary substitutions, i. e. $L_\Sigma(\alpha) = \{\sigma(\alpha) \mid \sigma : (\Sigma \cup X)^* \longrightarrow \Sigma^* \text{ is a substitution}\}$.

Question: Let Σ be a terminal alphabet. Is the equivalence problem for E-pattern languages decidable, i. e. is there a total computable function which, given any pair of patterns α, β, decides whether or not $L_\Sigma(\alpha) = L_\Sigma(\beta)$?

Pattern languages have been introduced by Angluin and Shinohara [1,7]. The equivalence problem is one of the most discussed questions on the subject, but nevertheless it is widely unresolved. Problem statements, related properties and conditions have been presented in [2-6].

Conjecture (Ohlebusch, Ukkonen [5]): Let Σ be a terminal alphabet with $|\Sigma| \geq 3$, and let α, β be patterns. Then $L_\Sigma(\alpha) = L_\Sigma(\beta)$ iff there are terminal-preserving morphisms ϕ, ψ such that $\phi(\alpha) = \beta$ and $\psi(\beta) = \alpha$.

Simple examples show that it is necessary to restrict the conjecture to terminal alphabets with more than two letters: if $|\Sigma| = 1$, then $\alpha = x_1 x_1$ and $\beta = x_1 x_1 x_2 x_2$ generate the same language, but there is no terminal-preserving morphism mapping α onto β and if $|\Sigma| = 2$, then $\alpha = x_1 \mathtt{a} \mathtt{b} x_2$ and $\beta = x_1 \mathtt{a} x_2 \mathtt{b} x_3$ provides a counterexample. Recent results disprove the conjecture for terminal alphabets with three or four distinct letters [6]. With the current state of knowledge, this does not imply the undecidability of the equivalence problem. It is unkown whether counter-examples to the conjecture exist for every alphabet size.

[1] D. Angluin. Finding patterns common to a set of strings. *J. Comp. Syst. Sci.*, 21:46–62, 1980.

[2] G. Filè. The relation of two patterns with comparable language. In *Proc. 5th Annual Symposium on Theoretical Aspects of Computer Science, STACS 1988, LNCS* 294:184–192, 1988.

[3] T. Jiang, E. Kinber, A. Salomaa, K. Salomaa, S. Yu. Pattern languages with and without erasing. *Intern. J. Comp. Math.*, 50:147–163, 1994.

[4] T. Jiang, A. Salomaa, K. Salomaa, S. Yu. Decision problems for patterns. *J. Comp. Syst. Sci.*, 50:53–63, 1995.

[5] E. Ohlebusch, E. Ukkonen. On the equivalence problem for E-pattern languages. *Theor. Comp. Sci.*, 186:231–248, 1997.

[6] D. Reidenbach. On the equivalence problem for E-pattern languages over small alphabets. In *Proc. 8th International Conference on Developments in Language Theory, DLT 2004, LNCS* 3340:368–380, 2004.

[7] T. Shinohara. Polynomial time inference of extended regular pattern languages. In *Proc. RIMS Symposia on Software Science and Engineering, LNCS* 147:115–127, 1982.

Abstracts of PhD Theses

Abstract of PhD Thesis

Author: Ugo Dal Lago
Title: Semantic Frameworks for Implicit Computational Complexity
Language: English
Supervisor: Simone Martini
Institute: Università di Bologna, Italy
Date: April 27th, 2006

Abstract

This thesis is about computational complexity, programming languages and mathematical logic. The main contribution of this thesis is the study of two unifying frameworks for the analysis of complexity properties of higher-order programs and proofs.

The first one is Gonthier, Abadi and Lévy's context semantics, which is shown to be applicable to the quantitative analysis of proofs and programs. A new notion of context semantics for multiplicative and exponential linear logic is introduced and related to the complexity of normalizing proofs. Moreover, a context semantics for higher-order primitive recursion is studied, obtaining some novel characterization results for fragments of the calculus.

The second one is a framework based on realizability models. It leads to new proofs of soundness for three different subsystems of linear logic. As a preliminary step, a new invariant cost model for the pure, call-by-value lambda calculus is defined.

Semantical models described in this thesis are modifications (or generalizations) of already known models. Nonetheless, here they are shown to be applicable to the quantitative analysis of a wide range of systems. This leads to some interesting new results.

Table of Contents

Author's correspondence address

Ugo Dal Lago
LIPN
Institut Galilée
Université Paris-Nord
99, avenue Jean-Baptiste Clément
93430 Villetaneuse
France

Abstract of PhD Thesis

Author:	Gabriele Fici
Title:	Minimal Forbidden Words and Applications
Language:	English
Supervisor:	Marie-Pierre Béal and Filippo Mignosi
Institute:	Université de Marne-la-Vallée (France) Università degli Studi di Palermo (Italy)
Date:	13 February 2006

Abstract

This thesis describes the theory and some applications of minimal forbidden words, that are the most little words that do not appear as factors of a given word.

In the first part we start with the description of the properties of minimal forbidden words and we show some particular cases, as that of a finite word, a finite set of finite words, and a regular factorial language. We also present the procedures for the computation of the theoretical results.

Then we generalize the minimal forbidden words to the case of the existence of a period, which determines the positions of occurrences of the factors modulo a fixed integer. These are called minimal periodic forbidden words. We study their basic properties and we give the algorithms for the computation in the case of a finite word and of a finite set of finite words.

In the second part we show two applications of minimal forbidden words.

The first one is related to constrained systems. We give a polynomial-time construction of the set of sequences that satisfy a constraint defined by a finite list of forbidden blocks, with a specified set of bit position unconstrained. We also give a linear-time construction of a finite-state presentation of a constrained system defined by a periodic list of forbidden blocks.

The second one is a problem issued from biology, the reconstruction of a genomic sequence starting from a set of its fragments. We show that a theoretical formalization of this problem can be solved in linear time using minimal forbidden words. We also prove that our algorithm solves a special case of the Shortest Superstring Problem.

At the end of the thesis we present a detailed example of computation of our algorithm for the reconstruction of a finite word.

Table of Contents

Author's correspondence address Gabriele Fici
Dip. Informatica ed Applicazioni
Università degli Studi di Salerno
Via Ponte Don Melillo
84084 - Fisciano (SA)
Italy

European

Association for

Theoretical

Computer

Science

EATCS

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, which elects a President, Vice Presidents, and a Treasurer. Policy guidelines are determined by the Council and the General Assembly of EATCS. This assembly is scheduled to take place during the annual **I**nternational **C**olloquium on **A**utomata, **L**anguages and **P**rogramming (ICALP), the conference of EATCS.

MAJOR ACTIVITIES OF EATCS

- Organization of ICALP;
- Publication of the "Bulletin of the EATCS;"
- Award of research and academic careers prizes, including the "EATCS Award," the "Gödel Prize" (with SIGACT) and best papers awards at several top conferences;
- Active involvement in publications generally within 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: ETAPS (The European Joint Conferences on Theory and Practice of Software), 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 offered by EATCS include:
- Subscription to the "Bulletin of the EATCS;"
- Reduced registration fees at various conferences;
- Reciprocity agreements with other organizations;
- 25% discount when 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, data security, 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
- Bologne, 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
- Lisabon, Portugal 2005
- Venezia, Italy 2006
- Wrocław, Poland 2007
- Reykjavik, Iceland 2008

(2) THE BULLETIN OF THE EATCS

Three issues of the Bulletin are published annually, in February, June and October respectively. The Bulletin is a medium for *rapid* publication and wide distribution of material such as:

- EATCS matters;
- Technical contributions;
- Columns;
- Surveys and tutorials;
- Reports on conferences;
- Information about the current ICALP;
- Reports on computer science departments and institutes;
- Open problems and solutions;
- Abstracts of Ph.D.-Theses;
- Entertainments and pictures related to computer science.

Contributions to any of the above areas are solicited, in electronic form only according to formats, deadlines and submissions procedures illustrated at `http://www.eatcs.org/bulletin`. Questions and proposals can be addressed to the Editor by email at `bulletin@eatcs.org`.

(3) OTHER PUBLICATIONS

EATCS has played a major role in establishing what today are some of the most prestigious publication within theoretical computer science.

These include the *EATCS Texts* and the *EATCS Monographs* published by Springer-Verlag and launched during ICALP in 1984. The Springer series include *monographs* covering all areas of theoretical computer science, and aimed at the research community and graduate students, as well as *texts* intended mostly for the graduate level, where an undergraduate background in computer science is typically assumed.

Updated information about the series can be obtained from the publisher.

The editors of the series are W. Brauer (Munich), J. Hromkovic (Aachen), G. Rozenberg (Leiden), and A. Salomaa (Turku). Potential authors should contact one of the editors.

EATCS members can purchase books from the series with 25% discount. Order should be sent to:

Prof.Dr. G. Rozenberg, LIACS, University of Leiden,
P.O. Box 9512, 2300 RA Leiden, The Netherlands

who acknowledges EATCS membership and forwards the order to Springer-Verlag.

The journal *Theoretical Computer Science*, founded in 1975 on the initiative of EATCS, is published by Elsevier Science Publishers. 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.

The Editors-in-Chief of the journal currently are G. Ausiello (Rome), D. Sannella (Edinburgh), G. Rozenberg (Leiden), and M.W. Mislove (Tulane).

ADDITIONAL EATCS INFORMATION

For further information please visit `http://www.eatcs.org`, or contact the President of EATCS:

Prof. Dr. Giorgio Ausiello, Dipartimento di Informatica e Sistemistica
Universita di Roma "La Sapienza", Via Salaria 113, 00198 Rome, ITALY
Email: `president@eatcs.org`

EATCS MEMBERSHIP

DUES

The dues are €30 for a period of one year. A new membership starts upon registration of the payment. Memberships can always be prolonged for one or more years.

In order to encourage double registration, we are offering a discount for SIGACT members, who can join EATCS for €25 per year. Additional €25 fee is required for ensuring the *air mail* delivery of the EATCS Bulletin outside Europe.

HOW TO JOIN EATCS

You are strongly encouraged to join (or prolong your membership) directly from the EATCS website `www.eatcs.org`, where you will find an online registration form and the possibility of secure online payment. Alternatively, a subscription form can be downloaded from `www.eatcs.org` to be filled and sent together with the annual dues (or a multiple thereof, if membership for multiple years is required) to the **Treasurer** of EATCS:

Prof. Dr. Dirk Janssens, University of Antwerp, Dept. of Math. and Computer Science
Middelheimlaan 1, B-2020 Antwerpen, Belgium
Email: `treasurer@eatcs.org`, *Tel: +32 3 2653904, Fax: +32 3 2653777*

The dues can be paid (in order of preference) by VISA or EUROCARD/MASTERCARD credit card, by cheques, or convertible currency cash. Transfers of larger amounts may be made via the following bank account. Please, add €5 per transfer to cover bank charges, and send the necessary information (reason for the payment, name and address) to the treasurer.

Fortis Bank, Bist 156, B-2610 Wilrijk, Belgium
Account number: 220–0596350–30–01130
IBAN code: BE 15 2200 5963 5030, SWIFT code: GEBABE BB 18A