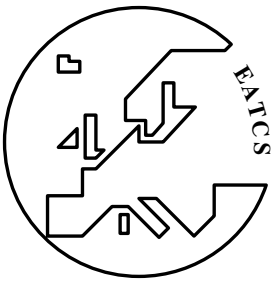
ISSN 0252–9742

Bulletin

of the

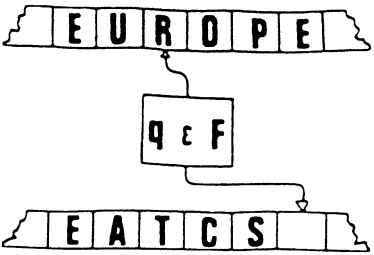
European Association for Theoretical Computer Science

EATCS



Number 84 October 2004

Council of the European Association for Theoretical Computer Science



Board:

President:	Mogens Nielsen	Denmark
Vice Presidents:	Jan van Leeuwen	The Netherlands
	Paul Spirakis	Greece
Treasurer:	Dirk Janssens	Belgium
Secretary:	Branislav Rovan	Slovakia
Bulletin Editor:	Vladimiro Sassone	United Kingdom

Other Council Members:

Pierpaolo Degano	Italy	Juhani Karhumäki	Finland
M. Dezani-Ciancaglini	Italy	David Peleg	Israel
Josep Díaz	Spain	Jiří Sgall	Czech Republic
Zoltán Ésik	Hungary	Andrzej Tarlecki	Poland
Javier Esparza	Germany	Wolfgang Thomas	Germany
Hal Gabow	USA	Dorothea Wagner	Germany
Alan Gibbons	UK	Emo Welzl	Switzerland
Kazuo Iwama	Japan	Gerhard Wöeginger	The Netherlands
Jean-Pierre Jouannaud	France	Uri Zwick	Israel

EATCS Monographs and TCS:

Monographs Editors:	Wilfried Brauer	Germany
	Grzegorz Rozenberg	The Netherlands
	Arto Salomaa	Finland
TCS Editors:	Giorgio Ausiello	Italy
	Michael Mislove	USA
	Don Sannella	United Kingdom

Past Presidents:

Maurice Nivat	(1972–1977)	Mike Paterson	(1977–1979)
Arto Salomaa	(1979–1985)	Grzegorz Rozenberg	(1985–1994)
Wilfred Brauer	(1994–1997)	Josep Díaz	(1997–2002)

EATCS Council Members

EMAIL ADDRESSES

Giorgio Ausiello . . . . . ausiello@dis.uniroma1.it
Wilfried Brauer . . . . . brauer@informatik.tu-muenchen.de
Pierpaolo Degano . . . . . degano@di.unipi.it
Mariangiola Dezani-Ciancaglini . . . . . dezani@di.unito.it
Josep Díaz . . . . . diaz@lsi.upc.es
Zoltán Ésik . . . . . ze@inf.u-szeged.hu
Javier Esparza . . . . . esparza@informatik.uni-stuttgart.de
Hal Gabow . . . . . hal@research.cs.colorado.edu
Alan Gibbons . . . . . amg@dcs.kcl.ac.uk
Kazuo Iwama . . . . . iwama@kuis.kyoto-u.ac.jp
Dirk Janssens . . . . . Dirk.Janssens@ua.ac.be
Jean-Pierre Jouannaud . . . . . jouannaud@lix.polytechnique.fr
Juhani Karhumäki . . . . . karhumak@cs.utu.fi
Jan van Leeuwen . . . . . jan@cs.uu.nl
Michael Mislove . . . . . mwm@math.tulane.edu
Mogens Nielsen . . . . . mn@brics.dk
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@susx.ac.uk
Paul Spirakis . . . . . spirakis@cti.gr
Andrzej Tarlecki . . . . . tarlecki@mimuw.edu.pl
Wolfgang Thomas . . . . . thomas@informatik.rwth-aachen.de
Dorothea Wagner . . . . . Dorothea.Wagner@uni-konstanz.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, Sussex, BN1 9QH, 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.

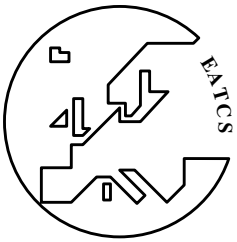
The details can be found at `http://www.eatcs.org/bulletin`, including class files, their documentation, and guidelines to deal with things such as pictures and overfull boxes. When in doubt, email `bulletin@eatcs.org`.

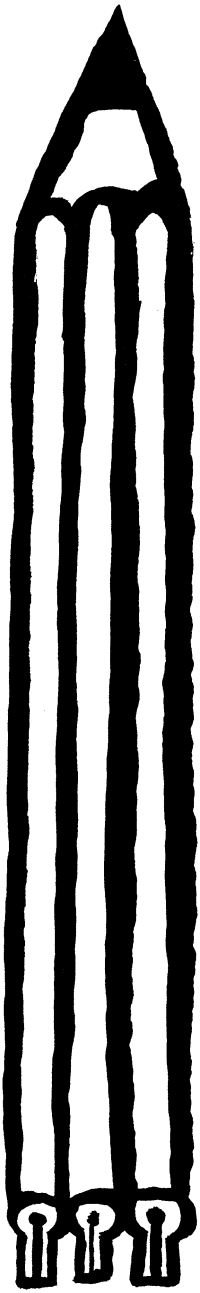
Deadlines for submissions of reports are January, May and September 15th, respectively for the February, June and October issues. Editorial decisions about submitted technical contributions will normally be made in 6/8 weeks. Accepted papers will appear in print as soon as possible thereafter.

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.

Table of Contents

EATCS MATTERS





Letter from the President

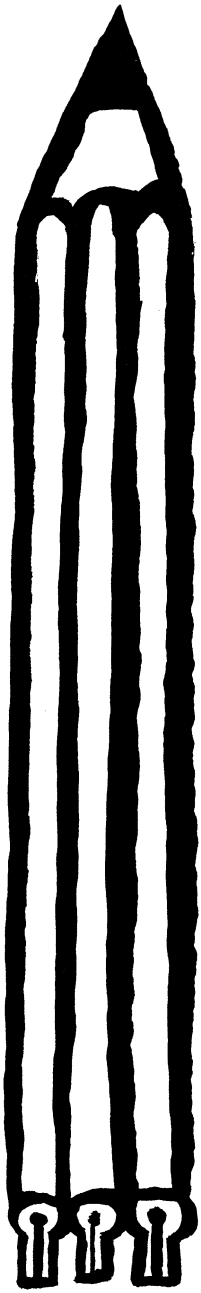
Dear EATCS members,

As many of you will know, we had a very successful 31st ICALP in Turku this year, co-located with the annual IEEE Symposium on Logic in Computer Science, LICS'04. We are grateful to Juhani Karhumäki and all his colleagues in Turku for an excellent organization, and also to Phokion Kolaitis and our colleagues from the LICS community for making the event so special.

On top of the two conferences, we had a number of interesting additional events, including a number of workshops, the Gödel prize ceremony, and the presentation of the 2004 EATCS Award . You will find many more details in this Bulletin.

As usual we also had the annual EATCS General Assembly during ICALP, and you will find a report from this in this Bulletin. As you will see in the report, the number of EATCS members has increased during the past year, following a decrease in recent years. I hope that all our members will continue to encourage new young researchers in theoretical computer science to join our organization.

Also you will see a number of new EATCS initiatives mentioned, including an EATCS Grand Challenge Initiative, based on a series of workshops aimed at identifying the major long term research challenges for our field. It is planned that this initiative will be launched at a workshop during ICALP'05.



The planning of ICALP'05 in Lisbon is running very well. I hope that we will see again a record number of submissions and workshop proposals. Please look out for the call for papers.

At the General Assembly in Turku, Venice was unanimously chosen as the venue for ICALP'06, to be organised on the Island of San Servolo by Michele Bugliesi, University of Venice Ca' Foscari. Also, it was decide to continue in 2006 with the new format of ICALP to be introduced in 2005.

If you have any views on this or other EATCS matters, please let us know.

Mogens Nielsen, Aarhus
September 2004

Letter from the Bulletin Editor

Dear Reader,

*Welcome to the October 2004 installment of the Bulletin of the EATCS. Coming just after major conferences, prizes and awards, October issues are usually very rich, and this one is no exception. ICALP 2004 in Turku was a great success. Besides the usual 'best paper' awards, it assigned the "*Gödel Prize*" for outstanding papers in theoretical computer science to* Maurice Herlily, Nir Shanit, Michael Saks *and* Fotios Zaharoglou*, for their landmark papers on distributed computation, and the "*EATCS Award*" for distinguished achievements to our own* Arto Salomaa*. The award ceremony was completed by two wonderful talks, which you'll be able to read about in the Bulletin. Arto's talk - a witty recollection of over 40 years of scientific 'encounters' - is recorded in this issue in the paper "*Myhill, Turku and Sauna Poetry,*" and a paper is in preparation by Maurice Herlily and Nir Shavit on the result that won them the Gödel Prize. It will appear in the next issue of our regular "*Distributed Computing Column.*"*

We recently endured three very sad losses: Shimon Even, Harald Ganzinger, and Larry Stockmeyer. I hope to publish papers in the coming issues to celebrate the lives and activities of such remarkable, distinguished members of our community.

This issue marks my first year as the Bulletin Editor. I notice with satisfaction things getting easier, doubtless because of the efforts of column editors, regular and occasional contributors, and authors altogether, all of whom I thank heartily.

Enjoy

Vladimiro Sassone, Sussex
September 2004

ICALP 2004

Report on the EATCS General Assembly 2004

The 2004 General Assembly of EATCS took place on Tuesday, July 13th, 2004, at the Mauno Koivisto Centre in Turku, the site of the ICALP. President Mogens Nielsen opened the General Assembly at 19:05. The agenda consisted of the following items.

Report of the EATCS President

Mogens Nielsen reported briefly on the EATCS activities between ICALP 2003 and ICALP 2004. 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.

The number of members of EATCS increased after many years of gradual decline. It is hoped that this trend will continue in the future. The financial matters improved as well. By reducing the number of pages of the last issue of the bulletin the bulletin editor Vladimiro Sassone helped to keep the income and expenses for the last year in balance. In an attempt to reduce the strain on the budget and taking into account that the last increase in the membership dues was in 1996 the Council of EATCS has decided to *increase the membership fee* to €30.

Mogens Nielsen informed the GA about further activities the Council has decided EATCS should undertake. These include the Grand Challenge Initiative (to identify the grand challenges for theoretical computer science, based on the annual workshops held at ICALPs), the Revision of the Statutes (mainly to correct some inconsistencies and remove some restrictions that hinder proper functioning, e.g., time constraint on the Council elections to be held shortly after ICALP places the election period to the holiday season, removing explicit mention of specific publications in the Statutes, etc.). The president reported on the new composition of the award committees, where the EATCS part of the Gödel Prize committee will consist of G. Ausiello, P.-L. Curien, and P. Vitanyi and the EATCS Award committee consists of J. van Leeuwen (chair), M. Dezani-Ciancaglini and W. Thomas. The Council also decided to keep the new structure of the ICALP 2005 also for ICALP 2006 (topics and their number open so far).

EATCS continued to sponsor prizes for the best papers or best student papers at conferences (ICALP, ETAPS, ESA and MFCS), sponsor conferences and acknowledges activity of its chapters. More details in the report on the web.

ICALP 2004. Juhani Karhumäki, the general ICALP 2004 chair, gave information about the local arrangements. It was for the first time ICALP returned to the same city. ICALP 1977 was held in Turku. ICALP 2004 was collocated with LICS, and 11 pre/post-conference workshops, one of them dedicated to the 70^{th} birthday of Arto Salomaa. A record number of participants (275) registered for ICALP itself and there were 340 registered for ICALP and LICS together. Including all the workshops there were 430 participants registered. ICALP and LICS were united in certain aspects (joint plenary talks, social events, overlap in time) and divided in others (held in adjacent but different buildings, run in their own tradition). Most of the record number of submissions arrived in the very last day(s), thus keeping the organizers on their tiptoes to the last minute. Due to some technical error the ICALP proceedings were not ready for the conference but will be mailed to all participants. As a compensation a CD, one month free access to the proceedings on the web and printouts of the papers were provided to participants.

There were separate Program Committees for Track A and Track B. Josep Díaz reported on the track A. He thanked Arto Lepisto, Mika Hirvensalo, Petri Salmela, and others from the Turku team for aptly stepping in and helping with the Program Committee support when Juhani was forced to be in a hospital. There were 272 papers submitted, 2 of them withdrawn, 69 accepted and one of them withdrawn afterwards. This was a record number of submissions for track A. There were 4 papers rejected due to double submission. The 20 members of the Track A PC deliberated electronically and 9 of them took part in the physical selection meeting in Barcelona. The reported breakdown by topics and countries can be found in Manfred Kudlek's report in this issue of the Bulletin. Josep Díaz also thanked many referees (a list flashed on the overhead projector) some of whom wrote really detailed reports (6-8 pages of comments). Don Sannella reported on the track B which had also a record number of submissions (107) out of which 1 was rejected as a double submission and 28 accepted. The competition was very high this year and many very good papers had to be rejected. There was no physical meeting of the PC. The 2 weeks electronic discussion was very detailed (the record size of the discussion for one paper was 32KB), almost all papers had 4 reports and the discussion lead to visible deviation from accepting simply by weighted average as demonstrated by a graphical representation.

Mogens Nielsen kept the tradition presenting Josep Díaz, Juhani Karhumäki, and Don Sannella with small gifts of the President and thanked all of them for the excellent work done.

ICALP 2005. Luís Monteiro reported on the organisation of the ICALP 2005 in Lisabon on July 11–15, 2005. He presented basic information (including expected conference fee €350, hotel prices €60–150). The conference site will

be the Guggenheim Foundation Centre, the associated workshops and hotels will be in a walking distance. The ICALP 2005 will have the new structure, adding to the traditional two tracks a third track devoted in 2005 to Security and Cryptography Foundations. The Program Committees for the three tracks are almost complete. The three chairs are Giuseppe Italiano, Catuscia Palamidessi, and Moti Yung. More information can be found at `http://icalp05.di.fct.unl.pt`.

ICALP 2006. Mogens Nielsen announced that the only contender for the ICALP 2006 he is aware of is the University of Venice. When nobody from those present brought up another proposal Michele Bugliesi presented basic information about the Department of Computer Science, the conference site (Island of San Servolo), the conference center, accommodation facilities, and the plans of the organizers (including expected conference fee, hotel prices, etc.). The presented tentative dates for ICALP were (on requests from the audience and after considering other meetings taking place) fixed to the 'usual' ICALP week – July 10–14, 2006. The General Assembly approved Venice as the site for ICALP 2006.

Report of the EATCS Bulletin Editor

Vladimiro Sassone reported on the first year of his Bulletin editorship. He thanked to the Column Editors for keeping to provide a distinctive feature to the bulletin and an important benefit to the EATCS members. He also thanked the News Editors and listed all other contributors to the three issues of the bulletin that appeared since the last ICALP. He gave a brief statistics on the number of pages printed, the costs involved and thanked to all who helped with the technical matters. Finally he announced a call for a new cover design for the bulletin. Mogens Nielsen thanked to Vladimiro for great job he has done on the new layout of the bulletin, for making it available in an electronic form and arranging the mailing.

Report of the TCS editors

Don Sannella reported on TCS. There were 20 volumes (about 12000 pages) printed within the past year. TCS opened the third 'track' - TCS 'C' - devoted to theoretical aspects of Natural Computing, lead by Grzegorz Rozenberg who was asked to comment on the first issue that just appeared. Don continued by informing that the time for production was reduced to less than a year now and the backlog reduced. There will soon be another improvement introduced - a web based system supporting the work of the editors. Without attaching any significance to the impact factor he announced that this parameter has improved for TCS by 50% in the last year.

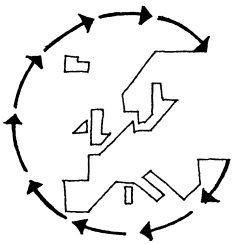
EATCS Texts and Monographs

Wilfried Brauer reported on the EATCS series published by Springer-Verlag. It is the 20^{th} anniversary of the series and the 30^{th} anniversary of cooperation with Springer (which started to publish the ICALP Proceedings as of the second ICALP in 1974). He commented on an excellent collaboration, stressing the importance of the fact there was very little change in the Springer personell EATCS dealt with over those years (with I. Mayer since 76 and Dr. Woessner since 1985). Reminding some history, the first three volumes (by Kurt Mehlhorn) were unveiled at the ICALP 1984, by 1994 there were 28 volumes published. In 1994 the new colored cover design was introduced and the new series of EATCS Texts was introduced. By 2004 20 volumes of Texts were printed and 13 more monographs. Three monographs and 5 texts were published since the last ICALP and about 8 more are to come by the next ICALP. He thanked to the Springer Verlag and explicitly to Mrs. Mayer and Dr. Woessner present in the audience for the work on the series.

Specials. Manfred Kudlek presented the statistics of the authors who published repeatedly at ICALP and announced that Kurt Mehlhorn is the second person to exceed the 10 (full paper equivalents) papers at ICALP and he is getting the second EATCS Golden badge (the first went to Jean-Eric Pin earlier). There was nobody crossing the 5 paper boundary for Silver badge at this ICALP. By tradition Manfred awarded the EATCS badges to the four editors of the ICALP 2004 proceedings. Mogens Nielsen presented on behalf of Juhani Karhumäki gifts to the seven ICALP 2004 participants who also participated at ICALP 1977 held in Turku (G. Ausiello, M. Kudlek, B. Monien, M. Nivat, A. Paz, G. Rozenberg, and J. van Leeuwen).

At this point, at 21:05, Mogens Nielsen thanked all present and concluded the 2005 General Assembly of the EATCS.

Branislav Rovan



The Distinguished Achievements Award

EATCS AWARD 2004

In a special afternoon ceremony on July 13th during ICALP'2004 in Turku's Mauno Koivisto Center, the EATCS Distinguished Achievements Award 2004 was awarded to

Professor Arto Salomaa

from the University of Turku, Finland, for his outstanding contributions to theoretical computer science. The Award was given to him by the EATCS president, Mogens Nielsen (Aarhus).

Arto's first publications in theoretical computer science appeared around 1964, exactly forty years ago. These publications were devoted to the theory of Moore automata and the algebra of regular events, and they appeared in fact in the Annals of the University of Turku. Many papers followed, first on finite automata and variants like probabilistic and time-varying automata and later on formal language theory. From 1969 onward, Arto began contributing extensively to the theory of rewriting systems, from extensions of context-free grammars to (especially) the revolutionary notions of parallel rewriting in so-called Lindenmayer systems and the new views they created for language theory. Arto is without doubt the founder of automata and formal language theory in Europe.

However, Arto's contributions to theoretical computer science go much further than this. As Michael Arbib wrote 35 years ago: *Automata Theory may be defined, approximately, as the mathematical investigation of the general questions raised by the study of information processing systems, be they men or machines.* It is an accurate statement for theoretical computer science in general, but it also is a perfect characterization of Arto's work. Already in the late 1960's the 'general questions raised by the study of information processing systems' began to influence Arto's research, ranging from many themes in what was called 'unusual automata theory' to the work on combinatorics of words and morphisms. From the late 1980's onward Arto's work on public key cryptography begins to appear, as do his many in-depth studies of *bio-inspired* rewriting systems.

Arto's remarkable record as author and co-author of more than 25 books (including translations in languages like Chinese, Japanese, Russian and Roumenian) shows that his books often consolidate entire research fields. His books on *Theory of Automata* from 1969 and on *Formal Languages* from 1973 were the starting point for many later researchers in theoretical computer science. But other books have followed, on L-systems (with Grzegorz Rozenberg), on Formal Power Series (with Matti Soittola), on the semiring-approach to automata and language theory (with Werner Kuich), on Public-Key Cryptography, and on DNA-Computing (with Gheorghe Paun and Grzegorz again). His *Jewels of Formal Language Theory* was the first of several books that showed the beauty of theoretical computer science as a mathematical discipline. Also, Arto was one the editors (with Grzegorz) of the impressive three-volume *Handbook of Formal Languages* that appeared in 1997.

For over forty years, Arto has played a leading role in theoretical computer science, as editor (for over ten journals), as invited lecturer at conferences, and as guest lecturer in many computer science departments in the world. Close to 25 PhD researchers graduated under his supervision, many of whom are now respected scientists themselves and hold important positions at universities in Finland and elsewhere. Many people were inspired by their contacts with Arto, his excellent style as a computer scientist and mathematician, his great intuitions for insightful theory and clear writing in unimitable Finnish style.

Arto received his PhD in Mathematics in 1960 at the University of Turku. He became a Professor of Mathematics in Turku in 1966 and held visiting positions at the Universities of Western Ontario (London), Aarhus (Denmark), and Waterloo (Canada). In 1970 Arto became a member of the Finnish Academy of Science and, for several periods in his career, he was a research professor at the Academy of Finland. Since 2001 he holds the highly distinguished title of 'Academician' in the Academy. Arto is also a member of the Academia Europaea, and of the Hungarian Academy of Sciences. He is a 'doctor honoris causa' at seven universities and recipient of several highly esteemed prizes and other distinctions.

Last but not least, Arto has a long and active record within EATCS: he was member of the EATCS Council for many years from the founding of the Association in 1973 onward, President of the Association from 1979 till 1985, co-founder and -editor (since 1983) of the series EATCS Monographs and Texts in Theoretical Computer Science, editor of the Formal Language Theory column in the EATCS Bulletin, and not to forget, program committee member of 14 ICALP's and pc chair of ICALP twice, in 1977 (Turku) and in 1988 (Tampere).

Jan van Leeuwen

Myhill, Turku and sauna poetry: Recollections arising from the EATCS Award

Arto Salomaa

My presentation will consist mainly of personal views and recollections. First, I am of course very grateful for the great honor. I want to share it with my whole clan, so to speak. I have been very lucky in that I have had most wonderful collaborators and students during all stages of my career. It is perhaps quite common in science, more so than in other realms of life, that your best collaborators become also your best friends. This has certainly happened in my case. The editors of my Festschrift were J. Karhumäki, H. Maurer, G. Paŭn and G. Rozenberg, whereas G. Rozenberg, M. Nivat, W. Kuich, W. Thomas, G. Paŭn and S. Yu were speakers in my Festival Colloquium last Sunday. I will mention some other names here later but do not try to be exhaustive in any sense. I also cannot list here all my really great students, many of whom have become remarkable scientists. One can say that the student-teacher relation has been reversed.

I have been very fortunate also in participating in EATCS and ICALP activities from their very beginning. The activity that should be especially mentioned now is our Monograph and Text Series, because of the 20th Anniversary. The cooperation between the editors and the publisher has been great. I think our series is good and continues to be strong.

When I speak of my clan, my family is of course central: my wife, children and grandchildren. In my web pages I mention sauna and grandchildren as my special interests. Now when I am retired I have more time with my grandchildren. Sauna I always had time for.

It is quite special for me to get this award in my home town Turku. I was born here and, more importantly, I satisfy practically all of the criteria characterizing a genuine *Turkuer*, Turku-citizen. Some of the criteria are unsatisfiable for newcomers, for instance, being born in the Heideken hospital (doesn't exist any more), or seen Paavo Nurmi running (not possible any more). Some others are still satisfiable like swimming across the Aura river or learning the Turku dialect.

After this discussion on the significance of the award, I would like to mention a couple of recollections from the early days. My first acquaintance with finite automata and regular languages stems from the spring 1957 when I was a student in John Myhill's seminar in Berkeley. The topics were chosen from the newly

appeared red-cover Princeton book '*Automata Studies.*' For instance, the Kleene basic paper about the '*representation of events in nerve nets*' is in this book. Myhill also gave lectures himself and was very impressive, to say the least. He was also out of this world and occasionally taken to a sanatorium. Once he did not show up at all. Eventually we found him in a wrong room with no audience. He was lecturing, the board already half-full. Myhill's speech was referred to as the *Birmingham accent* and was not easy to understand, at least not for me. Once he formulated a theorem about regular events and, at the end of the lecture, asked us to prove the converse. Next time he was very upset when nobody had done it. Later on I found out that the theorem and its converse constitute what is now known as the *Myhill-Nerode Theorem.*

I took courses also from Alfred Tarski. Some later well-known people, for instance Roger Lyndon, were in the audience. Of the big names I had some contacts also with Alonzo Church. He even visited Turku in the 70's.

Marco Schützenberger would surely have received the EATCS Award had he still been alive when the Award was initiated. As a person he was most remarkable and memorable. I always had the feeling that I lagged two steps behind in understanding his arguments and jokes. I met Schützenberger first time in Paris in March 1971. He asked me all kinds of questions about ω-words. Most of the matters I could say nothing about. Schützenberger was an invited speaker in the first Turku ICALP in 1977. His lectures were not easy to follow. Seymour Ginsburg said that *it doesn't make any difference whether Marco lectures in English or French, I don't understand it anyway.* In my experience it made a difference, though. Usually at a conference, if Schützenberger gave his talk in French, so did all the other French participants.

I gave my first course on computability theory in 1962. I used the book of Martin Davis which was the only one available. There were some thirty students in their 3rd to 5th year of studies. As far as I remember, they were very enthusiastic and did not mind the detailed machine language constructions of the book. They would undoubtedly seem tedious nowadays. I also had seminars on automata theory, mostly based on Russian literature which was quite strong on the subject in mid-60's. In recent years there has been a renaissance of Glushkov-type constructions with finite automata. My courses were courses in mathematics. Computer science was still in its formation stages.

Let us go then to the early 70's when both ICALP and EATCS started, Maurice Nivat being the key person in both efforts. EATCS did not exist at the time of the first ICALP in Paris in 1972, only later ICALP became the conference of the EATCS. Also the participants of the Paris ICALP did not get the impression that the Paris ICALP would start a series of conferences, as it actually did. A somewhat similar meeting took place in Haifa in 1971, with no direct continuations.

The Programme Committee of the Paris ICALP consisted of C. Böhm, S. Eilenberg, P. Fischer, S. Ginsburg, G. Hotz, M. Nivat, L. Nolin, D. Park, M. Rabin, A. Salomaa, M.P. Schützenberger (Chair), and A. van Wijngaarden. Everybody received all the submissions, and there were no specific rules for the work. I guess Maurice Nivat did most of the work. Much of the correspondence concerning the conference and the committee was in French.

I was those times quite much with Seymour Ginsburg. Still in the early 90's he tried to buy from Turku our statue of Lenin, to some garden in California, but the details did not work out. The statue still stands in its place. Also Mogens Nielsen and Jan van Leeuwen, let alone Grzegorz Rozenberg, entered my life as top researchers in Lindenmayer systems. I and Grzegorz were smoking like chimneys, and I had yet no idea what a pair of researchers we would later form.

It is hard to visualize nowadays the scientific landscape and academic surroundings in the early 70's in fields now referred to as *computer science.* Everything was unorganized and scattered. Automata theory was already a very advanced mathematical theory. On the other hand, what was labeled as *theoretical computer science* or *theoretical informatics* varied enormously from place to place. It could be numerical analysis or some parts of theoretical physics. Some order and uniformity was called for, this was the motivation behind EATCS. The impact was not restricted to Europe, both EATCS and ICALP were quite international from the very beginning.

To give an idea about theoretical computer science in the early 70's, I quote from the editorial of Maurice Nivat in the first issue of the EATCS Bulletin. The issue itself is a real collectors's item; only a few copies exist any more.

> As the major points of Theoretical Computer Science the following fields of interest are considered.
>
> - Theory of Automata
> - Formal Languages
> - Algorithms and their complexity
> - Theory of programming, if such a thing exists: we may now consider as chapters of this theory formal semantics, proving properties of programs, and all applications of logical concepts and methods to the study and design of programming languages.
>
> The list is by no means limitative: defining the limits of Theoretical Computer Science is at least as difficult as defining the limits of Computer Science itself. And we strongly believe that a science is what the scientists at work make it: certainly new areas of Computer Science will be open to theory in a very near future. Let us *start small and grow*.

Some of this is equally valid now as it was thirty years ago. The program of this conference shows the tremendous expansion of Theoretical computer science. New areas such as bio-computing and quantum computing offer challenging new problems. However, some fundamental theory is still missing also in classical areas such as words, trees or context-free languages. I do not want to predict which problems or areas will be the most significant. I can only quote the famous baseball player Yogi Berra: *The future ain't any more what it used to be.*

The original meaning of the abbreviation ICALP was *International Colloquium of Automata, Languages and Programming.* It reflects also the quoted editorial. This interpretation is now too narrow. Numerous other suggestions have been proposed. Here are some from an old speech of mine:

Interesting Combination of Attractive and Lovely Problems,
Intensive Course in Almost Logarithmic Pleasure,
Ideal Choice of Adorable and Loaded Programs,
Interesting Conference Always Leads to Progress,
Immense Cordiality Added to a Luxurious Program.

Since we are now in Finland, I will say very little about sauna, sauna itself being much better than any talk about it. Sauna has many positive effects. In the MSW research group, with Hermann Maurer and Derick Wood, we spoke of "three-sauna problems", instead of the Sherlock Holmes "three-pipe problems". Sauna has such an effect of opening the veins in your brain. My guests have provided me with many poems about sauna, I have a whole collection of sauna poetry. Nobody can be more convincing than Werner Kuich:

> Salosauna, Finnische Freunde, Ruhiges Rauhala
> allzulange entbehrt.
>
> Kaarina, die kundige Köchin,
> and Arto, den Allgewaltigen
> sowie Salosauna,
> grüsst Werner aus Wien
> der Deutsche Dichter.
>
> Wer diese *letzten Sieben* überlebt hat,
> der kann wahrlich sagen,
> dass ihm nichts Saunamässiges mehr fremd ist.

I want to conclude with a greeting I heard from the late Ron Book the last time I met him:

Good Theorems, Good Theorems!!

EATCS AWARD 2005

Call for Nominations

EATCS annually distinguishes a respected scientist from our community with the prestigious EATCS Distinguished Achievements Award. The award is given to honour extensive and widely recognised contributions to theoretical computer science over a long period of the career, scientific and otherwise.

For the EATCS Award 2005, 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 1, 2004**.

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

Professor Jan van Leeuwen
Institute of Information and Computing Sciences
Utrecht University
3508 CH Utrecht
The Netherlands

Email: jan@cs.uu.nl

Previous recipients of the EATCS Award are

2000:	R.M. Karp	2003:	G. Rozenberg
2001:	C. Böhm	2004:	A. Salomaa
2002:	M. Nivat		

The next award is to be presented during ICALP 2005 in Lisbon.

Report from the Japanese Chapter

O. Watanabe (Tokyo Inst. of Tech.)

EATCS/LA Workshop on TCS

The third EATCS/LA Workshop on TCS will be held at Univ. Kyoto, Research Institute of Mathematical Sciences, January 31 ~ February 2, 2005. 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 announced early January. 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!

On TCS Related Activities in Japan

1. New Research Project Has Started!

A new research project on algorithms and computation proposed by a team of active TCS Japanese researchers lead by *Prof. Kazuo Iwama* has been approved by Japanese government as one of the projects of *Scientific Research on Priority Areas*, one of the biggest grant categories that the government offers. The project term is four years and the budget size (in four years) is about 485 million yen.

This new project — New Horizons of Computation: How to Overcome Them — aims to investigate how to guarantee the performance of algorithms based on the "social impact" and to develop a new paradigm for the design and analysis of such algorithms. To this goal, the following three major approaches are proposed in the project: (i) Model Research, search for computational models reflecting the social impact, (ii) Lower-Bound Research, investigation of performance limits, and (iii) Upper-Bound Research, development of novel algorithms.

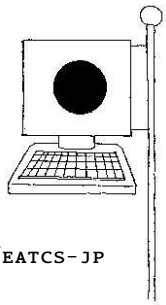
The project consists of approximately 30 independent but closely related subprojects, each of which is proposed by a research group chaired by a top TCS researcher in Japan. Most of the grant will be divided into subprojects in advance, but some not-so-small fraction of it is reserved for common purposes, such as holding meetings and inviting people. For example, it is planned to have a kick-off meeting on February, 2005, in Kyoto, whose organization has already

started under the chair of Prof. Magnús Halldórsson. If you have an interest on this project, please contact Prof. Iwama (iwama@kuis.kyoto-u.ac.jp).

2. TGCOMP Meetings, January ~ June, 2004

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 2004, *TGCOMP* organized 5 meetings and about 50 papers were presented there. See our web page for the list of presented papers (title, authors, key words, email).

The Japanese Chapter



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

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

BRICS
Basic Research
in Computer Science

Computer Science Departments of
University of Aarhus and Aalborg University

Coming Events: For details on comming events, please see the BRICS Activities web page: `www.brics.dk/Activities`.

Dissertation Abstracts

An Abstract Coalgebraic Approach to Process Equivalence for Well-Behaved Operational Semantics

Bartosz Klin

This thesis is part of the programme aimed at finding a mathematical theory of well-behaved structural operational semantics. General and basic results shown in 1997 in a seminal paper by Turi and Plotkin are extended in two directions, aiming at greater expressivity of the framework.

The so-called bialgebraic framework of Turi and Plotkin is an abstract generalization of the well-known structural operational semantics format GSOS, and provides a theory of operational semantic rules for which bisimulation equivalence is a congruence.

The first part of this thesis aims at extending that framework to cover other operational equivalences and preorders (e.g. trace equivalence), known collectively as the van Glabbeek spectrum. To do this, a novel coalgebraic approach to relations on processes is desirable, since the usual approach to coalgebraic bisimulations as spans of coalgebras does not extend easily to other known equivalences on processes. Such an approach, based on fibrations of test suites, is presented. Based on this, an abstract characterization of congruence formats is given, parametrized by the relation on processes that is expected to be compositional. This abstract characterization is then specialized to the case of trace equivalence, completed trace equivalence and failures equivalence. In the two latter cases, novel congruence formats are obtained, extending the current state of the art in this area of research.

The second part of the thesis aims at extending the bialgebraic framework to cover a general class of recursive language constructs, defined by (possibly unguarded) recursive equations. Since unguarded equations may be a source of divergence, the entire framework is interpreted in a suitable domain category, in-

stead of the category of sets and functions. It is shown that a class of recursive equations called regular equations can be merged seamlessly with GSOS operational rules, yielding well-behaved operational semantics for languages extended with recursive constructs. See [DS-04-1].

New in the BRICS Report Series, 2004

ISSN 0909-0878

13 Jens Groth and Gorm Salomonsen. *Strong Privacy Protection in Electronic Voting*. July 2004. 12 pp. Preliminary abstract presented at Tjoa and Wagner, editors, *13th International Workshop on Database and Expert Systems Applications*, DEXA '02 Proceedings, 2002, page 436.

12 Olivier Danvy and Ulrik P. Schultz. *Lambda-Lifting in Quadratic Time*. June 2004. 34 pp. To appear in *Journal of Functional and Logic Programming*. This report supersedes the earlier BRICS report RS-03-36 which was an extended version of a paper appearing in Hu and Rodríguez-Artalejo, editors, *Sixth International Symposium on Functional and Logic Programming*, FLOPS '02 Proceedings, LNCS 2441, 2002, pages 134–151.

11 Vladimiro Sassone and Paweł Sobociński. *Congruences for Contextual Graph-Rewriting*. June 2004. 29 pp.

10 Daniele Varacca, Hagen Völzer, and Glynn Winskel. *Probabilistic Event Structures and Domains*. June 2004. 41 pp. Extended version of an article to appear in Gardner and Yoshida, editors, *Concurrency Theory: 15th International Conference*, CONCUR '04 Proceedings, LNCS, 2004.

9 Ivan B. Damgård, Serge Fehr, and Louis Salvail. *Zero-Knowledge Proofs and String Commitments Withstanding Quantum Attacks*. May 2004. 22 pp.

8 Petr Jančar and Jiří Srba. *Highly Undecidable Questions for Process Algebras*. April 2004. 25 pp. To appear in Lévy, Mayr and Mitchell, editors, *3rd IFIP International Conference on Theoretical Computer Science*, TCS '04 Proceedings, 2004.

New in the BRICS Notes Series, 2004

ISSN 0909-3206

1 Luca Aceto, Willem Jan Fokkink, and Irek Ulidowski, editors. *Preliminary Proceedings of the Workshop on Structural Operational Semantics, SOS '04*, (London, United Kingdom, August 30, 2004), August 2003. vi+56.

New in the BRICS Dissertation Series, 2004

ISSN 1396-7002

1 Bartosz Klin. *An Abstract Coalgebraic Approach to Process Equivalence for Well-Behaved Operational Semantics*. May 2004. PhD thesis. x+152 pp.

http://www.win.tue.nl/ipa

INSTITUTE FOR PROGRAMMING RESEARCH AND ALGORITHMICS

Over the summer, IPA was extended with a new research group. This brings the total number of participating groups to 26, distributed over eight Dutch universities and the research institute CWI in Amsterdam.

Meanwhile, preparations are underway for the annual Herfstdagen, which will be dedicated to Intelligent Algorithms, and for the celebration of a milestone in the history of the IPA Dissertation Series.

For more IPA news we refer you to our web site.

New research group joins IPA

This summer, the research group Biomodeling and Informatics (BMI) of the Department of Biomedical Engineering of the Technische Universiteit Eindhoven joined IPA. The group, founded by Peter Hilbers in 2001, focusses on the modeling of processes in living systems. Besides the development of methods and tools for biomedical modeling, it is concerned with building specific biomedical models and implementing them by means of algorithms and simulations. IPA welcomes the BMI group, which will enrich the Institute's field of research.
See www.bmi2.bmt.tue.nl/Biomedinf/

100th dissertation in series

In 1996, IPA started a dissertation series in which theses of Ph.D. students in IPA are collected. On October 7, the series will reach its 100th dissertation, when Nicolae Goga defends his thesis 'Control and Selection Techniques for the Automated Testing of Reactive Systems' at the Department of Mathematics and Computer Science of the Technische Universiteit Eindhoven.

Coming events

IPA **Herfstdagen** on **Intelligent Algorithms**
November 22-26, 2004, Tulip Inn, Callantsoog, The Netherlands.
The Herfstdagen are an annual five day event, dedicated to one of IPA's current main application areas: Networked Embedded Systems, Security, Intelligent Algorithms, and Compositional Programming Methods.

Algorithms are vital building blocks for many software systems. The ever widening range of application for systems with algorithmic components in both industry and science (e.g. in ambient intelligence en bioinformatics) brings different requirements to the fore than those traditionally studied in algorithmics research. For instance, algorithmic systems can be required to be 'always on', to be aware of their (unpredictable) surroundings, or to adapt their behaviour to that of their users over time. The Herfstdagen aim to provide an overview of research in and around IPA on algorithms with these and other 'intelligent' properties. The program is composed by Emile Aarts (Technische Universiteit Eindhoven, Philips Research), Joost Kok (Leiden University), and Jan van Leeuwen (Utrecht University). More information will become available through the Herfstdagen webpage.
See: www.win.tue.nl/ipa/activities/falldays2004/

IPA sponsors **FMCO 2004**
November 2 - 5, 2004, Lorentz Center, Leiden University, The Netherlands.
The objective of this third international symposium on Formal Methods for Components and Objects is to bring together top researchers in the area of software engineering to discuss the state-of-the-art and future applications of formal methods in the development of large component-based and object oriented software systems. Key-note speakers are Robin Milner (Cambridge), Kim Bruce (Williams College), Tom Henzinger (Berkeley), Thomas Ball (Microsoft Research Redmond), Kim Larsen (Aalborg), Chris Hankin (Imperial College), Samson Abramsky (Oxford), and Reinhard Wilhelm (Saarland University).
See: http://fmco.liacs.nl/fmco04.html

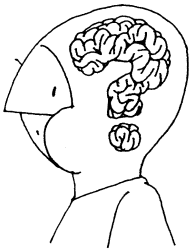
Addresses

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

Postal address
IPA, Fac. of Math. and Comp. Sci.
Technische Universiteit Eindhoven
P.O. Box 513
5600 MB Eindhoven
The Netherlands

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

EATCS News



News from Australia

BY

C.J. Fidge

School of Information Technology and Electrical Engineering
The University of Queensland, Australia
http://www.itee.uq.edu.au/~cjf

Regular readers of this column may recall the announcement of the National ICT Australia (NICTA) research organisation, which was established in 2002 through a $129.5 million commitment from the Australian government. The organisation was controversial from the start, with many people complaining that awarding the funding to a consortium consisting of the University of New South Wales and the Australian National University merely centralised more money in institutions that were already well funded.

Further discontent has arisen from the seeming slowness in getting the organisation off the ground. In May 2004 a further funding increase of $250 million was budgeted for NICTA by the federal government. However, at the same time, it was reported in the national media that questions were being asked during a Senate Estimates Committee about the lack of evident progress in the two years since NICTA's establishment and its apparent lack of accountability.

Nevertheless, two substantial developments have taken place since. Firstly, NICTA has identified two 'Priority Challenges' for itself:

From Data to Knowledge aims to produce social, environmental and economic value from the gathering and use of information. The issues here are data collection, data management, and data mining.

Trusted Wireless Networks aims to enable greater confidence, freedom and capability through improved efficiency, reliability and security of wireless en-

vironments. The concerns here are with the performance and trustworthiness of mobile computing devices.

Secondly, and perhaps more significantly, NICTA recently announced plans to establish two new research nodes, outside its home state of New South Wales.

- The Victorian node will be based at the University of Melbourne and will focus on networking technologies.
- The Queensland node will involve several Queensland-based universities and will focus on security issues.

Of the two, plans for the Victorian node are more advanced. Having been personally involved in the negotiations for the Queensland node of NICTA, I can confirm that its technical programme is still largely undefined.

Nevertheless, despite these teething troubles, NICTA remains the largest single commitment made by the Australian government to computer science research in Australia, and it is destined to dominate IT research in this country for many years to come. Readers interested in undertaking computing research in Australia should keep an eye on NICTA's web pages (`http://nicta.com.au/`), where research positions are advertised regularly.

News from India

BY

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

In this edition, we look ahead to some of the conferences coming up in India this winter.

FSTTCS 2004. The 24th annual conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS) will be held in Chennai (Madras) from December 16–18. The conference will be held at the Institute of Mathematical Sciences. This year's invited speakers are Javier Esparza, Piotr Indyk, Pavel Pevzner, John C. Reynolds and Denis Thérien. A total of 38 contributed papers have been selected for presentation.

In addition to invited talks and contributed papers, the FSTTCS 2004 programme will have two pre-conference workshops.

- 13–14 December: *Algorithms for dynamic data*, coordinated by S. Muthukrishnan and Pankaj Agarwal.
- 14–15 December: *Logic for dynamic data*, coordinated by Uday Reddy.

Look up `http://www.fsttcs.org` for up-to-date information about the conference, including registration details.

Indocrypt 2005. The 5th International Conference on Cryptology in India (Indocrypt 2005) will be held at the Institute of Mathematical Sciences, Chennai from December 20–22, 2004. The conference includes invited talks, tutorials and contributed papers. More details about Indocrypt 2005 can be found at `http://www-rocq.inria.fr/codes/indocrypt2004/cfp.html`

Indian Conference on Logic. The first Indian conference on Logic and its Relationship With Other Disciplines will be held from January 8–12, 2005 at the Indian Institute of Technology Bombay, Mumbai.

The conference will cover three basic themes: Indian systems of Logic, Systems of Formal Logic and Foundational issues in Philosophical Logic, Issues arising out of applications of Logic in the relevant disciplines.

Here are some of the invited speakers at the meeting:

- **Philosophical Logic** John Crossley, Yuri Gurevich, Petr Hajek, Wilfrid A. Hodges, Rohit Parikh, Krister Segerberg.
- **Indian systems of logic** S.M. Bhave, Pradeep Gokhale, D. Prahladacharya, K. Ramasubramanian, V.V.S. Sarma, M.D. Srinivas, S.P. Suresh

The conference web page is at `http://logic2005.hss.iitb.ac.in`

TECS Week 2005. The 3rd annual TCS Excellence in Computer Science (TECS) Week will be organized by the Tata Research Development and Design Centre at Pune from January 4–8 2005. The theme of this year's meeting is *Security Modeling*, with an emphasis on the formal modeling, analysis and verification of the security aspects of computer systems. The invited speakers at TECS Week 2005 are Butler Lampson (Microsoft Research), John Mitchell (Stanford) Xavier Leroy (INRIA) and Edward W. Felten (Princeton). For more details, look up `www.tcs-trddc.com/tecs`.

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

News from Ireland

BY

Anthony K. Seda

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

Quite a lot happened in Ireland of interest to the TCS community in the last couple of months. In particular, IJCAR'2004 took place in the National University of Ireland, Cork and MFCSIT'2004 took place in Trinity College Dublin, and I would like to report briefly on these two conferences.

IJCAR'2004, The Second International Joint Conference on Automated Reasoning, took place at NUI Cork from 4th July to 8th July, 2004 and was very ably organized, at the local level, by Toby Walsh and Barry O'Sullivan of the Cork Constraint Computation Centre (4C) and the Department of Computer Science, NUI, Cork. The invited talks were "Rewriting Logic Semantics: From Language Specifications to Formal Analysis Tools", by José Meseguer, "Second Order Logic over Finite Structures – Report on a Research Programme", by Georg Gottlob, and "Solving Constraints by Elimination Methods", by Volker Weispfenning, and I had the pleasure of listening to these and many other impressive lectures as a local participant. A full report of this conference will appear elsewhere. (The conference website is `http://www.4c.ucc.ie/ijcar/index.html` .)

The Third Irish Conference on the Mathematical Foundations of Computer Science and Information Technology (MFCSIT'2004), took place in Trinity College Dublin (TCD) on 22nd and 23rd July, 2004. Once again, we were fortunate in having invited speakers of the highest calibre who gave talks as follows: "Information is Physical, but Physics is Logical" by Samson Abramsky of Oxford University, "Mathematics for Software Engineers" by David Parnas of University of Limerick, "Topological analysis of refinement" by Michael Huth of Imperial College, and "Using Multi-Agent Systems to Represent Uncertainty" by Joseph

Halpern of Cornell University. Submitted talks were given by: Georg Essl of Media Lab Europe: “Computation of Wavefronts on a Disk I: Numerical Experiments”; John Power of LFCS, Edinburgh: “Discrete Lawvere Theories”; Colm O’hEigeartaigh and Mike Scott of Dublin City University: “A Comparison of Point Counting methods for Hyperelliptic Curves over Prime Fields and Fields of Characteristic 2”; Andrew Butterfield and Jim Woodcock of TCD and University of Kent: “Formal Models of CSP-Like hardware”; Sharon Flynn and Dick Hamlet of NUI Galway and Portland State University: “Composition of Imperfect Formal Specifications for Component-based Software”; Anthony Seda and Máire Lane of NUI, Cork and BCRI: “On the Integration of Connectionist and Logic-Based Systems”; Alessandra Di Pierro and Herbert Wiklicky of University of Pisa and Imperial College: “Operator Algebras and the Operational Semantics of Probabilistic Languages”; Michael B. Smyth and R Tsaur of Imperial College: “Convenient Categories of Geometric Graphs”; S. Romaguera, E.A. Sánchez-Pérez and O. Valero: “The dual complexity space as the dual of a normed cone”; Homeeira Pajoohesh and Michel Schellekens of NUI, Cork: “Binary trees equiped with semivaluations”; Xiang Feng and Michael B. Smyth: “Partial Matroid approach to Geometric Computations”; Micheal O’Heigheartaigh of NUI, Dublin: “r-Chains in Graphs: Applications in Counting Hamiltonian Tours”; and Paul Harrington, Chee K. Yap and Colm O Dunlaing of Trinity College, Dublin: “Efficient Voronoi diagram construction for convex sites in three dimensions”. The high quality of all these talks and the relaxed atmosphere in TCD ensured a scientifically valuable and enjoyable meeting.

The Conference proceedings will again appear as a volume in ENTCS, Elsevier’s series “Electronic Notes in Theoretical Computer Science”. More information can be found at `http://www.cs.tcd.ie/MFCSIT2004/` .

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 reports on the sixth **L**atin **A**merican **T**heoretical **In**formatics conference (LATIN 2004) by Joachim von zur Gathen and on the First Latin American conference on Combinatorics, Graphs, and Algorithms (LACGA 04) by Sebastián Ceria, and a reminder of the Call for papers of the second Brazilian Symposium on Graphs Algorithms and Combinatorics (GRACO 2005). 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.

Report on LATIN 2004 by Joachim von zur Gathen

The sixth of the **L**atin **A**merican **T**heoretical **In**formatics conference series was held from April 5 to 8, 2004, in Buenos Aires, the capital of Argentina. See `latin04.org` for details.

Imre Simon from the Universidade de São Paulo had founded this enterprise with its first meeting in 1992 in São Paulo, and continued to infuse it with his energy and ideas. His involvement and his sixtieth birthday were celebrated at the meeting, with two technical sessions dedicated to Imre, and a birthday festivity, complete with a huge birthday cake (adorned with self-lighting candles that Imre blew out—and then they lit up again ...) and short speeches by Ricardo Baeza-Yates, Volker Diekert, Marcos Kiwi, and Yoshiharu Kohayakawa.

The program committee, headed by Martín Farach-Colton, had selected for presentation 59 out of 178 submissions (acceptance rate = 33%). Among the conference highlights were the five invited talks by Cynthia Dwork (on fighting spam), Yoshiharu Kohayakawa (regularity method), Dexter Kozen (Kleene algebra and program analysis), and Jean-Eric Pin (Imre Simon's contributions to automata, languages, semigroups).

The proceedings, published as Springer Lecture Notes in Computer Science **2976**, were available at the meeting. We learned of the difficulties of getting such a shipment through Argentine customs.

Among the topics, algorithm design in its many guises (graphs, geometry, data streams, approximation and online, communication) was the most popular subject, with more than half of the talks. There were a number of presentations on computational complexity and on automata theory, and some on logic and on combinatorics.

The conference series is now well established and mature enough to have its by-laws. Suggestions had been prepared by Marcos Kiwi, Daniel Panario, Sergio Rajsbaum, and Alfredo Viola, were presented at the business meeting, and accepted by unanimous vote. The legality of such a vote is clearly unclear, and clearly nobody worried. The result constitutes an example of a successful transition from no authority to a legal government. (It was particularly easy because LATIN has no oil `;-)`.) The LATIN conferences will now be supervised by a steering committee, whose current members are: Ricardo Baeza-Yates, Martín Farach-Colton, Gastón Gonnet, Sergio Rajsbaum, and Gadiel Seroussi. See `www.latintcs.org` for details.

The meeting took place in downtown Buenos Aires, a vibrant city with many sights, from its colonial heritage to the waterfront and innumerable cafés and restaurants, inviting you to overdose on good coffee and excellent beef. The conference banquet was held in a beautifully situated restaurant, on the waterfront in an upscale part of the former port. The food was delicious, and the many people who had helped make the conference a success received their just awards: presents and plenty of applause.

Competitors for the 2006 venue are Cuzco and Rio de Janeiro. The success of the 2004 meeting bodes well for the future of this series.

Report on LACGA04 by Sebastián Ceria

The First Latin American conference on Combinatorics, Graphs, and Algorithms (LACGA 04) took place from August 16 to 20, 2004, at the Universdad de Chile, in Santiago, Chile (see `www.dii.uchile.cl/lacga04` for more details)

The program committee was headed by Prof. Thomas Liebling, from the Ecole Polytechnique Federale de Lausanne, Switzerland, and included academics and

practitioners from Argentina, Brazil, Chile, Mexico, Uruguay, USA, France, Italy, Switzerland, Germany, and Israel.

Topics of the Conference included combinatorial optimization and computational complexity; graph theory and matroids, and applications of Operations Research to fields as varied as Transportation, Forestry, Finance, Contract Allocation, and Sports Scheduling. There were a total of eighteen invited presentations, four applied talks, and forty contributed papers (whose extended abstracts will be published in the Electronic Notes in Discrete Mathematics, from Elsevier). The selection of contributed presentations followed a refereeing process by the Organizing Committee, and the selected papers will be published in a special issue of Discrete Applied Mathematics. The conference was hosted by the Department of Industrial Engineering and the Department of Mathematical Engineering of the Faculty of Physical and Mathematical Sciences of the Universdad de Chile, and sponsored by the Millenium Science Initiative and the Center of Mathematical Modeling, as well as the Ecole Polytechnique Federale de Lausanne.

Every day of the conference included plenary presentations in the morning, and contributed parallel sessions in the afternoon. Highlights of the conference included the plenary presentations by Prof. Gerard Cornuejols, from Carnegie Mellon University, on "Latest Advances in Integer Programming Theory and Practice"; by Prof. Andres Weintraub, from the Universidad de Chile, on "Applications of Operations Research to Forestry Planning Problems"; by Prof. Thomas Liebling, on "School Bus Routing"; by Prof. George Nemhauser, on "Sports Scheduling"; by Prof. Adrian Bondy, on "Ten Beautiful Conjectures in Graph Theory"; and by Jayme Szwarcfiter on "A Huffman-like Code with Error Detection Capability".

The meeting took place in the beautiful city of Santiago, Chile, at the modern auditorium of the Universidad de Chile. The conference included an afternoon trip to Isla Negra, to visit the house of the famous chilenean poet Pablo Neruda, and a Conference Banquet at the Santiago Park Plaza Hotel. We all enjoyed the great seafood that Chile has to offer, and the famous Pisco Sours, the landmark drink from Chile.

The organizers decided, given the great interest in the topics covered at the conference, the enthusiasm of the almost 100 participants, and the positive feedback received, that to organize LACGA II in 2007. Several participants offered to organize the next edition of the conference in either Rio de Janeiro, Brazil, or Rosario, Argentina.

GRACO 2005

GRACO 2005, the 2nd Brazilian Symposium on Graphs Algorithms and Combinatorics, will be held in Rio de Janeiro, Brazil, on April 27–29, 2005. The

deadline for the extended abstract submission is October 25, 2004. The Web page of the event is www.cos.ufrj.br/~celina/graco2005.

Regional Events

- September 20 - 21, 2004, Córdoba, Argentina: Argentine Symposium on Artificial Intelligence (www.exa.unicen.edu.ar/~asai2004/).
- September 20 - 24, 2004, Córdoba, Argentina: ASIS 2004 - Simposio Argentino de Sistemas de Información (www.cs.famaf.unc.edu.ar/33JAIIO/).
- September 20 - 24, 2004, Colima, Mexico: ENC'04 - Mexican International Conference in Computer Science 2004 (www.smcc.org//enc04/).
- September 27 - October 1, Arequipa, Peru: CLEI - XXX Latin American Conference in Informatics (www.spc.org.pe/clei2004/).
- April 27 - 29, 2005, Rio de Janeiro, Brazil: GRACO 2005 - II Brazilian Symposium on Graphs, Algorithms and Combinatorics (www.cos.ufrj.br/~celina/graco2005/).

News from New Zealand

BY

C.S. Calude

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

Recent conferences and workshops to be held in New Zealand:

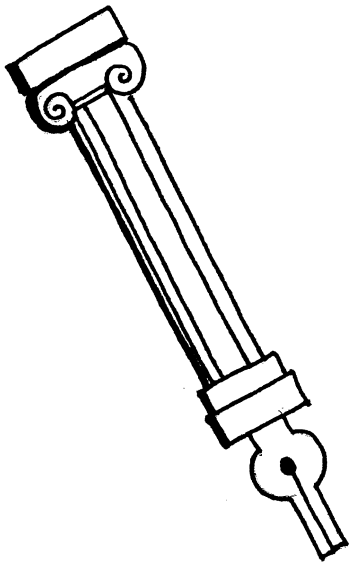
- The 10th "International Workshop on Combinatorial Image Analysis" will be held in Auckland on 1–3 December 2004, www.citr.auckland.ac.nz/~IWCIA04/.
- The Workshop on "Automata, Structures and Logic" will be held in Auckland on 11–13 December 2004, www.cs.auckland.ac.nz/wasl2004.
- The "International Workshop on Tilings and Cellular Automata" WTCA'04 will be held in Auckland on 12 December 2004, www.cs.auckland.ac.nz/dlt04/index.php?DOC=wtca/wtca.html.
- The Eighth International Conference "Developments in Language Theory" (DLT'04) will be held in Auckland on 13–17 December 2004, www.cs.auckland.ac.nz/CDMTCS/conferences/dlt04.
- The 2004 NZIMA Conference "Combinatorics and its Applications" and the 29th Australasian Conference "Combinatorial Mathematics and Combinatorial Computing" (29th ACCMCC) will be held in Lake Taupo, from 13 to 18 December, 2004, www.nzima.auckland.ac.nz/combinatorics/conference.html.

- The 2005 “Information Theory Workshop” (ITW2005) will be held in Rotorua from 29 August to 1 September 2005, under the banner of “Coding and Complexity", https://www.cs.auckland.ac.nz/itw2005.

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

241. C.S. Calude and H. Jürgensen. Is Complexity a Source of Incompleteness? 06/2004
242. A. Juarna and V. Vajnovszki. Fast Generation of Fibonacci Permutations. 07/2004
243. M. Stay. Inexpensive Linear-Optical Implementations of Deutsch’s Algorithm. 07/2004
244. D. Schultes. Rainbow Sort: Sorting at the Speed of Light. 07/2004
245. M. Harmer. Fitting Parameters for a Solvable Model of a Quantum Network. 07/2004
246. C.S. Calude and G. Păun. *Computing with Cells and Atoms*: After Five Years. 08/2004
247. C. Grozea. Plagiarism Detection with State of the Art Compression Programs. 08/2004

The EATCS Columns



The Algorithmics Column

by

Josep Diaz

Department of Languages and Computer Systems
Polytechnical University of Catalunya
c/ Jodi Girona 1-3, 080304 Barcelona, Spain
diaz@lsi.upc.es

In this issue of the bulletin, the AMORE people at the ETH in Zurich present an interesting overview of some algorithmic issues in the field of *Railway Optimization Problems*.

Theory on the Tracks: A Selection of Railway Optimization Problems

Michael Gatto, Riko Jacob, Leon Peeters,
Birgitta Weber, and Peter Widmayer *

Abstract

Railway optimization problems have been studied from a mathematical programming perspective for decades. This approach is particularly well suited to model the multitude of constraints that typically arises in the context of a railway system. In recent years, these problems have attracted some attention in the algorithms community, with a focus on individual problem aspects, in an attempt to understand precisely where the inherent complexity of railway optimization lies. We sketch the state of affairs by discussing a few selected examples at different levels of detail.

*Institute of Theoretical Computer Science, ETH Zurich, gattom,rjacob,leon.peeters, weberb,widmayer@inf.ethz.ch

1 Introduction

Imagine an ideal railway system, perfectly optimized in every aspect, fully adaptive to travellers' needs and the dynamic change of circumstances, with convenient and reliable connections to all destinations at all times. Obviously, we can approach this ideal only with computer assistance or even full computer control. Such an ideal would be desirable not only to provide good service for the individual commuter and traveler, but also to make best possible use of scarce resources that a modern society should not afford to waste. Now, put this in contrast with the reality of railway optimization, where timetables manually undergo incremental changes once a year, and where delays of trains and other distortions are managed by groups of experienced people in front of large walls full of computer screens. What is the reason behind this automation gap? In our view, the main reason is the complexity of the planning and the operational problems. These problems are far more demanding than similar problems for, say, airlines, due to the size of problem instances and the multitude of constraints to be satisfied. This multitude of aspects makes railway optimization also more complex than optimization of most other large scale technical systems, such as telecommunication networks or computer chips, where events happen in a much more orderly fashion, with less or no influence from physical reality and involved people. Our goal in the following is to support this claim with a few examples for individual railway optimization problems. We do so from an algorithmic complexity perspective, in an attempt to understand what ingredients make a problem hard, and what might be done to cope with hardness. It is only after a fundamental algorithmic understanding of the individual components of railway optimization that we can hope for the ideal railway system.

The operations research community has studied railway optimization for decades very successfully. The literature on operations research and on transportation science abounds with interesting studies, with a nice survey [9] making for a very good start. Most realistic railway optimization problems, defined by a complicated set of constraints and a variety of competing objectives, are NP-hard. A wealth of powerful generic techniques have been proposed to attack them, such as branch and bound, branch and cut, mixed integer linear programming, Lagrangean relaxation, in addition to an abundance of specific heuristics. These methods tend to deliver good solutions for modest problem instance sizes, mostly after running for a long time. Our approach is different: We are not confident that all constraints can be modelled truthfully in a single shot, and we therefore advocate an iterative, interactive approach in which the railway planner asks the system for a solution to a problem that she specified in a rather simple setting. Then, she evaluates the proposed solution, perhaps agreeing with some part of the solution, while modifying some of the problem parameters for the next iteration. For this approach

to be feasible, the individual iterations of this process need to be very fast, and an overall optimum solution can certainly not be guaranteed. On the other hand, such a rapid response system would allow the strategic planner to experiment by asking "what if" questions, a prerequisite in our view for long term planning of change.

In the following, we will introduce a set of railway planning tasks, defining optimization problems that we then consider in more detail. In Section 2, we discuss the problem of how to react to delays of trains. Section 3 treats the problem of assigning physical trains to rides according to schedule, and Section 4 addresses the potential that changes in the timetable may have for saving on rolling stock. Section 5 discusses the issue of building extra stations along existing tracks. Finally, Section 6 takes an algorithmic look at the representation of timetables for querying.

The topics we picked to illustrate railway optimization are highly personal, but (as we feel) fairly representative of the field. Our point of view has been shaped by the interaction with experts in railways and in optimization algorithms, predominantly in a joint European project on railway optimization (AMORE, HPRN-CT-1999-00104) that Dorothea Wagner initiated and guided throughout. We learnt a lot from our project partners Dines Bjorner, Gabriele di Stefano, Bert Gerards, Leo Kroon, Alexander Schrijver, Alberto Marchetti-Spaccamela, and Christos Zaroliagis. In the Computer Science Department at ETH Zürich, we had the benefit over the years of working with Luzi Anderegg, Mark Cieliebak, Stephan Eidenbenz, Thomas Erlebach, Martin Gantenbein, Björn Glaus, Fabian Hennecke, Sonia Mansilla, Gabriele Neyer, Marc Nunkesser, Aris Pagourtzis, Paolo Penna, Konrad Schlude, Anita Schöbel, Christoph Stamm, Kathleen Steinhöfel, and David Scot Taylor. Railway experts that helped us understand the problems and provided us with data include Daniel Hürlimann from ETH Zürich, as well as Jean-Claude Strüby and Peter Hostettler from the Swiss Federal Railways SBB, and Frank Wagner and Martin Neuser from Deutsche Bahn.

Planning and operations in railways

This section briefly describes the major planning and operational problems in a railway system, so as to sketch the origins of the railway optimization problems, and the interactions between them (see also the excellent review by Bussieck, Winter, and Zimmermann [9]).

Figure 1 depicts the usual planning processes for a railway operator, and the time dependencies between them. The figure contains a mix of strategic planning problems, such as demand estimation and line planning, and operational planning problems, such as rolling stock scheduling and crew scheduling. For the latter two, it is essential to construct a strategic plan with respect to capacities, since

acquiring new rolling stock and hiring or re-training crews usually takes quite some time.

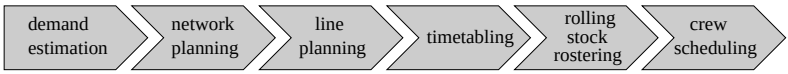


Figure 1: A railway operator's planning processes

Demand estimation. The estimation of the demand for railway services lies at the basis of a railway system. Travel demand is estimated as the number of people that wish to travel from an origin to a destination. By estimating the travel demand for each possible Origin-Destination combination, a so-called *OD-matrix* of the total travel demand is obtained. Passenger counts, passenger interviews, and ticket sales form the basic information for constructing the OD-matrix. Advanced statistical models are used to estimate the travel demand between geographical zones, and usually split that demand on factors such as day of the week, time of the day, mode of transportation, and travel motivation. Usually, this estimation is carried out on a yearly basis, but re-estimation during the year is possible in case of significant changes in mobility demand. Additionally, it might be useful to model the influence of the quality of service on the demand, i.e., a very good connection will in general attract passengers from other modes of transportation, or even create travel demand.

Network planning. At the heart of the railway infrastructure are the tracks and stations. It usually takes considerable amounts of money and time to build new tracks and stations, partly due to the time consuming political processes. Hence, the planning of infrastructure has a long term perspective. Some aspects of such planning can be addressed by optimization methods, as for example for locating new stations. Given the travel demand at its various origins, the corresponding travelers will only consider using rail transportation when a station is located in their vicinity. Here, the concept of vicinity depends on the individual mode of feeder transport, for example, car, bus, bicycle, or foot. Station location considers the travelers that do not fall into some vicinity radius of an existing station. For such a traveler, rail transportation can become a viable mode of transport by opening a new station in his or her vicinity. In the station location phase, a trade-off is made between the benefit of attracting potential customers, and the costs of constructing and operating the required new stations. Additionally, each stop at a station effectively slows down the servicing train, such that we have to trade off the distance of the travelers to the station with the average speed of the train.

Line planning. After the OD-matrix has been estimated and the network is determined, one proceeds with deciding which recurring trains will be part of the railway system. The set of operated trains is called the train line system. A *train line* is a train connection between an origin station and a destination station, including some intermediate stops. In practice, it proves to be most convenient to run the trains on a periodic schedule, where each train line is assigned a frequency and a type, the latter determining the stations that the line calls at. The most common line types are the intercity type, which only calls at the major stations, and the local type, which stops at every station it passes. Usually, at least one intermediate line type between these two extremes is used, such as a regional type. The line planning problem considers how to cover the railway network with lines, such that all traffic demand can be satisfied, while meeting certain objectives. Common criteria are maximizing the number of direct travelers, and minimizing the costs of the railway system.

Timetabling. With the constructed line plan as input, a *timetable* for its train lines can be constructed. In the timetable we prescribe the particular *rides* of the trains, i.e., the precise times when a certain *train* serves a line. Such a timetable has to meet various requirements. Railway safety regulations enforce any two trains using the same track to be separated by a minimum headway time. And clearly, the meeting and overtaking of trains on the same track are impossible. Also, bounds are specified for the dwell times at stations, in order to give passengers enough time to alight and board, and to control the trains' total travel time. Whenever there is no direct train between two stations, a scheduled transfer guarantees that a train heading for the destination station departs from a transfer station shortly after a train originating from the origin station has arrived. In this way, the timetable still offers a good travel scheme between the two stations. A timetable has to meet these and some other requirements, regarding turn-around times, a priori fixed departure times, and synchronizations between train lines. Simultaneously, a timetable objective function should be optimized, such as short travel times, high robustness with respect to small disruptions, and low associated operating costs, that are mainly influenced by the required number of trains.

Rolling stock rostering. The next planning problem to be solved is the assignment of train units to the train lines in the timetable. A railway company typically owns a variety of rolling stock types, such as single deck and double deck units, wagons that need a locomotive, or units that have their own engine. When each train line has been assigned one or more types of rolling stock, a plan is constructed specifying how many units each train consists of. Moreover, a unit does not have to be assigned to the same train line for the entire day. A unit may be

used for several train lines during the day, and it might even be separated from one train, wait for some time at a station, and then get attached to another train. During off-peak hours, not all rolling stock is in use, and the idle train units need to be shunted from the platforms to shunting yards. For this purpose, a shunting plan is constructed in a later planning phase. Also, each train unit needs to be taken out of circulation after having traveled a certain distance, in order to be serviced. Finally we need to adjust the plans of the flow of train units such that each unit is routed to a workshop when it requires maintenance.

Crew scheduling. Each train has to be manned by a driver and one or more conductors. This poses complex planning problems, since the crew plan needs to respect some side constraints, and several usually complicated labor rules. In general, the drivers and conductors have to return to their home bases by the end of the working day. Working shifts have to contain a meal break of at least half an hour, and may not contain a continuous period of work of more than five hours. Drivers and conductors are allowed to transfer from one train to another only when sufficient buffer time is available for the transfer. A delayed arrival of the crew could otherwise result in a delayed departure of their next train. More complicated rules also exist, for example that a shift must contain some variation, so a driver or conductor may not be assigned to the same train line all day, going back-and-forth on that line. Furthermore, the crew schedule should incorporate various crew member characteristics, such as rostering qualifications, individual requests of crew members, and the past rosters of crew members. These past rosters are of importance for the labor rules. Finally, taking all these rules and characteristics into account, a railway operator aims at constructing a crew plan that optimizes certain objectives, such as minimizing the number of persons required to cover all the work, or maximizing crew satisfaction by honoring the individual requests.

The flow of the overall planning process as shown in Figure 1 arises because certain processes provide the input for others. However, the planning flow is not as linear as the traditional separation into subproblems suggests, since most of the planning stages influence each other. For example, the timetable may be changed to improve the rolling stock circulation or the crew schedules. Actually, all these planning phases constitute one big optimization problem. It is beyond our current technology to optimize this complicated problem for real life instances. Hence, we stick to the traditional separation of problems and try to develop fast methods to optimize a single stage, and use some feedback mechanisms to get a reasonable method for the complete planning chain.

Apart from the estimation of demand, most of the above described problems

also play at the level of operations. For example, the timetable, rolling stock schedule, or crew schedule may need to be adjusted during operations due to last-minute disturbances. Such disturbances can have a wide range of causes, from broken carriages and crew member illness to accidents on the tracks. But, as an overall plan has already been constructed, and should be adhered to as much as possible, these operational problems are inherently of a different nature. Also, some problems at the operational level do not have a planning equivalent, such as the fast answering of timetable queries in an automated system. Below, we describe two operational problems that have received quite some attention lately.

Delay management. Even with the careful planning of the line plan, timetable, rolling stock, and crews described above, some unavoidable delays may occur during operation. In such a case, the discomfort a customer faces can be reduced by maintaining some of the future connections of his trip. When maintaining the connections for the passengers in a delayed train, the connecting trains are ordered to wait, and thus to deviate from their timetable. Hence, the connecting trains deliberately depart with a delay, and the passengers already inside such a connecting train may face an arrival delay at their destination. Moreover, a deliberately delayed connecting train may propagate the original delay to subsequent stations. Therefore, these decisions must be taken with great care. One version of the problem is concerned with a snap-shot of the railway system with some delayed trains. The task is to minimize the negative impact on customer traveling times by deciding which connecting trains should wait for a delayed feeder train, and which ones should depart as scheduled.

Timetable queries. Nowadays, most passengers use electronic timetable information systems to plan their railway trips, rather than the old fashioned printed timetable booklet. Especially web based timetable information systems have the advantage of being flexible and containing up-to-date information. For example, when a track is temporarily closed due to maintenance, the information system readily returns the best de-tour around that track. Further, an electronic timetable information system can be individually tailored, with respect to preferred type of train, minimum transfer times, required intermediate stops, and many more. Fast and precise query results are crucial for such systems, as they are the main user criteria. However, returning precise results in little time is not easy, especially if the system also contains information on bus, metro, and tram for pre- and post-transport, and if the system considers international connections as well.

2 Delay management

Delay management is concerned with responding to unexpected delays of trains. If trains delay, some passengers traveling on the delayed train might miss a connecting on-time train if the operation continues as planned. One option for guaranteeing such connections is to let the on-time train wait for its delayed connecting passengers (we ignore other options for the time being). Delay management considers the aspect of determining which trains should wait for which connecting (incoming) trains, and which trains should depart on time. The problem can be illustrated by an example. Consider a passenger with destination Moscow in a train from Berlin to Warsaw, and assume that this train waits for 10 minutes in Berlin to guarantee a connection from Paris. Then all passengers on the train will reach Warsaw with a delay of 10 minutes. Hence, our passenger from Berlin might miss in Warsaw her connection to Moscow. So in Warsaw we face the decision to make the train to Moscow wait. Perhaps, it does hence not pay off to have the train wait in Berlin in the first place, and thus delay a lot of passengers by 10 minutes, instead of making only a couple of passengers wait in Berlin for the next train to Warsaw. One reasonable criterion upon which to judge such a trade-off is to minimize the sum of all passenger delays. The delay management problem has been analyzed for the last 20 years with a twofold focus. On the one hand, delay management is a practical problem. Hence, a solution is of any practical use only if it satisfies all side constraints of the real world. In order to find such a solution, one has to model the problem with as much detail as possible. This aspect of delay management has been addressed through simulations [1, 30, 23]. On the other hand, delay management is also a challenging theoretical problem. Results from theory focus on simplified models of delay management. In an attempt to understand the underlying algorithmic complexity, we consider the problem of minimizing the total passenger delay, under the very special assumption that all initial delays of trains are part of the input.

In the following subsections, we present the results obtained with different approaches. We start discussing the integer linear programming formulation of delay management on event activity networks. Then we describe efficient algorithmic approaches on a simple network model, as well as the computational complexity of delay management. After discussing this off-line setting of delay management, where all delays are given as part of the input, we give a short overview of a simulation approach with stochastic delay appearance, as well as an outline of competitive on-line algorithms for a simple network topology.

Integer linear programming approaches

A prominent approach taken in the literature is to formulate this problem as an integer linear program (ILP). The resulting models include a very detailed description of the delay management problem (trains require a certain travel time between stations, trains must wait at stations for certain times, and many more).

In one approach, an ILP maximizes the number of passengers transported, given that some trains can be canceled and additional trains can be scheduled [1]. A different ILP formulation [39] supports different weights for waiting times, allowing to model the fact that waiting on a cold and windy platform is worse than waiting in a train. The model only allows trains to wait for the delayed connections, or to depart as scheduled, and the ILP minimizes the total passenger delay. Unfortunately, these models do not provide enough structure to be solved by other means than general ILP solvers. The great number of involved variables and constraints make the ILPs difficult to be applied to real-life instances.

In an alternative approach, the delay management problem is modeled on an event-activity network [29], a commonly used graph representation of railway systems: the nodes in the graph represent arrival and departure events of trains at and from stations; the directed edges represent driving activities of trains between stations, waiting activities of trains at stations and transfer activities of passengers between different trains within a station. An example of an event-activity network is shown in Figure 2. In general, nodes and edges have associated data. Arrival

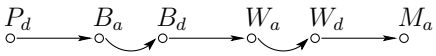


Figure 2: An event activity network for the Paris-Moscow example. Straight edges represent driving trains, rounded edges changing passengers. As we consider three different trains, there are no waiting activities. Nodes are labeled with the station's name, the subscript distinguishes arrival from departure events. Passenger flows are not shown.

and departure nodes have the scheduled time of the event, while each activity has a scheduled duration and a minimum requested time for performing the activity. Note that a potential time difference between scheduled duration and minimum requested times is the amount of time that can be saved in case of a delay (by speeding up while driving, for instance), the *slack time*. Passengers can travel from any departure event to any arrival event if both are connected by a directed path. The number of passengers traveling between any origin-destination pair along a specified route is given as part of the input; we call these *passenger paths*. Indeed, such data is frequently collected by the railway companies.

Assume for the moment that the connections between trains that have to be guaranteed are fixed a priori and given as part of the input. Then, the problem

of minimizing the total passenger delay can be easily modeled as an ILP. For a guaranteed connection from train *a* to train *b*, a constraint enforces that the departure time of train *b* at the connecting station is later than *a*'s arrival time plus the minimum required time for connecting. Similarly, other constraints enforce the minimum requested driving times and the minimum requested waiting times at stations. Finally, a family of constraints enforces that the events cannot take place earlier than scheduled. The derived constraint matrix can be shown to be totally unimodular, thus the LP relaxation of the problem always delivers an integral solution [34]. Hence, the problem of minimizing the total passenger delay under the assumption of guaranteed connections can be solved in polynomial time. This problem can also be solved by a combinatorial approach [34].

In general, however, the problem is to decide which connections should be guaranteed. As a first step, consider a special case. First, we assume that the timetable is *periodic*, with a period of T time units for all train lines. Further, we assume that delays do not propagate to the next period of the timetable; we call this *no inter-cycle delay propagation*. Hence, missing a connection causes a fixed delay of T time units, the same for each connection. With these two assumptions, the delay management problem can be modeled as an ILP [33]. The ILP uses three classes of variables: one class represents the delay of departure events, the second class the delay of arrival events. These two classes are real-valued variables. Finally, a class of Boolean variables models whether or not passenger paths miss connections. For a connection from train *a* to train *b*, a passenger path *misses* this connection if *b*'s departure delay is smaller than *a*'s arrival delay, corrected by an available slack time.

The difficulty of deciding whether to wait or not for some delayed passengers is a consequence of the fact that delayed passengers and trains may meet later somewhere in the network. This possibility makes the ILP formulation more complex. If we assume that in an optimal solution no two delayed vehicles (or vehicle and passenger) meet at any node (the "never-meet-property" in [34]), then some integer programming formulations can be partly relaxed to linearity and still provide an optimal solution. The constraints of the ILP are still similar to the ones sketched before.

To analyze the delay management problem further, we introduce the *constant delay assumption*: whenever the delay of a train is non-zero, it has a fixed size δ. Further, we assume that there are *no slack times* in the network. The resulting model appears to be the simplest possible that still captures the structure of delay management. We hope to gain some understanding by analyzing it in some detail (see the next section). For this restricted model, that is, with the constant delay assumption, no slack times, with a periodic timetable, no inter-cycle propagation and the "never-meet-property", the constraint matrix of the relaxed integer programming formulation is totally unimodular, thus the problem of minimizing the

total passenger delay is solvable in polynomial time. Thanks to these properties, it is possible to derive an appropriate branch-and-bound algorithm for solving the delay management problem with the sole assumptions of a periodic timetable and no inter-cycle delay propagation [34].

Combinatorial algorithms and computational complexity

Let us now turn to the simple version of the event-activity network described in the following.

The basic model. The railway network is modeled as a directed acyclic graph. Each node represents a station in the network at a particular time, so that a station can be associated with more than one node. Each edge represents a specific train traveling non-stop between two stations. Since a passenger cannot be at the same station at the same time twice in a row, the graph is acyclic. Passenger flows are weighted paths in the network, where each weight represents the number of passengers on the path. Differently from the earlier version of the event-activity network, delays are defined on the passenger paths. Initially, each path is either on-time, or has a given delay. We refer to this initial delay as *source delay*. In the off-line setting, path delays correspond to taking a snapshot of the network situation, and checking which passengers are influenced by the currently delayed trains. We analyze the model with the constant delay assumption, no slack times on train travelling times, with a periodic timetable of period T and no inter-cycle delay propagation. This appears to be the simplest model exhibiting a non-trivial wait / non-wait decision.

Considering these restrictions, a path with source delay δ does not miss any connection only if all the trains it uses wait for it. Should one of those trains not wait, the passenger path misses the corresponding connection and arrives at its destination with a delay of T time units. Similarly, a path with no source delay arrives at its destination on-time only if all its used trains depart on-time. On-time paths can, on the other hand, be delayed if some of the trains along their path wait for some other delayed path, and the delay propagates. Depending on the delay configuration of the trains in the path, the arrival delay of the passenger path is either of size δ or, if a subsequent connection is missed somewhere, of size T.

An example in Figure 3 explains the model, with the situation described at the beginning of Section 2.

The objective functions. For this basic model, the objective is to minimize the total passenger delay, defined as the sum of all passengers' arrival delays. This objective also includes the constant offset delay of source delayed paths. In contrast,

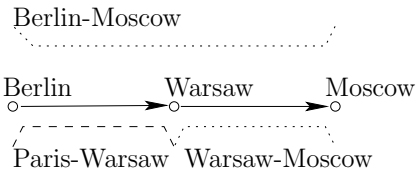


Figure 3: The Paris-Moscow problem as a snapshot of the situation in Berlin. Directed edges represent direct trains. The dashed line represents the source delayed Paris-Warsaw path, the dotted lines the on-time paths.

it would also be reasonable to account only for the additional delay, not counting the (unavoidable) constant offset in the problem input, because this is the only delay we can optimize. The optimization problem is the same in both cases, but for approximate solutions, the approximation ratio would be drastically different.

Computational complexity. The basic model teaches us about the combinatorial structure of delay management. It allows to assess the computational complexity of the basic problem, and of the problem with specific properties of the instances.

A solution to the delay management problem above consists of a partition of the trains into two sets, namely the set of trains which wait, and the set of trains that depart as scheduled. For a special case, assume that passengers change train at most twice. Then, this partition can be computed efficiently for arbitrary graphs, by reducing the delay management problem to a minimum directed cut computation [18]. In more detail, the trains are mapped to vertices in a graph G_c, and two special vertices s and t are added. For an illustration, see Figure 4. A directed $s-t$-cut [42, p. 178] in G_c splits the vertices into two sets $[S, \overline{S}]$, with $s \in S, t \in \overline{S}$. By choosing the weights of the edges in the graph appropriately, the size of the cut can be made equal to the total delay on the network induced by letting all trains in S wait, and all trains in $\overline{S}$ depart on time. As an example, illustrated in Figure 4, consider a passenger path of weight p with no source delay using, in sequence, the two trains a and b. During his journey, the passenger needs to connect from a to b. In G_c we add the three edges $(a,t),(a,b)$ and (b,a) having weight $w(a,t) = p \cdot \delta, w(a,b) = (T-\delta) \cdot p$ and $w(b,a) = p \cdot \delta$, respectively. If both trains depart on-time, the path arrives at its destination on-time. Correspondingly, if $\{a,b\} \subseteq \overline{S}$, no edge traverses the cut. If both trains wait, the path arrives with an arrival delay of δ time units, thus contributing to the objective with weight $p \cdot \delta$. Correspondingly, if $\{a,b\} \subseteq S$, the edge (a,t) traverses the cut, contributing with weight $p \cdot \delta$ to the cut's size. Similarly, this construction also handles correctly the last two possible waiting policies, see Figure 4. A similar construction takes

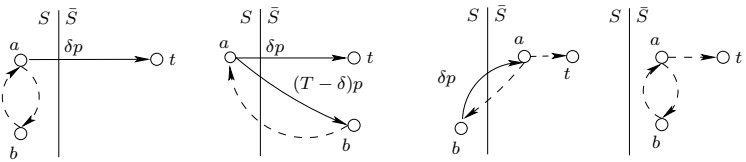


Figure 4: The four different delay policies for the two trains, with the corresponding costs of the cut.

care of paths with source delay. Further, the construction can be extended for paths connecting twice. Actually, the model used in [18] is slightly different from the one described above: instead of considering initial delays on paths, there is a single initial delay defined on one train. This delayed train induces a delay on all paths which use it. This is what happens at the initial example: the delay of the Paris-Berlin train induces a delay on the passenger path Paris-Warsaw. The reduction in [18] can nevertheless easily be modified to suit the basic model described above.

Consider the decision version of the delay management problem above, that asks for the existence of a delay policy with total passenger delay at most a given value. If we assume that some passengers are allowed to change train three times, the problem is strongly NP-complete [19]. The reduction is from independent set on the graph class 2-subdivision; this answers a long open question in the railway optimization community. If we wish to optimize the additional delay only, a different reduction from independent set, which does not constrain the number of train changes, leads to an inapproximability result of $\frac{15}{14}$.

The NP-hardness result applies to arbitrary network topologies. For another special case, consider the basic model allowing an arbitrary number of train changes, but restricting the network topology to be a simple path. Then, the delay management problem can be efficiently solved by dynamic programming [18]. The key consideration for the dynamic program is that given two subsequent trains, the first of which waits and the second departs on-time, all passenger paths carried by the first train will not reach the latter train. Hence, the optimal solution for the subsequent trains adds a constant offset for all solutions having, in the prefix, this wait / non wait transition. Consider, for example, the two partial solutions of Figure 5: on the left hand side, the first two trains wait, the third one departs on time, while the last train is yet to be decided. On the right hand side of the figure, the first and the third train depart on time, while the second train waits, and the last train is yet to be decided. Although these two partial solutions are different, it is clear that in both complete solutions the fourth train behaves the same way. In fact, exactly the same delay situation arises for that train in both

Figure 5: The two example configurations for the dynamic program.

partial solutions, since no passengers from the first and second train can get on the fourth train. Indeed, the third train departs as scheduled, thus every passenger path coming from the first and second train misses that connection. Algorithmically, it is hence sufficient to consider the best prefix solution having this wait / non wait transition, and with it explore the remaining solution space of the suffix. Thanks to this consideration, the dynamic program only explores $O(m^3)$ solutions in total, with m the number of trains on the line. This dynamic program can be extended such that it handles two specific classes of trees. It can be applied to in-trees, that is, directed rooted trees where all edges point towards the root node. The dynamic program applies the same idea as above, and starts from a leaf node. Similarly, the dynamic program can be extended to out trees, that is, directed rooted trees where all edges point away from the root.

Slack times. Naturally, every reasonable timetable includes some slack time, in order to limit the propagation of small delays. Hence, we can extend the basic model such that a train can catch up a predefined amount of time in case of delays.

The question whether slack times in the train's traveling times make the problem combinatorially more challenging is answered in [19]: the problem is NP-complete as soon as some passengers change train twice. Furthermore, solving the basic model with slack times on a graph with the topology of a path is NP-complete. These hardness results are achieved by reduction from directed acyclic maximum cut, which is also shown to be NP-complete. Note that the same problems without slack times are polynomial-time solvable. To complete the spectrum, the problem with slack times can be solved efficiently on arbitrary graphs if passengers change train at most once, by a reduction to a minimum directed cut [19].

Bicriterial delay management. The objective of causing the least overall delay to passengers may, under certain circumstances, propagate the delays considerably in the network. Delay propagation is usually not convenient from an operational point of view, and it is desirable to at least return to the original timetable as quickly as possible. Nevertheless, the operational objective of minimizing the timetable perturbation should not cause a too big discomfort to passengers. This is a *bicriterial delay management* problem: the first objective is to minimize the sum of the arrival delays of the trains, thus minimizing the perturbation with respect to the timetable. The second objective is to minimize the weighted number of missed connections, as an effort to prevent passengers from having big arrival delays. The

corresponding decision problem (with bounds for both criteria) can be shown to be weakly NP-complete [21], by reduction from the knapsack problem.

Real-time settings

All the previously described work considers an off-line setting of delay management. In all optimization models the initially delayed trains are assumed to be known a priori. On the other hand, delay management is inherently an on-line problem, since the delays appear unexpectedly over time.

In the series of papers [39, 40, 6], the authors address the question of which connections should be maintained, by applying deterministic decision policies. For example, one of the policies stems from the decision policy used by Deutsche Bahn, another one compares the number of connecting passengers with the number of passengers that will be immediately delayed by the waiting decision. These policies are then tested in an agent-based simulation tool on real-world data of Deutsche Bahn. The simulation introduces delays on the trains randomly over time with an exponential distribution. The results of the simulation allow to draw qualitative conclusions on the different waiting policies. For instance, the simulations show that both, maintaining all connections and not maintaining any connection, cause a larger total passenger delay than the decision policy of Deutsche Bahn. The latter can be outperformed using more complex decision policies which take into account passenger information.

The first attempts to analyze the on-line version of delay management through competitive analysis focus on a very simple setting. Consider a single railway line, performing intermediate stops. At each station, a constant number of passengers are willing to board the train. These passengers can either be on-time, or be delayed by an overall constant time. If a passenger misses the connection, she will incur T time units of delay, i.e. the time until the next train travels on the line. Since we do not allow for slack times, the goal is to decide at which station the train waits, and thus starts catching delayed passengers. It can be shown that in this model, no on-line algorithm can have a reasonably bounded competitiveness if the objective is to minimize the additional delay. If we optimize the total passenger delay on the train line, the problem allows a family of 2-competitive algorithms, and it can be shown that no on-line algorithm can be more than golden-ratio competitive. Further, the problem is related to a generalization of the ski-rental problem [20].

Concluding remarks

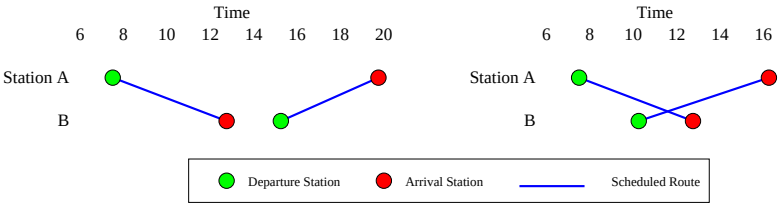
The methods described above show the twofold focus addressed at the beginning. Some models, as the ILPs and the simulations, have the goal of describing the

real-world problem with as much detail as possible. As a result it seems hard to gain a general understanding. Further, these models are normally so complex that they do not allow, with a few exceptions, to tackle problems of real-life size. On the other hand, the simpler models do provide insight on the complexity of the problem. However, the settings that showed to be solvable in polynomial time cover only a fraction of the detail of real-life problems. At present, it is still unclear how far these approaches can be extended, in order to model more complex scenarios. One of the challenges in delay management is, hence, to narrow the gap between practical relevance and theoretical understanding.

3 Rolling stock rostering

Introduction. *Rolling stock rostering* addresses the problem of assigning (physical) trains to the train routes prescribed by a given schedule. The train assignment needs to satisfy a variety of constraints, and it aims at minimizing its associated cost, such as the number of needed trains or the number of needed cars. This problem is also known as *train assignment*, *train rostering*, or *vehicle scheduling*. Today, train companies assign trains manually, with mostly incremental changes from one year's schedule to the next. The reason for this lack of automation are perhaps the modeling complications: it seems almost impossible to formulate all the constraints rigorously. As an alternative, one might wish to use an interactive optimization tool that lets a railway planner formulate an initial set of constraints and which returns an initial solution that she can inspect, modify, or partially keep as a basis for a subsequent iteration. The quest for understanding the problem as a whole has led to studies of simple problem variants, with the goal of identifying what can and what cannot be computed efficiently. This section sketches some of the most basic problem versions and their complexities; refer to [16] for more details.

An example. For the sake of concreteness, let each train ride be given by the station and the time of its departure and of its arrival, and assume the schedule to be periodic. Figure 6.a, taken from [16], shows two rides and two stations: the horizontal axis represents time and the vertical axis represents the stations, with an edge between two points representing a ride. For this example, one train traveling from A to B and then from B to A is sufficient to cover both rides each day. In contrast, two trains are needed for the example in Figure 6.b, because the train from A arrives too late in B to perform the ride from B to A. Hence, one train stays in A overnight and travels back to B the next day, and similarly one train stays in B and travels back to A. That is, a train comes back to its home location after two days, and therefore two trains are needed in this example.



(a) An instance requiring only one train. (b) An instance requiring two trains.

Figure 6: Two simple train schedules

Cycles. In general, a train assignment is represented by a set of cycles, where a link from an arrival node in a station to a departure node at the same station is a *wait during the day* if the arrival is before the departure, and is an *overnight wait* otherwise. The *length* of a cycle is an integer number of days, and it defines the number of trains needed to perform the schedule: in a k day cycle, k trains are needed to serve the rides in the cycle (see Figure 7, taken from [16], for an instructive instance).

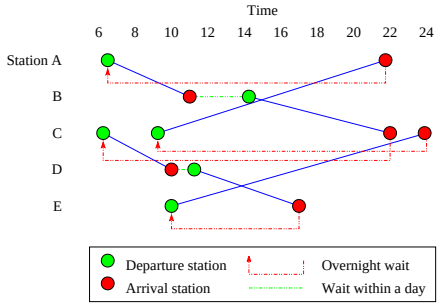


Figure 7: A cycle of four days

The basic problem. The most basic *rolling stock rostering problem* can now be formulated in terms of cycles: given a set of rides with stations and times of departure and arrival, partition this set into a collection of ordered sets of rides. Each ordered set represents a cycle of rides a train has to serve. The cost of the solution is the sum of cycle lengths, representing the total number of trains needed.

The problem with this simple objective, the minimization of the number of trains (implicitly assuming that all trains are identical), has been studied as one of the first problems in operations research. It is known as the minimum fleet size

problem [5]; the earliest proposal [14] solves it, quite naturally, as a minimum cost circulation (a minimum cost flow where the source and the sink nodes are identical [2]). In reality, the problem is far more complex than this variant. Let us pick two practical aspects to illustrate this: the option of deadheading (empty rides) and the necessity of maintenance.

Empty rides. An *empty ride* denotes a (non-scheduled) movement of a train from one station to another one, just in order to continue its service there. When empty rides are allowed, we assume that for every pair of stations, we are given the travelling time between these stations. Consequently, the output is allowed to contain some empty rides. This makes the rolling stock rostering problem more interesting, since we now need to decide which empty rides to pick. Again, the problem has been studied earlier [5, 14], and has been solved to optimality in polynomial time through bipartite matching, with arrival events at stations and departure events at stations as the nodes of the bipartition.

Maintenance. The second practical consideration is the *maintenance* of trains, from the simple collection of trash in cars at major stops to the demanding general overhaul of locomotives once in a few years or after a number of kilometers traveled. For the latter maintenance, only a subset of all train stations is suitable, the *maintenance stations*. Let us consider only the simplest maintenance requirement: eventually, every train must pass through a maintenance station. That is, every cycle must contain a maintenance station, but we do not limit the travel time or distance from one maintenance to the next. Interestingly, maintenance makes rolling stock rostering difficult. While even with empty rides, for the basic rolling stock rostering problem rides are combined into cycles in the same way every day (in general, in every period), this is no longer true when we also request maintenance. For an example, consult Figure 8, taken from [16]. Any train assignment that repeats every day needs three trains. If we change the assignment on alternating days, and hence get a two day cycle for each train, two trains suffice. That is, as a result of requesting maintenance in addition to allowing empty rides, the periodicity of the solution may change.

Hardness and approximation. Maintenance makes the rolling stock rostering problem $\mathcal{APX}$-hard [16]. This can be proved through an approximation preserving reduction from minimum vertex cover on cubic graphs. Theoretically, this is nicely complemented by a factor-2-approximation algorithm for rolling stock rostering with maintenance, but without empty rides, and by a factor-5-approximation with maintenance and empty rides. In practice, however, these approximation guarantees are clearly not good enough.

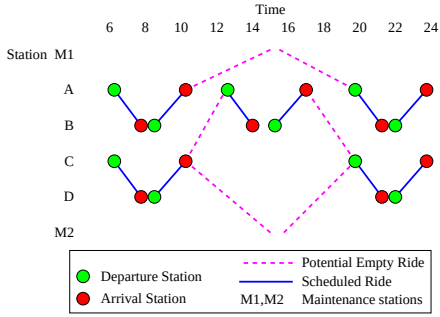


Figure 8: Output restrictions may change solutions.

Relation to the real world problem. Even for the simple problem versions discussed above, the computational complexity of finding an optimum solution can be prohibitive, and the approximation guarantees are unsatisfactory. Let us glimpse at the complications that more realistic models might entail, by discarding the implicit assumptions introduced above. We considered all trains to be identical. In reality, trains differ substantially. The units a train is composed of play an important role (the dining car should be in between first and second class, and those should be at the proper ends of the train, and the locomotive should be suitable for the train weight and the terrain traveled). The train length varies across trains, with lower bounds defined by customer demand, and upper bounds defined by technical constraints such as track length in stations. Furthermore, trains whose parts have their own engine can be split into parts and run separately, while trains with one locomotive cannot. Certainly, trains should in general be allowed to split and recombine. Furthermore, we ignored the fact that stations have limited capacities, and we ignored the topology of the tracks, inside and outside stations. This is a problem already for empty trains: We need to make sure that a chosen empty train will be able to find an empty track for its ride.

Luckily, there are also aspects that the basic rostering model ignored, but that can be taken care of without much extra complexity. Station-turn-around times are an example: whereas a train cannot arrive at a station and depart in an instant, it is obvious that this extra time can just be added to the arrival time of a train. If the train, however, needs to be reconfigured at a station, the extra shunting time depends on the chosen cycle, which in turn depends on the extra shunting time, a complicated problem. Further, we assumed that maintenance happens at an instant, so that the train can continue its journey immediately. This is not quite as unrealistic as it may seem, since train companies typically keep an inventory of a few extra trains that are rotated into service for a train that is being maintained (but an inventory might not be the optimal solution, either). A detailed study and

experimental results with data from Deutsche Bahn is given in [3].

As to the objective function, for our simple rolling stock rostering version with unit trains, the number of trains was the natural focus. In reality, we should at least generalize to the number of locomotives and cars needed. In addition, we need to take the cost per traveled unit of distance into account, as well as the cost for train crew, for shunting, and for coupling and decoupling trains.

For some of the above problems with a variety of aspects, valuable solutions have been found. As an example, an integration of vehicle scheduling and crew scheduling has been proposed [17], based on Lagrangean relaxation. Nevertheless, from a computational complexity point of view, the understanding of the mix of the above problem ingredients is in its infancy.

4 Timetable adjustments and rolling stock assignment

In Section 3 we studied the problem of assigning trains to scheduled rides. We saw that the minimum number of train units needed to run a fixed schedule can be efficiently determined by means of flow algorithms. To minimize the number of necessary train units is indeed an important objective, since just owning a locomotive is expensive, no matter if it is used a lot, or just standing around. Hence it might be considered reasonable to allow slight changes to the timetable, if this is necessary to reduce the number of train units. We denote this problem as Flexible Rostering Problem.

Consider for example the following two rides. The first one is leaving Berlin at 6 o'clock, and is arriving in Warsaw six hours latter. The other ride is leaving Warsaw at 11 o'clock back to Berlin. Both rides can be performed with the same train unit if the second ride is postponed by 1 hours and 30 minutes.

The Scheduling with Release Times and Deadlines on a Minimum Number of Machines (SRDM) problem has been studied in [12]. It is a special case of the described flexible rostering problem, where there is only one station, and all rides depart and arrive at this station. For every ride a time interval is given that indicates when it can be carried out. The task is to schedule all rides with a minimal number of trains.

In this section we present selected results from [12]. Since the SRDM problem is a classical scheduling problem we use the appropriate notation. Rides correspond to jobs and durations of rides correspond to the processing time of a job.

Model and notation. Each job of the input is associated with a release time r, a deadline d, and a processing time p, where r, d, and p are integers and $d - r \geq p > 0$. The interval $[r, d)$ is the window in which an interval of size p will be placed. If the size $d - r$ of the window is equal to p, the job occupies the whole window. If the window of a job is larger than its processing time, the choice of a schedule implies for each considered job shifting an interval (the processing interval) into a position within a larger interval (the window). Thus a triple $\langle r, d, p\rangle$ is called a shiftable interval. The difference $\delta = d - r - p$ is the *slack* and corresponds to the maximum amount the interval can be moved within its window. Observe that this notion of slack is different from the one considered in Section 2. For the SRDM problem we will stick to the notation of [12].

For every interval we have to select a position within its window. This position is described by a *placement* $\phi \in \{0, \ldots, \delta\}$. The processing interval according to a placement ϕ is denoted by $J^\phi = [r + \phi, r + \phi + p)$.

For an n-tuple $\mathcal{S} = (J_1, \ldots, J_n)$ of shiftable intervals, $\Phi = (\phi_1, \ldots, \phi_n)$ defines a placement, where for $1 \leq i \leq n$ the value ϕ_i is the placement of the shiftable interval J_i. Together $\mathcal{S}$ and Φ describe a finite collection of intervals $\mathcal{S}^\Phi = \{J_i^{\phi_i} \mid i = 1, \ldots, n\}$, which can be interpreted as an interval graph G (as defined for example in [7]). The maximum number of intervals that contain some position x is called the *height* of $\mathcal{S}^\Phi$.

The Scheduling with Release Times and Deadlines on a Minimum Number of Machines (SRDM) problem can now be formulated in terms of shiftable intervals: given an n-tuple $\mathcal{S}$ of shiftable intervals, find a placement Φ minimizing the height of the interval set $\mathcal{S}^\Phi$.

Combinatorial and complexity results. Several exact and approximation algorithms for the SRDM problem and special cases of it are considered in [12]. If all shiftable intervals have slack 0 the SRDM problem is equal to finding the maximum clique of the corresponding interval graph, which can be solved in polynomial time. On the other hand, if for all shiftable intervals the release times and deadlines are equal the SRDM problem is equal to the classical bin packing problem [13], which is NP-hard.

For train companies usually only a slight change of the schedule is acceptable. This means that the window of a job is just slightly larger than its processing time. Thus understanding SRDM instances with small slack is important. In this section we present a polynomial time algorithm for instances where the slack of the shiftable intervals is bounded by $\delta_{\max} = 1$.

Although the SRDM problem is easy to solve if the maximum slack $\delta_{\max}$ is 0 or 1, the problem becomes NP-hard for the any restriction $\delta_{\max} \geq 2$. Furthermore there is an $\Omega(\log n)$-approximation algorithm to SRDM [12]. Interestingly, for

small windows even an arbitrary placement is a good approximation to SRDM. Constant approximation algorithms are known for instances with equal processing times and with a restricted ratio of processing times [12].

A polynomial time algorithm for SRDM **with maximum slack** 1**.** The placement of a shiftable interval $J = \langle r, d, p\rangle$ with slack 1 is either 0 or 1. Hence, the corresponding interval contains either the point r ($\phi = 0$) or $d - 1$ ($\phi = 1$). This observation leads to a network flow formulation to solve the SRDM(m) problem.

We use a small example to explain this algorithm which is described in detail in [12]. We are given three shiftable intervals $J_1 = \langle 0, 3, 2\rangle$, $J_2 = \langle 2, 6, 3\rangle$, and $J_3 = \langle 5, 7, 2\rangle$. The shiftable intervals J_1 and J_2 have slack 1. The problem is to decide if it is possible to schedule all jobs on one machine.

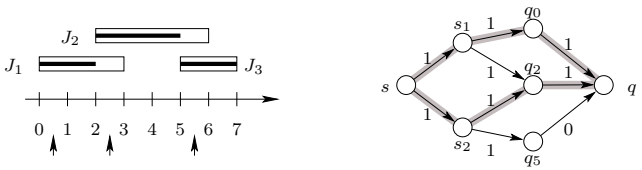


Figure 9: Example for slack at most 1.

On the left of Figure 9 the three shiftable intervals are depicted. To its right the corresponding network flow formulation is shown. If it is possible to schedule all jobs on one machine then for every shiftable interval with slack 1, one unit of flow is sent from source s to sink q. For every shiftable interval with slack 1 we add a node s. In our example these are nodes s_1 and s_2. These nodes are connected with the source s by edges with capacity 1.

Since shiftable intervals with slack 1 only have two possible placements, J_1 contains either point 0 or 2. For both placements it always contains point 1. Similarly the shiftable interval J_2 can either occupy point 2 or 5, but independently of the placement it always occupies points 3 and 4. We add to the network three nodes q_0, q_2, and q_5. Node s_1 is connected to q_0 and q_2 furthermore s_2 is connected to q_2 and q_5. Again these edges have a capacity of 1. A flow on one of these edges determines a unique placement of the corresponding shiftable interval. For instance a flow on edge (s_2, q_2) corresponds to placing J_2 rightmost.

Since all jobs have to be scheduled on one machine the placement of J_2 can only be 0 when J_1 is placed left as well. To model this we introduce capacities on the edges (q_0, q), (q_2, q), and (q_5, q). The capacities of these edges depend on the number of jobs always occupying point $0, 2$, or 5. Since J_3 always occupies point 5 and all jobs have to be scheduled on one machine, it is not possible to place J_2

at its rightmost position. Thus the capacity of edge (q_5, q) has to be 0. On the other hand point 0 and 2 can only be occupied by a placement of J_1 or J_2. Thus the capacity of the edges (q_1, q) and (q_2, q) is 1.

In Figure 9 a maximum flow of size 2 is depicted by the thick gray edges. Both shiftable intervals J_1 and J_2 have to be placed to the left.

The described construction can be extended to more machines by adjusting the capacities on the edges of type (q_i, q).

Related work. The SRDM problem has recently gained interest. Chuzhoy and Naor [11] have studied the machine minimization problem for sets of alternative processing intervals. The input consists of n jobs and each of them is associated with a set of time intervals. A job is scheduled by choosing one of its intervals. The objective is to schedule all jobs on a minimum number of machines. They show that the machine minimization problem is $\Omega(\log \log n)$-hard to approximate unless NP $\subseteq DTIME(n^{O(\log \log \log n)})$. Very recently Chuzhoy, Guha, S. Khanna, and Naor [10] presented an $O(\text{OPT})$ approximation algorithm for the machine minimization problem with continuous input. In the continuous version they allowed intervals associated with each job to form a continuous line segment, described by a release time and a deadline.

So far instances with numerous stations and a periodic time table have not been considered. Flexible train rostering for periodic time tables has been studied in [15]. There the notions of unlimited flexibility and limited flexibility is introduced. If we allow unlimited flexibility, only the duration of rides are given and we are free to determine the most appropriate departure time. For problem instances with limited flexibility the original departure times are given, together with a slack ε. Furthermore empty rides between stations are allowed. For instances with unlimited flexibility and empty rides a schedule with a minimum number of train units can be determined in polynomial time. In contrast, not allowing empty rides makes the problem NP-hard [15].

5 Network planning: Locating new stations

The stations in a given and fixed railway network are only attractive to travelers that reside in the vicinity of the stations. This vicinity is a subjective measure that depends on the travelers' preferences, as well as on their available modes of pre- and post-transportation.

When considering locations for new stations, one typically assumes that a new station is attractive for all travelers residing in a certain radius R around that station. Further, the potential travelers are assumed to reside at settlements, modeled by points in the Euclidean plane, and each settlement has a weight, representing

the number of residents. Any settlement within the radius R of a potential new station is said to be *covered* by that station. The new stations can only be located along the tracks of the existing railway network, which is considered as a connected set of straight line segments.

Thus, the input of the station location problem consists of a set of straight line segments and a set of settlements, both in the Euclidean plane, and the problem asks for a set of stations along these line segments. Typically, this set of stations should either minimize the number of stations required to cover all settlements, or maximize the weight of the settlements coverable with a fixed number of stations.

The station location problem was first considered by Hamacher, Liebers, Schöbel and D. Wagner and F. Wagner [22], who proved that it is NP-hard to determine whether all settlements can be covered by a fixed number of new stations. The problem input may be such that the railway network decomposes into a (disconnected) set of line segments. For such a single line, Schöbel, Hamacher, Liebers and Wagner [37] observe that the problem boils down to a set covering problem with a totally unimodular constraint matrix, to be more precise, an interval matrix. Thus, in this case the problem can be solved in polynomial time by Linear Programming. For the case of two intersecting lines, Mammana, Mecke, and Wagner [25] propose an algorithm that runs in polynomial time, under the restriction that the angle between the two segments is large enough. The algorithm utilizes results from the covering by discs problem.

On a single line segment the problem can be solved by a dynamic programming approach [24]. Under certain restrictions, this dynamic program can be modified to solve the case of two parallel lines. Alternative objective functions have been considered, for example of minimizing the saved travel time over all passengers [22]. This objective considers the trade-off between, on one hand, the travel time decrease for travelers whose distance to the nearest station decreases, and on the other hand the increase in travel time due to the fact that trains call at the new stations. For this objective of overall saved travel time the problem is NP-hard [22]. Also the bicriterial version of this problem has been considered and is shown to be NP-hard in general, but can be solved efficiently on a single straight line [35].

6 Timetable information systems

One of the few computer systems that are already visible to passengers are timetable information systems. The task is to supply the passengers with a good itinerary from the timetable of the train system. We judge the quality of an itinerary usually by its duration, the number of transfers and sometimes monetary cost, but other measures are conceivable as well. For efficient optimization

we need to understand and exploit the structure of the cost functions. If this is unknown, for example if a real world price-system is only available as a black box that can price an itinerary, there is in general no way to optimize other than to consider an exponential number of possibilities. But even if the details of the price system are known, it is usually non-trivial to optimize for these prices. If, for example, a fixed price supplement is necessary if a certain train type is used at all, the price for the complete journey is no longer the sum of the prices of the connections, because the supplement might be necessary for all connections, but only once for the complete journey. Additionally, the passengers are not used to provide the system with their individual weighting of travel-time versus number of transfers versus monetary price. Instead we might want to present all Pareto-optima, i.e., all different itineraries that minimize duration for some constraint on the number of transfers and a price limit. The number of such itineraries can in general be exponential, but in the real world there seems to be a sufficient correlation between the different measures such that it is feasible to compute them all [27]. Another important aspect for a concrete system is the timetable data. For the user of the system it is advantageous if many timetables (different countries, local trains, busses, etc.) are included. The size of such data does not only require more computational effort, it is also a challenge to keep huge data sets from different sources correct and up-to-date.

We can easily model itineraries as paths in a graph, and find the optimal itinerary by searching for a shortest path. In its simplest version we are given the timetable, and as a request the source and destination station of a traveler, and an earliest possible departure time. Then we are searching for a sequence of trains that gets the traveler to the destination station at the earliest possible time. In the simplest model we assume that changing trains takes no time, i.e., that the journey can continue with any train that leaves after the traveler has arrived at the station. We further assume that the timetable is repeated daily. Avoiding these assumptions is conceptually not very complicated, but the resulting increase in network size might very well be a problem for an implementation. There are two main modeling approaches known, using either a time expanded or a time dependent network.

The time expanded approach (used for example in [26, 27, 36]) constructs a directed graph with one node for every event that a train arrives or departs at some station, very similar to the event-activity network described in Section 2. These nodes are connected at the stations in a cyclic manner, representing the possibility to wait at the station. Furthermore, there is one link from every departure event to the corresponding arrival event of the same train at its next station. The weight of an edge is given by the time difference of its events. Now any valid itinerary corresponds to a path in this time expanded graph, and vice versa. A detailed description of this approach can be found for example in [36]. A request for an

itinerary translates to a single-source, multi-sink shortest path question with the earliest event at the source station after the possible departure time as source, and all events at the target station as sink. The optimal such path can be found using Dijkstra's algorithm once. This approach is direct, but it has the disadvantage that it might continue to explore events at a station, even though all next connections have already been explored.

Alternatively, the time dependent approach, as used for example in [4, 8, 28, 31, 32] works on a smaller graph with more complicated edges. More precisely, there is one node per station, and an edge from one station to another, if there is a non-stop connection. The edges are annotated with *link traversal* functions. Given an earliest possible departure time at the beginning station of a link, such a function gives the earliest possible time one can arrive at the end station of that link. For a timetable, this leads to piecewise constant functions. A path generalizes to a timed path, where we additionally have a time when the path visits a particular vertex. We require that the time at a successor vertex is given by applying the link traversal function to the current time. Given that the link traversal functions are monotonic and correspond to non-negative delay (i.e. $f(t) \geq t$), a slight modification of Dijkstra's algorithm can compute a path with earliest possible arrival at the destination.

The time dependent approach has the clear advantage that the shortest path algorithm has to consider a smaller graph. It is additionally very easy to incorporate other *modes* of travel, like walking or taking a taxi, into the network [4]. In return it is necessary to evaluate the link traversal functions, basically searching for the definition interval of the piecewise constant function. This overhead can be avoided by replacing the resulting time with a pointer into the table representing the function of the links at the next station. This modification can be carried out in a way that the time dependent algorithm is guaranteed not to perform more CPU work than the time expanded approach [8].

This worst-case analysis is confirmed by experiments [32] where the time dependent approach was 10 to 40 times faster for this simple model.

This clear picture is not the whole story. Usually we would like to disallow impossibly short transfer times, to have a limit on the number of train changes, and other extensions that allow a more detailed modeling of reality. This is usually done by introducing additional nodes (splitting stations [4, 31]) and links, which tends to hurt the time dependent approach. In contrast the time expanded approach already has lots of vertices, such that the more detailed models are mainly changing some links but do not introduce a significant number of additional vertices and edges.

A practical system need not necessarily perform shortest path computations. >From a theoretical computer science point of view, we are looking at a data structure question. The goal is to build once a data-structure from the timetable,

and then be able to answer queries quickly. Such a query consists of constantly many objects, a pair of nodes, one time, one mode, perhaps one intermediate stop. This immediately gives rise to a data structure, since the number of different queries is polynomial, so that storing a big table with the answers to all possible queries leads to polynomial space usage. For real systems this approach has not been followed. Instead, there are many heuristics to speed up the computation of an optimal shortest path. Many of them use the geometric embedding of the graph that is given by the geographical positions. Most prominent is the use of goal directed search, also known as A^*-algorithm in the artificial intelligence literature. We can modify the weights of the links by adding a potential, as long as this does not introduce negative weights. Then the length-difference of the paths in the network remains unchanged, and in particular the shortest paths remain shortest. If we know the Euclidean distance to the destination and the maximum speed we have a lower bound on the travel time. Using this estimate on the travel time as potential, we do not introduce negative weight edges. Even though there are worst-case examples where Dijkstra's algorithm still needs to explore the whole graph only to realize that the shortest path consists of a single edge, we know that there is an expected performance advantage for a certain class of random geometric graphs [38]. Also in practice this approach seems to be successful. There are also approaches that perform a preprocessing step that computes all shortest paths, and remembers only some condensed information, for example for each edge e a sector of the plane that contains all target stations t such that there is some shortest path to t using the edge e [41]. More engineering aspects have been investigated, for example reducing the space usage [26].

In contrast with the other problems we considered, it seems that the fundamental issues of timetable information systems have been addressed and are reasonably well understood. Naturally it remains important to optimize the actually used algorithms to exploit the special structure of the concrete data-sets as much as possible.

References

[1] B. Adenso-Diaz, M. Gonzalez, and P. Gonzalez Torre. On-line timetable rescheduling in regional train services. *Transportation Research-B*, 33/6:387–398, 1999.

[2] R. K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network flows - theory, algorithms, and applications*. Prentice Hall, 1993.

[3] L. Anderegg, S. Eidenbenz, M. Gantenbein, C. Stamm, D.S. Taylor, B. Weber, and P. Widmayer. Train routing algorithms: Concepts, design choices, and practical

considerations. In *Proc. 5th Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 106–118. SIAM, 2003.

[4] C. Barrett, K. Bisset, R. Jacob, G. Konjevod, and M. Marathe. Classical and contemporary shortest path problems in road networks: Implementation and experimental analysis of the transims router. In *Proc. 10th Annual European Symposium on Algorithms (ESA)*, pages 126–138. Springer-Verlag LNCS 2461, 2002.

[5] A. Bertossi, P. Carraresi, and G. Gallo. On some matching problems arising in vehicle scheduling models. *Networks*, 17:271–281, 1987.

[6] C. Biederbick and L. Suhl. Improving the quality of railway dispatching tasks via agent-based simulation. In *Computers in Railways IX*, pages 785–795. WIT Press, 2004.

[7] A. Brandstädt, V.B. Le, and J.P. Spinrad. *Graph Classes: a Survey*. SIAM Monographs on Discrete Mathematics and Applications, 1999.

[8] G. Brodal and R. Jacob. Time-dependent networks as models to achieve fast exact time-table queries. In *Proc. Algorithmic Methods and Models for Optimization of Railways (ATMOS)*, volume 92 of *Electronic Notes in Theoretical Computer Science*, pages 3–15. Elsevier Science, 2003.

[9] M. Bussieck, T. Winter, and U. Zimmermann. Discrete optimization in public rail transportation. *Mathematical Programming*, 79(3):415–444, 1997.

[10] J. Chuzhoy, S. Guha, S. Khanna, and J.S. Naor. Machine minimization for scheduling jobs with interval constraints. To appear in FOCS 2004.

[11] J. Chuzhoy and J. Naor. New hardness results for congestion minimization and machine scheduling. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 28–34, 2004.

[12] M. Cieliebak, T. Erlebach, F. Hennecke, B. Weber, and P. Widmayer. Scheduling jobs on a minimum number of machines. In *Proc. 3rd IFIP International Conference on Theoretical Computer Science*, pages 217 – 230. Kluwer, 2004.

[13] E.G. Coffman Jr., M.R. Garey, and D.S. Johnson. Approximation algorithms for bin packing: A survey. In D. Hochbaum, editor, *Approximation Algorithms for NP-hard Problems*. PWS, 1996.

[14] G. Dantzig and D. Fulkerson. Minimizing the number of tankers to meet a fixed schedule. *Nav. Res. Logistics Q.*, 1:217–222, 1954.

[15] S. Eidenbenz, A. Pagourtzis, and P. Widmayer. Flexible train rostering. In *Proc. 14th International Symposium on Algorithms and Computation (ISAAC)*, pages 615 – 624. Springer-Verlag LNCS 2906, 2003.

[16] T. Erlebach, M. Gantenbein, D. Hürlimann, G. Neyer, A. Pagourtzis, P. Penna, K. Schlude, K. Steinhöfel, D.S. Taylor, and P. Widmayer. On the complexity of train assignment problems. In *Proc. of the 12th Annual International Symposium on Algorithms and Computation (ISAAC)*, pages 390–402. Springer-Verlag LNCS 2223, 2001.

[17] R. Freling, D. Huisman, and A. Wagelmans. Models and algorithms for integration of vehicle and crew scheduling. *J. of Scheduling*, 6(1):63–85, 2003.

[18] M. Gatto, B. Glaus, R. Jacob, L. Peeters, and P. Widmayer. Railway delay management: Exploring its algorithmic complexity. In *Algorithm Theory - Proceedings SWAT 2004*, pages 199–211. Springer-Verlag LNCS 3111, 2004.

[19] M. Gatto, R. Jacob, L. Peeters, and A. Schöbel. The computational complexity of delay management. Technical Report 456, ETH Zurich, 2004.

[20] M. Gatto, R. Jacob, L. Peeters, and P. Widmayer. On-line delay management on a single line. In *Proc. Algorithmic Methods and Models for Optimization of Railways (ATMOS)*. Springer-Verlag LNCS, 2004. To appear.

[21] A. Ginkel and A. Schöbel. The bicriterial delay management problem. Report in Wirtschaftsmathematik 85/2002, University of Kaiserslautern, 2002.

[22] H. Hamacher, A. Liebers, A. Schöbel, D. Wagner, and F. Wagner. Locating new stops in a railway network. In *Electronic Notes in Theoretical Computer Science*, volume 50. Elsevier, 2001.

[23] D. Heimburger, A. Herzenberg, and N. Wilson. Using simple simulation models in the operational analysis of rail transit lines: A case study of the MBTA's red line. *Transportation Research Record*, 1677:21–30, 1999.

[24] E. Kranakis, P. Penna, K. Schlude, D. Taylor, and P. Widmayer. Improving customer proximity to railway stations. In *Proc. 5th Italian Conference on Algorithms and Complexity*, pages 264–276. Springer-Verlag LNCS 2653, 2003.

[25] F. Mammana, S. Mecke, and D. Wagner. The station location problem on two intersecting lines. In *Proc. Algorithmic Methods and Models for Optimization of Railways (ATMOS 2003)*, volume 92 of *Electronic Notes in Theoretical Computer Science*, pages 65–84. Elsevier Science, 2003.

[26] M. Müller-Hannemann, M. Schnee, and K. Weihe. Getting train timetables into the main storage. In *Proc. Algorithmic Methods and Models for Optimization of Railways (ATMOS)*, volume 66 of *Electronic Notes in Theoretical Computer Science*. Elsevier Science, 2002.

[27] M. Müller-Hannemann and K. Weihe. Pareto shortest paths is often feasible in practice. In *Proc. 5th International Workshop on Algorithm Engineering (WAE)*, pages 185–197. Springer Verlag, LNCS 2141, 2001.

[28] K Nachtigall. Time depending shortest-path problems with applications to railway networks. *European Journal of Operational Research*, 83(1):154–166, 1995.

[29] K. Nachtigall. *Periodic Network Optimization and Fixed Interval Timetables*. Habilitation Thesis, Braunschweig, Germany, 1998.

[30] S. O'Dell and N. Wilson. Optimal real-time control strategies for rail transit operations during disruptions. In *Computer-Aided Transit Scheduling*, volume 471 of *Lecture Notes in Economics and Mathematical Systems*, pages 299–323. Springer-Verlag, 1999.

[31] E. Pyrga, F. Schulz, D. Wagner, and C. Zaroliagis. Towards realistic modeling of time-table information through the time-dependent approach. In *Proc. Algorithmic Methods and Models for Optimization of Railways (ATMOS)*, volume 92 of *Electronic Notes in Theoretical Computer Science*, pages 85–103. Elsevier Science, 2003.

[32] E. Pyrga, F. Schulz, D. Wagner, and C. Zaroliagis. Experimental comparison of shortest path approaches for timetable information. In *Proc. 6th Workshop on Algorithm Engineering and Experiments and the First Workshop on Analytic Algorithmics and Combinatorics*, pages 88–99. SIAM, 2004.

[33] A. Schöbel. A model for the delay management problem based on mixed-integer-programming. In *Electronic Notes in Theoretical Computer Science*, volume 50. Elsevier, 2001.

[34] A. Schöbel. *Customer-oriented optimization in public transportation*. Habilitation Thesis, University of Kaiserslautern, 2003. To appear.

[35] A. Schöbel. Locating stops along bus or railway lines — a bicriterial problem. *Annals of Operations Research*, 2003. to appear.

[36] F. Schulz, D. Wagner, and K. Weihe. Dijkstra's algorithm on-line: An empirical case study from public railroad transport. *Journal of Experimental Algorithmics*, 5(12), 2000.

[37] A. Schöbel, H.W. Hamacher, A. Liebers, and D. Wagner. The continuous stop location problem in public transportation. Technical report, Universität Kaiserslautern, 2002. Report in Wirtschaftsmathematik Nr. 81/2001.

[38] R. Sedgewick and J. Vitter. Shortest paths in euclidean graphs. *Algorithmica*, 1:31–48, 1986.

[39] L. Suhl, C. Biederbick, and N. Kliewer. Design of customer-oriented dispatching support for railways. In *Computer-Aided Scheduling of Public Transport*, volume 505 of *Lecture Notes in Economics and Mathematical Systems*, pages 365–386. Springer-Verlag, 2001.

[40] L. Suhl, T. Mellouli, C. Biederbick, and J. Goecke. Managing and preventing delays in railway traffic by simulation and optimization. In *Mathematical methods on optimization in transportation systems*, volume 48, pages 3–16. Kluver Academic Publishers, 2001.

[41] D. Wagner, T. Willhalm, and C. Zaroliagis. Dynamic shortest paths containers. In *Proc. Algorithmic Methods and Models for Optimization of Railways (ATMOS)*, volume 92 of *Electronic Notes in Theoretical Computer Science*, pages 65–84. Elsevier Science, 2003.

[42] D. West. *Introduction to Graph Theory*. Prentice Hall, Upper Saddle River, N.J., second edition, 2001.

The Computational Complexity Column

by

Jacobo Torán

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

Parameterized Complexity in its origins was considered by many researchers to be an exotic research field, orthogonal to the standard way of classifying problems in complexity theory. In the last years however many surprising connections between Parameterized Complexity and "classical" areas in complexity theory have been established. Jörg Flum and Martin Grohe survey in this column some of these interesting connections including links to the areas of bounded nondeterminism, subexponential complexity or syntactic complexity classes.

Parameterized Complexity and Subexponential Time

Jörg Flum* Martin Grohe†

*Abteilung für Mathematische Logik, Albert-Ludwigs-Universität Freiburg, Eckerstr. 1, 79104 Freiburg, Germany. Email: flum@uni-freiburg.de

†Institut für Informatik, Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany. Email: grohe@informatik.hu-berlin.de

1. Introduction

Over the last 15 years, the theory of fixed-parameter tractability [13] has developed into a well-established branch of algorithm design and complexity theory. In this theory, the running time of algorithms is analyzed not only in terms of the input size, but also in terms of an additional parameter of problem instances. An algorithm is called *fixed-parameter tractable (fpt)* if its running time is possibly super-polynomial in terms of the parameter of the instance, but polynomial in the size. More precisely, an algorithm is fpt if its running time is

$$f(k) \cdot n^{O(1)} \tag{1.1}$$

for some computable function f, where n denotes the size of the input and k the parameter. The idea is to choose the parameterization in such a way that the parameter is small for problem instances appearing in a concrete application at hand. Since $f(k)$ is expected to be moderate for small k, fixed-parameter tractability is a reasonable approximation of practical tractability for such problem instances.

Fixed-parameter tractability is thus a specific approach to the design of *exact algorithms for hard algorithmic problems*, an area which has received much attention in recent years (see, for example, [17, 27]). Well known examples of nontrivial exact algorithms are the ever improving algorithms for the 3-satisfiability problem [24, 25, 8, 20], the currently best being due to Iwama and Tamaki [20] with a running time of roughly 1.324^n, where n is the number of variables of the input formula. In this article, we are mainly interested in lower bounds for exact algorithms. For example, is there an algorithm that solves the 3-satisfiability problem in time $2^{o(n)}$? The assumption that there is no such algorithm is known as the *exponential time hypothesis (ETH)*. The exponential time hypothesis and related assumptions have been studied from a complexity theoretic point of view in [14, 19, 18, 26]. Most notably, Impagliazzo, Paturi, and Zane [19] have started to develop a theory of hardness and completeness for problems with respect to subexponential time solvability. An ultimate goal of such a theory would be to show the equivalence of assumptions such as (ETH) with more established assumptions such as P $\neq$ NP. Of course it is not clear at all if such an equivalence can be proved without actually proving (ETH). Overall, we believe that it is fair to say that subexponential time complexity is not very well understood.

What singles out fixed-parameter tractability among other paradigms for the design of exact algorithms for hard algorithmic problems is that it is complemented by a very well-developed theory of intractability. It is known for quite a while that this intractability theory has close connections with subexponential time complexity and the exponential time hypothesis [1]. But only recently have these connections moved to the center of interest of researchers in parameterized

complexity theory [3, 10, 4, 5, 6]. This shift of interest was caused by attempts to prove lower bounds for the parameter dependence (the function f in (1.1)) of fpt-algorithms [3] and the investigations of *miniaturized problems* in this context [10].

The purpose of this article is to explain these connections between parameterized and subexponential complexity. The intention is not primarily to survey the most recent developments, but to explain the technical ideas in sufficient detail. (For a recent survey on parameterized complexity theory, see, for example, [9].) The main technical results are reductions between the satisfiability problem and the weighted satisfiability problem, which asks for satisfying assignments setting a specific number k of the variables to TRUE. We call an assignment setting exactly k variables to TRUE a *weight k assignment*. The reductions are based on a simple idea known as the *k-log-n trick*: Specifying a weight k assignment to a set of n variables requires $k \cdot \log n$ bits. This can be used to reduce weighted satisfiability of a formula with n variables to unweighted satisfiability of a formula with only $k \cdot \log n$ variables. A similar reduction can be used in the converse direction. To obtain reasonably tight reductions for specific classes of propositional formulas, some care is required. The construction is carried out in the proof of Theorem 4.4.

All results presented in this article are known (essentially, they go back to [1]), and they are not very deep. Nevertheless, we believe it is worth while to present the results in a uniform and introductory manner to a wider audience. Our presentation may be slightly unfamiliar for the experts in the area, as it is based on a new *M-hierarchy* of parameterized complexity classes. We show that this hierarchy is entangled with the familiar W-hierarchy. The M-hierarchy is a translation of a natural hierarchy of satisfiability problems into the world of parameterized complexity, and fixed-parameter tractability of the M-classes directly translates to subexponential complexity of the corresponding satisfiability problems. Let us emphasize that even though we will develop the theory in the setting of parameterized complexity, it directly applies to subexponential complexity. The connection will be made explicit in the last section of the article.

The article is organized as follows: After introducing our notation, we start with a brief introduction into parameterized complexity theory. In Section 4, we introduce the M-hierarchy and establish the connections between the M-hierarchy and subexponential time complexity on the one hand, and between the M-hierarchy and the W-hierarchy on the other hand. In Section 5, we study the miniaturized problems that originally led to the introduction of the class M[1]. We prove a number of completeness results for M[1], which are based on a combinatorial lemma known as the *Sparsification Lemma* [19]. (The proof of the Sparsification Lemma itself is beyond the scope of this article.) We put these results in the wider context of the syntactically defined complexity class SNP in Section 6. Finally, in Section 7, we translate the results back to the world of classical

complexity theory and the exponential time hypothesis.

One nice aspect of this area is that it has a number of very interesting open problems. We conclude this article by listing a few of them.

2. Notation

The set of natural numbers (that is, positive integers) is denoted by $\mathbb{N}$. For integers n, m, we let $[n, m] = \{n, n+1, \ldots, m\}$ and $[n] = [1, n]$. Unless mentioned explicitly otherwise, we encode integers in binary.

We use $\log n$ to denote the binary (base 2) logarithm of $n \in \mathbb{N}$.

For computable functions $f, g : \mathbb{N} \to \mathbb{N}$, we say that f is *effectively little-oh of* g and write $f \in o^{\text{eff}}(g)$ if there exist $n_0 \in \mathbb{N}$ and a computable function $\iota : \mathbb{N} \to \mathbb{N}$ that is non-decreasing and unbounded such that for all $n \geq n_0$,

$$f(n) \leq \frac{g(n)}{\iota(n)}.$$

We mostly use the letter ι to denote computable functions that are non-decreasing and unbounded (but possibly growing very slowly).

Throughout this paper we work with the effective version of "little-oh". In particular, we require subexponential algorithms to have a running time of $2^{o^{\text{eff}}(n)}$ and not just $2^{o(n)}$. The reason for this is that it gives us a correspondence between "strongly uniform" fixed-parameter tractability and subexponential complexity. A similar correspondence holds between "little-oh" instead of "effective little-oh" and "uniform fixed-parameter tractability" instead of "strongly uniform fixed-parameter tractability". We prefer to work with strongly uniform fixed-parameter tractability as it has a more robust theory.

2.1. Propositional Logic

Formulas of *propositional logic* are built up from *propositional variables* X_1, $X_2, \ldots$ by taking conjunctions, disjunctions, and negations. The negation of a formula α is denoted by $\neg\alpha$. We distinguish between *small conjunctions*, denoted by $\wedge$, which are just conjunctions of two formulas, and *big conjunctions*, denoted by $\bigwedge$, which are conjunctions of arbitrary finite sequences of formulas. Analogously, we distinguish between *small disjunctions*, denoted by $\vee$, and *big disjunctions*, denoted by $\bigvee$.

The set of variables of a formula α is denoted by $\text{var}(\alpha)$. An *assignment* for a formula α is a mapping $\mathcal{V} : \text{var}(\alpha) \to \{\text{TRUE}, \text{FALSE}\}$, and we write $\mathcal{V} \models \alpha$ to denote that $\mathcal{V}$ satisfies α.

We use a similar notation for *Boolean circuits*. In particular, we think of the input nodes of a circuit γ as being labeled with variables, use $\mathrm{var}(\gamma)$ to denote the set of these variables, and for an assignment $\mathcal{V} : \mathrm{var}(\gamma) \to \{\text{TRUE}, \text{FALSE}\}$ we write $\mathcal{V} \models \gamma$ to denote that γ computes TRUE if the input nodes are assigned values according to γ.

The class of all propositional formulas is denoted by PROP, and the class of all Boolean circuits by CIRC. Usually, we do not distinguish between formulas and circuits, that is, we view PROP as a subclass of CIRC.

For $t \geq 0$, $d \geq 1$ we inductively define the following classes $\Gamma_{t,d}$ and $\Delta_{t,d}$ of propositional formulas:[1]

$$\begin{aligned}
\Gamma_{0,d} &= \{\lambda_1 \wedge \ldots \wedge \lambda_c \mid c \leq d,\ \lambda_1, \ldots, \lambda_c \text{ literals}\},\\
\Delta_{0,d} &= \{\lambda_1 \vee \ldots \vee \lambda_c \mid c \leq d,\ \lambda_1, \ldots, \lambda_c \text{ literals}\},\\
\Gamma_{t+1,d} &= \{\bigwedge_{i \in I} \delta_i \mid I \text{ finite index set and } \delta_i \in \Delta_{t,d} \text{ for all } i \in I\},\\
\Delta_{t+1,d} &= \{\bigvee_{i \in I} \gamma_i \mid I \text{ finite index set and } \gamma_i \in \Gamma_{t,d} \text{ for all } i \in I\}.
\end{aligned}$$

$\Gamma_{2,1}$ is the class of all formulas in *conjunctive normal form*, which we often denote by CNF. For $d \geq 1$, $\Gamma_{1,d}$ is the class of all formulas in *d-conjunctive normal form*, which we denote by d-CNF.

The *size* $|\gamma|$ of a circuit γ is the number of nodes plus the number of edges; thus for formulas the size is O(number of nodes). We usually use the letter m to denote the size of a formula or circuit and the letter n to denote the number of variables.

3. Fundamentals of Parameterized Complexity Theory

3.1. Parameterized Problems and Fixed-Parameter Tractability

As it is common in complexity theory, we describe decision problems as languages over finite alphabets Σ. To distinguish them from parameterized problems, we refer to problems $Q \subseteq \Sigma^*$ as *classical problems*.

A *parameterization* of Σ^* is a mapping $\kappa : \Sigma^* \to \mathbb{N}$ that is polynomial time computable. A *parameterized problem* (over Σ) is a pair (Q, κ) consisting of a set

[1] We prefer to use Γ and Δ instead of the more common Π and Σ to denote classes of propositional formulas (Γ for conjunctions, Δ for disjunctions). The reason is that we want to reserve Π and Σ for classes of formulas of predicate logic. Often in parameterized complexity, it is necessary to jump back and forth between propositional and predicate logic, and it is helpful to keep them strictly separated on the notational level.

$Q \subseteq \Sigma^*$ and a parameterization κ of Σ^*. If (Q, κ) is a parameterized problem over the alphabet Σ, then we call strings $x \in \Sigma^*$ *instances* of Q or of (Q, κ) and the numbers $\kappa(x)$ the corresponding *parameters*. Slightly abusing notation, we call a parameterized problem (Q, κ) a *parameterization* of the classical problem Q.

Usually, when representing a parameterized problem we do not mention the underlying alphabet explicitly and use a notation as illustrated by the following examples.

Example 3.1. Recall that a *vertex cover* in a graph $G = (V, E)$ is a subset $S \subseteq V$ such that for each edge $\{u, v\} \in E$, either $u \in S$ or $v \in S$. The *parameterized vertex cover problem* is defined as follows:

p-VERTEX-COVER	
Instance:	A graph G and a natural number $k \in \mathbb{N}$.
Parameter:	k.
Problem:	Decide if G has a vertex cover of size k.

Example 3.2. The *parameterized satisfiability problem for Boolean circuits* is defined as follows:

p-SAT(CIRC)	
Instance:	A Boolean circuit γ.
Parameter:	$\lvert\mathrm{var}(\gamma)\rvert$.
Problem:	Decide if γ is satisfiable.

More generally, for a class Γ of circuits or formulas, we let *p*-SAT(Γ) denote the restriction of *p*-SAT(CIRC) to instances $\gamma \in \Gamma$.

p-SAT(Γ) is a parameterization of the classical problem SAT(Γ). There are other interesting parameterizations of SAT(Γ), and we will see some later.

Example 3.3. The *weight* of an assignment $\mathcal{V}$ is the number of variables set to TRUE by $\mathcal{V}$. A circuit γ is *k-satisfiable*, for some $k \in \mathbb{N}$, if there is a satisfying assignment $\mathcal{V}$ of weight k for γ. The *weighted satisfiability problem* WSAT(Γ) for a class Γ of circuits asks whether a given circuit $\gamma \in \Gamma$ is k-satisfiable for a given k. We consider the following parameterization:

p-WSAT(Γ)	
Instance:	$\gamma \in \Gamma$ and $k \in \mathbb{N}$.
Parameter:	k.
Problem:	Decide if γ is k-satisfiable.

Definition 3.4. Let Σ be a finite alphabet and $\kappa : \Sigma^* \to \mathbb{N}$ a parameterization.

(1) An algorithm $\mathbb{A}$ with input alphabet Σ is an *fpt-algorithm with respect to* κ if there is a computable function $f : \mathbb{N} \to \mathbb{N}$ such that the running time of $\mathbb{A}$ on input x is

$$f(\kappa(x)) \cdot |x|^{O(1)}.$$

(2) A parameterized problem (Q, κ) is *fixed-parameter tractable* if there is an fpt-algorithm with respect to κ that decides Q.

FPT denotes the class of all fixed-parameter tractable problems.[2]

Example 3.5. p-SAT(CIRC) is fixed-parameter tractable.

Indeed, the obvious brute-force search algorithm decides if a circuit γ of size m with n variables is satisfiable in time $O(2^n \cdot m)$.

We leave it to the reader to show that p-VERTEX-COVER is also fixed-parameter tractable. On the other hand, p-WSAT(2-CNF) does not seem to be fixed-parameter tractable. We shall now introduce the theory to give evidence for this and other intractability results.

3.2. Reductions

Definition 3.6. Let (Q, κ) and (Q', κ') be parameterized problems over the alphabets Σ and Σ', respectively. An *fpt-reduction* (more precisely, *fpt many-one reduction*) from (Q, κ) to (Q', κ') is a mapping $R : \Sigma^* \to (\Sigma')^*$ such that:

(1) For all $x \in \Sigma^*$ we have $x \in Q \iff R(x) \in Q'$.

(2) R is computable by an fpt-algorithm with respect to κ.

(3) There is a computable function $g : \mathbb{N} \to \mathbb{N}$ such that $\kappa'(R(x)) \leq g(\kappa(x))$ for all $x \in \Sigma^*$.

We write $(Q, \kappa) \leq^{\text{fpt}} (Q', \kappa')$ if there is an fpt-reduction from (Q, κ) to (Q', κ'), and we write $(Q, \kappa) \equiv^{\text{fpt}} (Q', \kappa')$ if $(Q, \kappa) \leq^{\text{fpt}} (Q', \kappa')$ and $(Q', \kappa') \leq^{\text{fpt}} (Q, \kappa)$. We let $[(Q, \kappa)]^{\text{fpt}}$ be the class of parameterized problems fpt-reducible to (Q, κ), that is,

$$[(Q, \kappa)]^{\text{fpt}} = \{(Q', \kappa') \mid (Q', \kappa') \leq^{\text{fpt}} (Q, \kappa)\}.$$

For every class C of parameterized problems, we define C-hardness and C-completeness of a parameterized problem (Q, κ) in the usual way.

[2] The notion of fixed-parameter tractability we introduce here is known as "strongly uniform fixed-parameter tractability." The alternative notion "uniform fixed-parameter tractability" does not require the function f to be computable.

Example 3.7. Recall that an independent set in a graph is a set of pairwise non-adjacent vertices and consider the *parameterized independent set problem*:

p-INDEPENDENT-SET	
Instance:	A graph G and $k \in \mathbb{N}$.
Parameter:	k.
Problem:	Decide if G has an independent set of size k.

Then *p*-INDEPENDENT-SET $\leq^{\text{fpt}}$ *p*-WSAT(2-CNF), where 2-CNF denotes the class of all propositional formulas in 2-conjunctive normal form.

To see this, let $G = (V, E)$ be a graph. For every vertex $v \in V$ we introduce a propositional variable X_v whose intended meaning is "v belongs to the independent set". We let

$$\gamma = \bigwedge_{\{v,w\} \in E} (\neg X_v \vee \neg X_w).$$

Then α is k-satisfiable if and only if G has an independent set of size k. (There is one detail here that requires attention: If v is an isolated vertex of G, then the variable X_v does not occur in γ. Thus the claimed equivalence is true for graphs without isolated vertices. We leave it to the reader to reduce the problem for arbitrary graphs to graphs without isolated vertices.)

The converse also holds, that is,

$$p\text{-WSAT(2-CNF)} \leq^{\text{fpt}} p\text{-INDEPENDENT-SET},$$

but is much harder to prove [12]. By reversing the argument above, it is easy to show that *p*-WSAT(2-CNF$^-$) $\leq^{\text{fpt}}$ *p*-INDEPENDENT-SET, where 2-CNF$^-$ denotes the class of all 2-CNF-formulas in which only negative literals occur.

We also need a notion of parameterized Turing reductions:

Definition 3.8. Let (Q, κ) and (Q', κ') be parameterized problems over the alphabets Σ and Σ', respectively. An *fpt Turing reduction* from (Q, κ) to (Q', κ') is an algorithm $\mathbb{A}$ with an oracle to Q' such that:

(1) $\mathbb{A}$ decides (Q, κ).

(2) $\mathbb{A}$ is an fpt-algorithm with respect to κ.

(3) There is a computable function $g : \mathbb{N} \to \mathbb{N}$ such that for all oracle queries "$y \in Q'$?" posed by $\mathbb{A}$ on input x we have $\kappa'(y) \leq g(\kappa(x))$.

We write $(Q, \kappa) \leq^{\text{fpt-T}} (Q', \kappa')$ if there is an fpt Turing reduction from (Q, κ) to (Q', κ'), and we write $(Q, \kappa) \equiv^{\text{fpt-T}} (Q', \kappa')$ if $(Q, \kappa) \leq^{\text{fpt-T}} (Q', \kappa')$ and $(Q', \kappa') \leq^{\text{fpt-T}} (Q, \kappa)$.

3.3. The W-Hierarchy

Recall the definitions of the classes $\Gamma_{t,d}$ of propositional formulas.

Definition 3.9. (1) For $t \geq 1$, W[t] is the class of all parameterized problems fpt-reducible to a problem p-WSAT($\Gamma_{t,d}$) for some $d \geq 1$, that is,

$$\mathrm{W}[t] = \bigcup_{d \geq 1} [p\text{-WSAT}(\Gamma_{t,d})]^{\mathrm{fpt}}.$$

(2) W[SAT] is the class of all parameterized problems fpt-reducible to p-WSAT(PROP), that is,

$$\mathrm{W}[\mathrm{SAT}] = [p\text{-WSAT}(\mathrm{PROP})]^{\mathrm{fpt}}.$$

(3) W[P] is the class of all parameterized problems fpt-reducible to p-WSAT(CIRC), that is,

$$\mathrm{W}[\mathrm{P}] = [p\text{-WSAT}(\mathrm{CIRC})]^{\mathrm{fpt}}.$$

Observe that

$$\mathrm{FPT} \subseteq \mathrm{W}[1] \subseteq \mathrm{W}[2] \subseteq \cdots \subseteq \mathrm{W}[\mathrm{SAT}] \subseteq \mathrm{W}[\mathrm{P}].$$

One of the fundamental structural results of parameterized complexity theory is the following normalization theorem for the W-hierarchy. For $t, d \geq 1$ we let $\Gamma^{+}_{t,d}$ be the class of $\Gamma_{t,d}$-formulas in which all literals are positive (that is, no negation symbols occur) and $\Gamma^{-}_{t,d}$ be the class of $\Gamma_{t,d}$-formulas in which all literals are negative

Theorem 3.10 (Downey and Fellows [12, 11]).

(1) $\mathrm{W}[1] = [\mathrm{WSAT}(\Gamma^{-}_{1,2})]^{\mathrm{fpt}}$.

(2) For even $t \geq 2$, $\mathrm{W}[t] = [\mathrm{WSAT}(\Gamma^{+}_{t,1})]^{\mathrm{fpt}}$.

(3) For odd $t \geq 3$, $\mathrm{W}[t] = [\mathrm{WSAT}(\Gamma^{-}_{t,1})]^{\mathrm{fpt}}$.

Many natural parameterized problems are complete for the first two levels of the W-hierarchy. For example, p-INDEPENDENT-SET is complete for W[1] [11], and the parameterized dominating set problem is complete for W[2] [12].

3.4. W[P] and Limited Nondeterminism

We close this introductory section by presenting two results that establish a very clean connection between the class W[P] and limited nondeterminism [22, 16]. The first is a machine characterization of W[P]:

Theorem 3.11 ([2, 7]). *A parameterized problem (Q, κ) over the alphabet Σ is in* W[P] *if and only if there are computable functions $f, h : \mathbb{N} \to \mathbb{N}$, a polynomial $p(X)$, and a nondeterministic Turing machine $\mathbb{M}$ deciding Q such that for every input x on every run the machine $\mathbb{M}$:*

(1) performs at most $f(k) \cdot p(n)$ steps;

(2) performs at most $h(k) \cdot \log n$ nondeterministic steps.

Here $n = |x|$ and $k = \kappa(x)$.

Let $f : \mathbb{N} \to \mathbb{N}$. A problem $Q \subseteq \Sigma^*$ is in NP[f] if there is a polynomial p and a nondeterministic Turing machine $\mathbb{M}$ deciding Q such that for every input x on every run the machine $\mathbb{M}$

(1) performs at most $p(|x|)$ steps;

(2) performs at most $f(|x|)$ nondeterministic steps.

There is an obvious similarity between the characterization of W[P] given in Theorem 3.11 and the (classical) classes NP[f]. The next theorem establishes a formal connection:

Theorem 3.12 ([2]). *The following statements are equivalent:*

(1) FPT = W[P].

(2) There is a computable function $\iota : \mathbb{N} \to \mathbb{N}$ that is non-decreasing and unbounded such that PTIME = NP[$\iota(n) \cdot \log n$].

The techniques used to prove this result are similar to those introduced in the next section. Indeed, the direction (1) $\Longrightarrow$ (2) is an easy consequence of Theorem 4.4.

The connection between parameterized complexity and limited nondeterminism can be broadened if one considers *bounded parameterized complexity theory*, where some bound is put on the growth of the dependence of the running time of an fpt-algorithm on the parameter (see [15]).

4. The M-Hierarchy

4.1. A New Parameterization of the Satisfiability Problem

In the following, we will consider different parameterizations of the satisfiability problem Sat(CIRC). We denote the input circuit by γ, its size by m, and its number of variables by n. Without loss of generality we can always assume that $m \leq 2^n$, because if $m > 2^n$ we can easily decide if γ is satisfiable in time $m^{O(1)}$. However, in general m can still be much larger than n.

If we parameterize Sat(CIRC) by n, we obtain the fixed-parameter tractable problem p-Sat(CIRC). Let us now see what happens if we decrease the parameter. Specifically, let us consider the parameterizations $(\text{Sat(CIRC)}, \kappa_h)$, where

$$\kappa_h(\gamma) = \left\lceil \frac{n}{h(m)} \right\rceil$$

for computable functions $h : \mathbb{N} \to \mathbb{N}$. For constant $h \equiv 1$, κ_h is just our old parameterization p-Sat(CIRC) $\in$ FPT. At the other end of the scale, for $h(m) \geq m \geq n$ we have $\kappa_h(\gamma) = 1$, and essentially $(\text{Sat(CIRC)}, \kappa_h)$ is just the NP-complete unparameterized problem Sat(CIRC). But what happens if we consider functions between these two extremes?

If $h(m) \in o^{\text{eff}}(\log m)$, then $(\text{Sat(CIRC)}, \kappa_h)$ is still fixed-parameter tractable. (To see this, use that Sat(CIRC) is trivially solvable in time $m^{O(1)}$ for instances with $m \geq 2^n$.) If $h(m) \in \omega^{\text{eff}}(\log m)$ then for large circuits of size close to 2^n, but still $2^{o^{\text{eff}}(n)}$, the parameter is 1 and fixed-parameter tractability coincides with polynomial time computability. The most interesting range from the perspective of parameterized complexity is

$$h(m) \in \Theta(\log m).$$

These considerations motivate us to introduce the following parameterization of the satisfiability problem for every class Γ of circuits.

p-log-Sat(Γ)	
Instance:	$\gamma \in \Gamma$ of size m with n variables.
Parameter:	$\left\lceil \frac{n}{\log m} \right\rceil$.
Problem:	Decide if γ is satisfiable.

Obviously, *p-log*-Sat(Γ) is solvable in time

$$2^n \cdot m^{O(1)} \leq 2^{k \cdot \log m} \cdot m^{O(1)} = m^{k+O(1)},$$

where $k = \left\lceil \frac{n}{\log m} \right\rceil$ is the parameter. Intuitively it seems unlikely that the problem is fixed-parameter tractable.

To phrase our first result in its most general form, we introduce a simple closure property of classes of circuits: We call a class Γ *paddable* if for every $\gamma \in \Gamma$ and for every $m' \geq |\gamma|$ there is a circuit $\gamma' \in \Gamma$ such that $\mathrm{var}(\gamma') = \mathrm{var}(\gamma)$, the circuits γ and γ' are equivalent, and $m' \leq |\gamma'| \leq O(m')$. We call Γ *efficiently paddable* if, in addition, there is an algorithm that computes γ' for given γ and $m' \geq |\gamma|$ in time $(m')^{O(1)}$. Most natural classes of formulas and circuits are efficiently paddable, in particular all classes $\Gamma_{t,d}$ and the classes PROP and CIRC. For example, for the $\Gamma_{1,2}$-formula

$$\gamma = \bigwedge_{i=1}^{m} (\lambda_{i1} \vee \lambda_{i2}),$$

we can let $\lambda_{ij} = \lambda_{mj}$ for $m < i \leq m'$ and $j = 1, 2$, and

$$\gamma' = \bigwedge_{i=1}^{m'} (\lambda_{i1} \vee \lambda_{i2}).$$

Proposition 4.1 ([3, 10]). *Let Γ be an efficiently paddable class of circuits. Then*

$$p\text{-}log\text{-}\textsc{Sat}(\Gamma) \in \mathrm{FPT} \iff \textsc{Sat}(\Gamma) \in \mathrm{DTIME}(2^{o^{\mathrm{eff}}(n)} \cdot m^{O(1)}),$$

where $n = |\mathrm{var}(\gamma)|$ is the number of variables and $m = |\gamma|$ the size of the input circuit γ.

Proof: Suppose first that $p\text{-}log\text{-}\textsc{Sat}(\Gamma) \in \mathrm{FPT}$. Let $f : \mathbb{N} \to \mathbb{N}$ be a computable function and $\mathbb{A}$ an fpt-algorithm that decides $p\text{-}log\text{-}\textsc{Sat}(\Gamma)$ in time

$$f(k) \cdot m^{O(1)},$$

where $k = \lceil n / \log m \rceil$ is the parameter. Without loss of generality, we may assume that f is increasing and time constructible, which implies that $f(i) \leq j$ can be decided in time $O(j)$. Let $\iota : \mathbb{N} \to \mathbb{N}$ be defined by

$$\iota(n) = \max\Big(\{1\} \cup \{i \in \mathbb{N} \mid f(i) \leq n\}\Big).$$

Then ι is non-decreasing and unbounded, $f(\iota(n)) \leq n$ for all but finitely many n, and $\iota(n)$ can be computed in time $O(n^2)$.

We shall prove that $\textsc{Sat}(\Gamma) \in \mathrm{DTIME}(2^{O(n/\iota(n))} \cdot m^{O(1)})$.

Let $\gamma \in \Gamma$, $m = |\gamma|$ and $n = |\mathrm{var}(\gamma)|$. Assume first that $m \geq 2^{n/\iota(n)}$. Note that

$$k = \left\lceil \frac{n}{\log m} \right\rceil \leq \iota(n).$$

Thus $f(k) \leq f(\iota(n)) \leq n$, and we can simply decide $\gamma \in \text{SAT}(\Gamma)$ with the fpt-algorithm $\mathbb{A}$ in time

$$f(k) \cdot m^{O(1)} \leq n \cdot m^{O(1)} = m^{O(1)}.$$

Assume next that $m < 2^{n/\iota(n)}$. Let $m' = 2^{\lceil n/\iota(n) \rceil}$. Let $\gamma' \in \Gamma$ such that $\text{var}(\gamma') = \text{var}(\gamma)$, the circuits γ and γ' are equivalent, and $m' \leq |\gamma'| \leq O(m')$. Since Γ is efficiently paddable, such a γ' can be computed in time polynomial in m', that is, time $2^{O(n/\iota(n))}$. Let $k' = n/\log|\gamma'|$. Then $k' \leq \iota(n)$. We decide $\gamma' \in \text{SAT}(\Gamma)$ with the fpt-algorithm $\mathbb{A}$ in time

$$f(k') \cdot (m')^{O(1)} \leq n \cdot 2^{O(n/\iota(n))}.$$

This completes the proof of the forward direction.

Regarding the backward direction, let $\mathbb{B}$ be an algorithm solving $\text{SAT}(\Gamma)$ in $\text{DTIME}(2^{O(n/\iota(n))} \cdot m^{O(1)})$ for some computable function $\iota : \mathbb{N} \to \mathbb{N}$ that is non-decreasing and unbounded. Let f be a non-decreasing computable function with $f(\iota(n)) \geq 2^n$ for all $n \in \mathbb{N}$. We claim that

$$p\text{-}log\text{-}\text{SAT}(\Gamma) \in \text{DTIME}(f(k) \cdot m^{O(1)}).$$

Let $\gamma \in \Gamma$, $m = |\gamma|$, $n = |\text{var}(\gamma)|$, and $k = \lceil n/\log m \rceil$. If $m \geq 2^{n/\iota(n)}$ then algorithm $\mathbb{B}$ decides $\gamma \in \text{SAT}(\Gamma)$ in time $m^{O(1)}$. If $m < 2^{n/\iota(n)}$, then

$$k = \left\lceil \frac{n}{\log m} \right\rceil \geq \iota(n)$$

and thus $f(k) \geq 2^n$. Thus we can decide $\gamma \in \text{SAT}(\Gamma)$ by exhaustive search in time $O(f(k) \cdot m)$. □

In the following, we shall say that $\text{SAT}(\Gamma)$ is *subexponential (with respect to the number of variables)* if it is solvable in $\text{DTIME}(2^{o^{\text{eff}}(n)} \cdot m^{O(1)})$.

4.2. The M-Hierarchy

Motivated by Proposition 4.1, we define another hierarchy of parameterized complexity classes in anlogy to Definition 3.8:

Definition 4.2. (1) For every $t \geq 1$, we let $\text{M}[t] = \bigcup_{d \geq 1} [p\text{-}log\text{-}\text{SAT}(\Gamma_{t,d})]^{\text{fpt}}$.

(2) $\text{M}[\text{SAT}] = [p\text{-}log\text{-}\text{SAT}(\text{PROP})]^{\text{fpt}}$.

(3) $\text{M}[\text{P}] = [p\text{-}log\text{-}\text{SAT}(\text{CIRC})]^{\text{fpt}}$.

Then by Proposition 4.1:

Corollary 4.3. *(1) For $t \geq 1$,* $\mathrm{M}[t] = \mathrm{FPT}$ *if and only if* $\textsc{Sat}(\Gamma_{t,d})$ *is subexponential for all $d \geq 1$.*

(2) $\mathrm{M}[\mathrm{SAT}] = \mathrm{FPT}$ *if and only if* $\textsc{Sat}(\mathrm{PROP})$ *is subexponential.*

(3) $\mathrm{M}[\mathrm{P}] = \mathrm{FPT}$ *if and only if* $\textsc{Sat}(\mathrm{CIRC})$ *is subexponential.*

The following theorem is essentially due to Abrahamson, Downey and Fellows [1] (also see [13]).

Theorem 4.4. *For every $t \geq 1$,*

$$\mathrm{M}[t] \subseteq \mathrm{W}[t] \subseteq \mathrm{M}[t+1].$$

Furthermore, $\mathrm{M}[\mathrm{SAT}] = \mathrm{W}[\mathrm{SAT}]$ *and* $\mathrm{M}[\mathrm{P}] = \mathrm{W}[\mathrm{P}]$.

Proof: We first prove $\mathrm{M}[t] \subseteq \mathrm{W}[t]$. For simplicity, let us assume that t is odd. Fix $d \geq 1$ such that $t + d \geq 3$. We shall prove that

$$p\text{-}log\text{-}\textsc{Sat}(\Gamma_{t,d}) \leq^{\mathrm{fpt}} p\text{-}\mathrm{W}\textsc{Sat}(\Gamma_{t,d}). \tag{4.1}$$

Let $\gamma \in \Gamma_{t,d}$. We shall construct a $\Gamma_{t,d}$-formula β such that

$$\gamma \text{ is satisfiable} \iff \beta \text{ is } k\text{-satisfiable}. \tag{4.2}$$

Let $m = |\gamma|$, $n = |\mathrm{var}(\gamma)|$. To simplify the notation, let us assume that $\ell = \log m$ and $k = n/\log m$ are integers. Then $n = k \cdot \ell$. Let $\mathcal{X} = \mathrm{var}(\gamma)$, and let $\mathcal{X}_1, \ldots, \mathcal{X}_k$ be a partition of $\mathcal{X}$ into k sets of size ℓ.

For $1 \leq i \leq k$ and every subset $S \subseteq \mathcal{X}_i$, let Y_i^S be a new variable. Let $\mathcal{Y}_i$ be the set of all Y_i^S and $\mathcal{Y} = \bigcup_{i=1}^k \mathcal{Y}_i$. Call a truth value assignment for $\mathcal{Y}$ *good* if for $1 \leq i \leq k$ exactly one variable in $\mathcal{Y}_i$ is set to TRUE. There is a bijection f between the truth value assignments $\mathcal{V}$ for $\mathcal{X}$ and the good truth value assignments for $\mathcal{Y}$ defined by

$$f(\mathcal{V})(Y_i^S) = \textsc{true} \iff \forall X \in \mathcal{X}_i : \ (\mathcal{V}(X) = \textsc{true} \iff X \in S)$$

for all $\mathcal{V} : \mathcal{X} \to \{\textsc{true}, \textsc{false}\}$, $1 \leq i \leq k$, and $S \subseteq \mathcal{X}_i$.

Let β'' be the formula obtained from γ by replacing, for $1 \leq i \leq k$ and $X \in \mathcal{X}_i$, each occurrence of the literal X by the formula

$$\bigwedge_{S \subseteq \mathcal{X}_i \text{ with } X \notin S} \neg Y_i^S$$

and each occurrence of the literal $\neg X$ by the formula

$$\bigwedge_{S \subseteq \mathcal{X}_i \text{ with } X \in S} \neg Y_i^S .$$

Then an assignment $\mathcal{V} : \mathcal{X} \to \{\text{TRUE}, \text{FALSE}\}$ satisfies γ if and only if $f(\mathcal{V})$ satisfies β''. Thus γ is satisfiable if and only if β'' has a good assignment. Note that the size of each of the sets $\mathcal{Y}_i$ is $2^\ell = m$. Thus the size of β'' is polynomial in m. Moreover, β'' can easily be computed from γ in polynomial time.

β'' is not a $\Gamma_{t,d}$-formula: The transformation from γ to β'' has turned the small disjunctions $(\lambda_1 \vee \ldots \vee \lambda_d)$ on the bottom level of γ into formulas

$$\bigwedge_i \nu_{1i} \vee \ldots \vee \bigwedge_i \nu_{di}.$$

Applying the distributive law to all these subformulas turns them into big conjunctions of disjunctions of at most d literals, and since t is odd, it turns the whole formula β'' into a $\Gamma_{t,d}$-formula β'. Since d is fixed, the size only increases polynomially, and β' can be computed from β'' in polynomial time. And we still have: γ is satisfiable if and only if β' has a good assignment.

All that remains to do is add a subformula stating that all assignments of weight k are good. We let

$$\alpha = \bigwedge_{i=1}^{k} \bigwedge_{\substack{S,T \subseteq \mathcal{X}_i \\ S \neq T}} (\neg Y_i^S \vee \neg Y_i^T)$$

and $\beta = \alpha \wedge \beta'$. Then β is (equivalent to) a $\Gamma_{t,d}$-formula that satisfies (4.2).

Next, we prove $\text{W}[t] \subseteq \text{M}[t+1]$. For simplicity, let us assume again that t is odd. Let $d = 2$ if $t = 1$ and $d = 1$ otherwise. Recall that by Theorem 3.10, $\text{WSAT}(\Gamma^-_{t,d})$ is $\text{W}[t]$-complete. We shall prove that

$$p\text{-WSAT}(\Gamma^-_{t,d}) \leq^{\text{fpt}} p\text{-}log\text{-SAT}(\Gamma_{t+1,1}). \tag{4.3}$$

We simply reverse the idea of the proof that $\text{M}[t] \subseteq \text{W}[t]$.

Let $\beta \in \Gamma^-_{t,d}$ and $k \geq 1$, say,

$$\beta = \bigwedge_{i_1 \in I_1} \bigvee_{i_2 \in I_2} \ldots \bigwedge_{i_t \in I_t} \delta(i_1, \ldots, i_t), \tag{4.4}$$

where each $\delta(i_1, \ldots, i_t)$ is a disjunction of at most d negative literals. Let $n = |\text{var}(\beta)|$ and $\ell = \log n$, and let us assume again that ℓ is an integer. Furthermore, we assume that the variables of β are indexed with subsets of $\{1, \ldots, \ell\}$, or more precisely, that

$$\text{var}(\beta) = \mathcal{Y} = \{Y^S \mid S \subseteq \{1, \ldots, \ell\}\}.$$

For $1 \leq i \leq k$ and $1 \leq j \leq \ell$, let X_{ij} be a new variable. As above, let $\mathcal{X}_i = \{X_{ij} \mid 1 \leq j \leq \ell\}$ and $\mathcal{X} = \bigcup_{i=1}^{k} \mathcal{X}_i$. The idea is that every assignment to the variables

in $\mathcal{X}_i$ corresponds to a subset $S_i \subseteq \{1, \ldots, \ell\}$ and hence to a variable Y^{S_i}. Thus an assignment to all variables in $\mathcal{X}$ corresponds to a subset $\{Y^{S_1}, \ldots, Y^{S_k}\} \subseteq \mathcal{Y}$ and hence to an assignment to the variables in $\mathcal{Y}$ of weight at most k ("at most" because the S_i are not necessarily distinct).

Formally, let g be the following mapping from the assignments for $\mathcal{X}$ to the assignments for $\mathcal{Y}$ of weight at most k: For every $\mathcal{V} : \mathcal{X} \to \{\text{TRUE}, \text{FALSE}\}$, we let $g(\mathcal{V}) : \mathcal{Y} \to \{\text{TRUE}, \text{FALSE}\}$ the assignment that sets $Y^{S_1}, \ldots, Y^{S_k}$ to TRUE and all other variables to FALSE, where for $1 \leq i \leq k$

$$S_i = \{j \mid \mathcal{V}(X_{ij}) = \text{TRUE}\}.$$

Let γ'' be the formula obtained from β by replacing each literal $\neg Y^S$ by the subformula

$$\chi_S = \bigwedge_{i=1}^{k} \Big(\bigvee_{j \in S} \neg X_{ij} \vee \bigvee_{j \in \{1,\ldots,\ell\} \setminus S} X_{ij} \Big).$$

(Remember that all literals in β are negative.) Then for every assignment $\mathcal{V} : \mathcal{X} \to \{\text{TRUE}, \text{FALSE}\}$,

$$\mathcal{V} \text{ satisfies } \gamma'' \iff g(\mathcal{V}) \text{ satisfies } \beta.$$

The translation from β to γ'' turns every $\delta = \delta(i_1, \ldots, i_t)$ in (4.4) into a disjunction δ' of at most d formulas χ_S. Say,

$$\delta' = (\chi_{S_1} \vee \ldots \vee \chi_{S_d})$$

By applying the distributive law, this formula can be turned into a conjunction χ of k^d disjunctions of $d \cdot \ell$ literals. Applying this operation to every δ' in γ'', we obtain an equivalent $\Gamma_{t+1,1}$-formula γ'. Then for every assignment $\mathcal{V} : \mathcal{X} \to \{\text{TRUE}, \text{FALSE}\}$,

$$\mathcal{V} \text{ satisfies } \gamma' \iff g(\mathcal{V}) \text{ satisfies } \beta.$$

This almost completes the proof. The only problem that remains to be solved is that not all assignments $g(\mathcal{V})$ have weight exactly k, because some of the induced S_i may be identical. Let

$$\alpha' = \bigwedge_{1 \leq i < i' \leq k} \bigvee_{j=1}^{\ell} \neg(X_{ij} \leftrightarrow X_{i'j}).$$

Then for every $\mathcal{V} : \mathcal{X} \to \{\text{TRUE}, \text{FALSE}\}$ that satisfies α', the assignment $g(\mathcal{V})$ has weight exactly k. Thus g induces a mapping from the assignments for $\mathcal{X}$ that satisfy α' onto the weight k assignments for $\mathcal{Y}$. Note that α' is equivalent to a

$\Gamma_{2,1}$-formula α of size $O(k^2 \cdot 2^{2\ell}) = O(k^2 \cdot n^2)$. Furthermore, given k, n, such a formula α can be computed in time polynomial in k and n.

We let $\gamma = \alpha \wedge \gamma'$. Then γ is satisfiable if and only if β is k-satisfiable. The size m of γ is polynomial in the size of β, and the number of variables is $k \cdot \ell$, where $\ell = \log n \leq \log m$. By adding dummy variables (to the outermost conjunction of γ) we can adjust the number of variables in such a way that $k = \lceil |\mathrm{var}(\gamma)| / \log m \rceil$.

It remains to prove M[SAT] = W[SAT] and M[P] = W[P]. We can simply carry out the preceding constructions without worrying about the form of the resulting formulas. □

Corollary 4.5. *Let $t, d \geq 1$.*

(1) If W[t] = FPT *then* SAT($\Gamma_{t,d}$) *is subexponential.*

(2) If SAT($\Gamma_{t+1,1}$) *is subexponential then* W[t] = FPT.

By a more refined argument based on the same idea, Chen et al. [4] strengthened part (1) of the corollary as follows:

Theorem 4.6 ([4]). *Let $t, d \geq 1$ such that $t + d \geq 3$. If*

$$\mathrm{WSAT}(\Gamma_{t,d}) \in \mathrm{DTIME}(f(k) \cdot n^{o^{\mathrm{eff}}(k)} \cdot m^{O(1)})$$

for some computable function f, then SAT($\Gamma_{t,d}$) *is subexponential.*

In [5], this has further been strengthened by restricting the range of values k for which the hypothesis is needed.

Let us briefly return to the connections between W[P] and limited nondeterminism. Recall Theorem 3.12. We encourage the reader to prove the direction (1) $\Longrightarrow$ (2); it follows easily from W[P] = M[P]. (The converse direction of the theorem is also not hard to prove.) Let us summarize our three characterizations of W[P] vs FPT in a corollary:

Corollary 4.7. *The following three statements are equivalent:*

(1) W[P] = FPT.

(2) SAT(CIRC) *is subexponential.*

(3) PTIME = NP[$\iota(n) \cdot \log n$] *for some computable function $\iota : \mathbb{N} \to \mathbb{N}$ that is non-decreasing and unbounded.*

5. M[1] and Miniaturized Problems

Originally, the class M[1] was defined in terms of so-called *parameterized miniaturizations* of NP-complete problems [10]. Let $Q \subseteq \Sigma^*$ be any decision problem. We define:

p-mini-Q	
Instance:	$x \in \Sigma^*$ and $m \in \mathbb{N}$ in unary.
Parameter:	$\left\lceil \frac{\lvert x\rvert}{\log m} \right\rceil$.
Problem:	Decide if $x \in Q$.

We call *p-mini-Q* the "miniaturization" of Q, because if we assume the parameter $k = \lceil |x|/\log m \rceil$ to be small, then the size $|x| = \lfloor k \cdot \log m \rfloor$ of the actual instance is very small compared to the "padded size" $|x| + m$. There is an equivalent way of formulating the problem making this explicit:

Instance:	$x \in \Sigma^*$ and $k, m \in \mathbb{N}$ in unary such that $\lvert x\rvert = \lfloor k \cdot \log m \rfloor$.
Parameter:	k.
Problem:	Decide if $x \in Q$.

The main reason that we are interested in these strange problems is the following equivalence:

Proposition 5.1. *Let Σ be a finite alphabet and $Q \subseteq \Sigma^*$. Then*

$$\textit{p-mini-}Q \in \mathrm{FPT} \iff Q \in \mathrm{DTIME}(2^{o^{\mathrm{eff}}(n)}),$$

where $n = |x|$ denotes the length of the instance x of Q.

We skip the proof, which is very similar to the proof of Proposition 4.1.

The main combinatorial tool in the development of a M[1]-completeness theory is the Sparsification Lemma due to Impagliazzo, Paturi, and Zane [19]. The lemma says that the satisfiability problem for d-CNF-formulas can be reduced to the satisfiability problem for d-CNF-formulas whose size is linear in the number of variables by a suitable reduction that preserves subexponential time solvability.

Lemma 5.2 (Sparsification Lemma [19]). *Let $d \geq 2$. There is a computable function $f : \mathbb{N} \to \mathbb{N}$ such that for every $k \in \mathbb{N}$ and every formula $\gamma \in d$-CNF with $n = |\mathrm{var}(\gamma)|$ variables there is a $\Delta_{2,d}$-formula*

$$\beta = \bigvee_{i=1}^{p} \beta_i$$

such that:

(1) β *is equivalent to* γ*,*

(2) $p \leq 2^{n/k}$*,*

(3) $|\beta_i| \leq f(k) \cdot n$ *for* $1 \leq i \leq p$*.*

Furthermore, there is an algorithm that, given γ *and* k*, computes* β *in time* $2^{n/k} \cdot |\gamma|^{O(1)}$*.*

The idea of using the Sparsification Lemma in this context goes back to Cai and Juedes [3].

Theorem 5.3 ([3, 10]). *p-mini-*Sat(3-CNF) *is* M[1]*-complete under fpt Turing reductions.*

Proof: To prove that *p-mini-*Sat(3-CNF) $\in$ M[1], we show that

$$\textit{p-mini}\text{-Sat(3-CNF)} \leq^{\text{fpt}} \textit{p-log}\text{-Sat(3-CNF)}.$$

Let (γ, m) be an instance of *p-mini-*Sat(3-CNF) and $k = |\gamma|/\log m$. By padding γ we obtain an equivalent formula γ' such that $\text{var}(\gamma') = \text{var}(\gamma)$ and $m' = |\gamma'| \geq m$. Then

$$k' = \frac{|\text{var}(\gamma')|}{\log m'} \leq \frac{|\text{var}(\gamma)|}{\log m} \leq \frac{|\gamma|}{\log m} = k.$$

Thus $(\gamma, m) \mapsto \gamma'$ is an fpt-reduction from *p-mini-*Sat(3-CNF) to *p-log-*Sat(3-CNF).

To prove hardness, we show that

$$\textit{p-log}\text{-Sat}(d\text{-CNF}) \leq^{\text{ftp-T}} \textit{p-mini}\text{-Sat(3-CNF)}$$

for all $d \geq 3$. Fix $d \geq 3$.

Let $\gamma \in d$-CNF. Let $m = |\gamma|$, $n = |\text{var}(\gamma)|$, and $k = \lceil n/\log m \rceil$.

Choose $f : \mathbb{N} \to \mathbb{N}$ (depending on d) and $\beta = \bigvee_{i=1}^{p} \beta_i$ (depending on γ and k) according to the Sparsification Lemma 5.2. Since

$$2^{n/k} = 2^{\frac{n}{\lceil n/\log m \rceil}} \leq m,$$

we have $p \leq m$, and β can be computed in time $m^{O(1)}$. Let $1 \leq i \leq p$. Since β_i has at most $f(k) \cdot n$ clauses, there is a 3-CNF-formula β_i' with at most $f(k) \cdot d \cdot n$ variables and of length $|\beta_i'| \in O(f(k) \cdot d \cdot n)$ such that

$$\beta_i \text{ is satisfiable} \iff \beta_i' \text{ is satisfiable.}$$

Thus γ is satisfiable if and only if there exists an $i, 1 \leq i \leq p$, such that β_i' is satisfiable.

For $1 \leq i \leq p$ we have

$$k'_i = \left\lceil \frac{|\mathrm{var}(\beta'_i)|}{\log m} \right\rceil = O\left(\frac{f(k) \cdot d \cdot n}{\log m} \right) = O(f(k) \cdot d \cdot k).$$

The desired Turing reduction decides if $\gamma \in$ Sat(d-CNF) by querying the instances (β'_i, m), for $1 \leq i \leq p$, of p-*mini*-Sat(3-CNF). □

Corollary 5.4. *p-log-*Sat(3-CNF) *is* M[1]*-complete under ftp Turing reductions.*

Proof: We have noted in the proof of Theorem 5.3 that p-*mini*-Sat(3-CNF) is fpt-reducible to p-*log*-Sat(3-CNF). □

Polynomial time reductions between problems do not automatically give fpt-reductions between their miniaturizations. Let us a call a polynomial time reduction R from a problem $Q \subseteq \Sigma^*$ to a problem $Q' \subseteq (\Sigma')^*$ *size preserving* if for all $x \in \Sigma^*$ we have $|R(x)| \in O(|x|)$.

Lemma 5.5. *Let $Q \subseteq \Sigma^*$ and $Q' \subseteq (\Sigma')^*$ such that there is a size preserving polynomial time reduction from from Q to Q'. Then there is an fpt-reduction from p-mini-Q to p-mini-Q'.*

Proof: If R is a size-preserving polynomial time reduction from Q to Q' then $(x, m) \mapsto (R(x), m)$ defines an fpt-reduction from p-*mini*-Q to p-*mini*-Q'. □

Corollary 5.6. *The following problems are* M[1]*-complete under fpt Turing reductions:*

*(1) p-mini-d-*Colorability *for every $d \geq 3$,*

*(2) p-mini-*Sat(d-CNF) *for every $d \geq 3$ and p-mini-*Sat(CIRC)*.*

Proof: The standard polynomial time reductions between d-Colorability and Sat(CIRC) and Sat(3-CNF) are size preserving. □

Lemma 5.7. *There is a size preserving polynomial time reduction from* WSat(CIRC) *to* Sat(CIRC)*.*

We skip the proof. The main idea is to use a linear size circuit to count the number of variables set to true.

Corollary 5.8. *The following problems are* M[1]*-complete under fpt Turing reductions:*

(1) *p-mini*-Independent-Set,

(2) *p-mini*-Vertex-Cover,

(3) *p-mini*-WSat(d-CNF) *for every* $d \geq 2$ *and p-mini*-WSat(CIRC).

Proof: The reductions from Sat(3-CNF) to Independent-Set, from Independent-Set to Vertex-Cover and vice versa, from Independent-Set to WSat(2-CNF), from WSat(2-CNF) to *p-mini*-WSat(CIRC) are all size preserving. Thus the equivalence of the problems here and in Corollary 5.6 follows from Lemma 5.7. □

It is an open problem if the miniaturization *p-mini*-Short-NTM-Halt of the following problem Short-NTM-Halt is M[1]-complete. The input Turing machine is supposed to have just one work tape (or a fixed number), but may have an arbitrary alphabet. The parameterization of this problem by n is known to be W[1]-complete [12].

Short-NTM-Halt	
Instance:	A nondeterministic Turing machine M and an $n \in \mathbb{N}$ in unary.
Problem:	Decide if M, started with the empty tape, halts in at most n steps.

Note that the standard reduction between Clique and Independent-Set is not size preserving. Actually, we have:

Corollary 5.9. *p-mini*-Clique $\in$ FPT.

Proof: Observe that Clique $\in$ DTIME($n^{O(\sqrt{n})}$), where n is the size of the input. The reason is that a clique of size ℓ has $\Omega(\ell^2)$ edges and thus can only exist in a graph of size $\Omega(\ell^2)$.

By Proposition 5.1, this implies that *p-mini*-Clique $\in$ FPT. □

Corollary 5.9 highlights how sensitive the whole theory is to the specific encoding of the input and our "size measure". For example, for graph problems it would also be natural to define the "size" of an instance to be the number of vertices. Then, obviously, there are "size"-preserving reductions between Clique and Independent-Set. To investigate the role of size measures, we define a *size measure* on Σ^* to be a polynomial time computable function $\nu : \Sigma^* \to \mathbb{N}$. Of course

a size measure is just another parameterization. For now, we use a different term and different symbols to avoid confusion between the two. We will discuss the relation between size measures and parameterizations below.

Obviously, the actual input size, that is, $\nu(x) = |x|$ is a size measure, which we call the *standard size measure*. Other natural size measures are the number of vertices of a graph, that is,

$$\nu_{\text{vert}}(x) = \begin{cases} |V| & \text{if } x \text{ is the encoding of a graph } (V, E), \\ |x| & \text{otherwise,} \end{cases}$$

and the number of variables of a formula or circuit, that is,

$$\nu_{\text{var}}(x) = \begin{cases} |\text{var}(\gamma)| & \text{if } x \text{ is the encoding of a circuit } \gamma, \\ |x| & \text{otherwise,} \end{cases}$$

We let

p-mini[ν]-Q	
Instance:	$x \in \Sigma^*$ and $m \in \mathbb{N}$ in unary.
Parameter:	$\left\lceil \frac{\nu(x)}{\log m} \right\rceil$.
Problem:	Decide if $x \in Q$.

By essentially the same proof as that of Propositions 4.1 and 5.1, we obtain the following slightly more general result.

Proposition 5.10. *Let Σ be a finite alphabet and $Q \subseteq \Sigma^*$. Then*

$$\textit{p-mini}[\nu]\text{-}Q \in \text{FPT} \iff Q \in \text{DTIME}(2^{o^{\text{eff}}(\nu(x))} \cdot |x|^{O(1)}).$$

By using the Sparsification Lemma, it can be proved that:

$$\begin{aligned}
\textit{p-mini}[\nu_{var}]\text{-Sat}(d\text{-CNF}) &\equiv^{\text{fpt-T}} \textit{p-mini}\text{-Sat}(d\text{-CNF}) \\
\textit{p-mini}[\nu_{var}]\text{-WSat}(d\text{-CNF}) &\equiv^{\text{fpt-T}} \textit{p-mini}\text{-WSat}(d\text{-CNF}) \\
\textit{p-mini}[\nu_{vert}]\text{-}d\text{-Colorability} &\equiv^{\text{fpt-T}} \textit{p-mini-}d\text{-Colorability} \\
\textit{p-mini}[\nu_{vert}]\text{-Independent-Set} &\equiv^{\text{fpt-T}} \textit{p-mini}\text{-Independent-Set} \\
\textit{p-mini}[\nu_{vert}]\text{-Vertex-Cover} &\equiv^{\text{fpt-T}} \textit{p-mini}\text{-Vertex-Cover}.
\end{aligned}$$

Furthermore, we clearly have

$$\textit{p-mini}[\nu_{vert}]\text{-Clique} \equiv^{\text{fpt-T}} \textit{p-mini}[\nu_{vert}]\text{-Independent-Set}.$$

Thus, by Corollaries 5.6 and 5.8, all these problems are M[1]-complete. It is not known if *p-mini*[ν_{var}]-SAT(CIRC) or just *p-mini*[ν_{var}]-SAT(CNF) is reducible to *p-mini*-SAT(CIRC).

Let us re-iterate that a size measure and a parameterization are really the same thing (though introduced with different intentions). This becomes most obvious for the problems *p*-SAT(Γ), whose parameterization is just the size measure ν_{var} for SAT(Γ). Proposition 5.10 can be read as stating that *p-mini*[ν]-$Q \in$ FPT if and only if the parameterized problem (Q, ν) can be solved by a *subexponential fpt-algorithm*, that is, an fpt-algorithm whose running time is $2^{o^{\text{eff}}(k)} \cdot n^{O(1)}$, where k is the parameter and n the input size.

A starting point for the whole theory was the question of whether the parameterized vertex cover problem *p*-VERTEX-COVER (cf. Example 3.1), which is easily seen to be solvable in time $O(2^k \cdot |G|)$, has a subexponential fpt-algorithm. Note that the parameterization of *p*-VERTEX-COVER is *not* the same as the size measure ν_{vert}. Nevertheless, it can be proved:

Theorem 5.11 ([3, 10]). *p*-VERTEX-COVER *has a subexponential fpt-algorithm if and only if* M[1] = FPT.

6. Miniaturizations of Problems in SNP

There is a more general principle behind the results of the previous section, which becomes apparent if we look at the *syntactic form* of the problems considered there: They all belong to the syntactically defined complexity class SNP [23]. In this section, we shall prove that essentially, the miniaturizations of all problems in SNP are in M[1]. Some care needs to be taken with regards to the size measure.

Let us first recall the definition of the class SNP. Instances of problems in SNP are *relational structures* such as graphs. Propositional formulas or circuits can also be encoded by relational structures. A problem is in SNP if it is *definable* by a formula φ of second-order logic of the form

$$\exists X_1 \ldots \exists X_k \, \forall y_1 \ldots \forall y_\ell \, \psi(X_1, \ldots, X_k, y_1, \ldots, y_\ell). \tag{6.1}$$

Here $X_1, \ldots, X_k$ are *relation variables*, each with a prescribed arity, which range over relations over the universe of the input structure. $y_1, \ldots, y_\ell$ are *individual variables*, ranging over elements of the input structure. $\psi(X_1, \ldots, X_k, y_1, \ldots, y_\ell)$ is a *quantifier free formula*, that is, a Boolean combination of *atomic formulas* of the form $Rz_1 \ldots z_r$ or $z_1 = z_2$, where $z_1, \ldots, z_r \in \{y_1, \ldots, y_\ell\}$ and R is either one of the relation variables $X_1, \ldots, X_k$ or a relation symbol representing one of the relations of the structure (such as the edge relation of a graph). $Rz_1 \ldots z_r$ is true

in a structure under some interpretation of the variables if the tuple $(a_1, \ldots, a_r)$ of elements interpreting the individual variables $(z_1, \ldots, z_r)$ is contained in the relation interpreting R. Then the meaning of the whole formula is defined inductively using the usual rules for Boolean connectives and quantifiers.

For a structure A we write $A \models \varphi$ if φ holds in A. We can associate the following problem with φ:

D_φ	
Instance:	Structure A.
Problem:	Decide if $A \models \varphi$.

Slightly abusing notation, we use SNP to denote both the class of formulas of the form (6.1) and the class of all problems D_φ, where φ is a formula of the form (6.1).

Example 6.1. Let $d \geq 1$. The following SNP-formula χ states that a graph is d-colorable:

$$\underbrace{\exists X_1 \ldots \exists X_d}_{\substack{X_i \text{ is the set of} \\ \text{elements of color } i}} \forall x \forall y \Big(\underbrace{\bigvee_{i=1}^{d} X_i x \wedge \bigwedge_{1 \leq i < j \leq d} \neg(X_i x \wedge X_j x)}_{\text{“Each element has exactly one color.”}} \wedge \underbrace{\bigwedge_{1 \leq i \leq d} (Exy \rightarrow \neg(X_i x \wedge X_i y))}_{\text{“Adjacent elements do not have the same color.”}} \Big).$$

Thus $\mathrm{D}_\chi = d$-Colorability and hence d-Colorability $\in$ SNP.

An SNP-formula as in (6.1) is *monadic* if the relation variables $X_1, \ldots, X_k$ are all unary, that is, range over subsets of the structure. MSNP denotes the class of all monadic SNP-formulas and at the same time the class of all problems defined by such formulas. For example, the formula in Example 6.1 is monadic, and thus d-Colorability $\in$ MSNP for all $d \geq 1$. It is also not hard to see that Sat(d-CNF) $\in$ MSNP for all $d \geq 1$.

We generalize the size measures ν_{vert} and ν_{var} to arbitrary input structures by letting

$$\nu_{\mathrm{elt}}(x) = \begin{cases} n & \text{if } x \text{ is the encoding of a structure } A \text{ with } n \text{ elements,} \\ |x| & \text{otherwise,} \end{cases}$$

Then on graphs, $\nu_{\mathrm{elt}} = \nu_{\mathrm{vert}}$, and on d-CNF-formulas (if represented by structures in a standard way), $\nu_{\mathrm{elt}} = \nu_{\mathrm{var}}$.

Proposition 6.2. *For any $Q \in$ MSNP, the miniaturized problem p-mini$[\nu_{elt}]$-Q is contained in the closure of* M[1] *under fpt Turing reductions.*

It shown in [6] that for every problem $Q \in$ MSNP the miniaturized problem *p-mini*$[\nu_{elt}]$-Q is contained in W[1].

Problems such as Independent-Set or Vertex-Cover, at least if represented naturally, are not in SNP simply because the problem instances are not just structures (graphs), but pairs consisting of graphs and natural numbers. For such problems, we define a variant of SNP: Instead of formulas (6.1) we consider formulas $\varphi(X_0)$ of the form

$$\exists X_1 \ldots \exists X_k \; \forall y_1 \ldots \forall y_\ell \; \psi(X_0, X_1, \ldots, X_k, y_1, \ldots, y_\ell), \tag{6.2}$$

which have one additional relation variable occurring freely. Say, X_0 is s-ary. For a structure A and an s-ary relation S on A we write $A \models \varphi(S)$ if φ holds in A if X_0 is interpreted by S. We can associate the following problem with $\varphi(X_0)$:

WD_φ	
Instance:	A structure A and $k \in \mathbb{N}$.
Problem:	Decide if there is an s-ary relation S on A of size $\lvert S\rvert = k$ such that $A \models \varphi(S)$.

We use W-SNP to denote the class of all problems WD_φ, where φ is an SNP-formula of the form (6.2), and W-MSNP to denote the subclass of all problems WD_φ, where φ is an MSNP-formula.

Example 6.3. The following formula witnesses that Independent-Set $\in$ W-MSNP

$$\forall y_1 \forall y_2 \Big((X_0 y_1 \wedge X_0 y_2) \rightarrow \neg E y_1 y_2 \Big),$$

where the binary relation symbol E represents the edge relation of the input graph.

Similarly, it can be shown that Vertex-Cover, Clique, and WSat(d-CNF) for $d \geq 1$ are in W-MSNP.

Proposition 6.4. *For every problem $Q \in$ W-MSNP, the miniaturized problem p-mini$[\nu_{elt}]$-Q is contained in the closure of* M[1] *under fpt Turing reductions.*

Propositions 6.2 and 6.4 can be generalized to arbitrary (not necessarily monadic) SNP-formulas, but only under an unnatural size measure. For $r \geq 1$, let

$$\nu^r_{\mathrm{elt}}(x) = \begin{cases} n^r & \text{if } x \text{ is the encoding of a structure } A \text{ with } n \text{ elements,} \\ |x| & \text{otherwise,} \end{cases}$$

Call a formula of the form (6.1) or (6.2) *r-ary* if the maximum arity of the relations $(X_0), X_1, \ldots, X_k$ is r. Observe that monadic formulas are 1-ary. Propositions 6.2 and 6.4 generalize to r-ary formulas for every $r \geq 1$, but only under the size measure ν^r_{elt}.

For a thorough discussion of miniaturized problems (in particular syntactically defined problems such as those in SNP and W-SNP) under various size measures we refer the reader to [6].

7. The Exponential Time Hypothesis

We are now ready to apply the results of the previous sections in a more conventional setting.

In this section, we assume that d-CNF-formulas contain no repeated clauses and are thus of size $m = O(n^d)$ (for fixed d).[3] In particular, for every computable function $f(n) \in \Omega(\log n)$, this implies that

$$\textsc{Sat}(d\text{-CNF}) \in \text{DTIME}(2^{O(f(n))}) \Longleftrightarrow \textsc{Sat}(d\text{-CNF}) \in \text{DTIME}(2^{O(f(n))} \cdot m^{O(1)}).$$

We are concerned here with the "effective" version of the exponential time hypothesis:

$$\textsc{Sat}(3\text{-CNF}) \notin \text{DTIME}(2^{o^{\text{eff}}(n)}) \qquad \text{(ETH)}$$

Recall that by Proposition 5.1 and Corollary 5.6 we have

$$\text{(ETH)} \iff \text{M[1]} \neq \text{FPT}.$$

We say that a problem $Q \subseteq \Sigma^*$ is *subexponential with respect to a size measure* $\nu : \Sigma^* \to \mathbb{N}$ if there is an algorithm deciding $x \in Q$ in time

$$2^{o^{\text{eff}}(\nu(x))} \cdot |x|^{O(1)}.$$

The negation of (ETH) will be denoted by ¬(ETH). The results of the previous two sections yield the following two corollaries:

Corollary 7.1 ([19]). *¬(ETH) is equivalent to either of the following problems being subexponential:*

(1) $\textsc{Sat}(d\text{-CNF})$ *for* $d \geq 3$ *with respect to the standard size measure and* ν_{var}.

(2) $\textsc{WSat}(d\text{-CNF})$ *for* $d \geq 2$ *with respect to the standard size measure and* ν_{var}.

[3] Some care needs to be taken with this assumption because the proofs of some of the earlier results involve padding arguments that are no longer available if we make the assumption. The reader may be assured that we take care here.

(3) SAT(CIRC) *and* WSAT(CIRC) *with respect to the standard size measure.*

(4) d-COLORABILITY *for* $d \geq 3$ *with respect to the standard size measure and* ν_{vert}.

(5) INDEPENDENT-SET *with respect to the standard size measure and* ν_{vert}.

(6) CLIQUE *with respect to* ν_{vert}.

(7) VERTEX-COVER *with respect to the standard size measure and* ν_{vert}.

It can further be proved that INDEPENDENT-SET restricted to graphs of degree at most 3 is equivalent to INDEPENDENT-SET on arbitrary graphs with respect to subexponential solvability [21].

Corollary 7.2 ([19]). *¬(ETH) implies that all problems in MSNP and W-MSNP are subexponential with respect to* ν_{elt}.

As Propositions 6.2 and 6.4, Corollary 7.2 can be generalized from monadic to arbitrary SNP-problems for the size measures ν^r_{elt}.

The fixed-parameter tractable Turing reductions between the miniaturized problems that we gave in Section 5 can be translated to "subexponential" reductions between the corresponding classical problems (so-called *SERF-reductions* as defined in [19]), and it follows that the problems mentioned in Corollary 7.1 are complete for MSNP or W-MSNP, respectively, under such reductions.

In view of the previous section, there is a natural generalization of (ETH) to $t \geq 1$:

$$\exists d \geq 1 : \ \text{SAT}(\Gamma_{t,d}) \notin \text{DTIME}(2^{o^{\text{eff}}(n)} \cdot m^{O(1)}) \qquad (\text{ETH}_t)$$

Then (ETH) = (ETH_1). By Corollary 4.3, for all $t \geq 1$ we have

$$(\text{ETH}_t) \iff \text{M}[t] = \text{FPT}.$$

Not much is known about (ETH_t) for $t \geq 2$. As a matter of fact, it is not even known if

$$\text{SAT}(\text{CNF}) \in \text{DTIME}(2^{\varepsilon \cdot n} \cdot m^{O(1)}) \qquad (7.1)$$

for some constant $\varepsilon < 1$. Suppose that (ETH) holds. Then for every $d \geq 3$ there exists a positive constant

$$\varepsilon_d = \inf \left\{ \varepsilon > 0 \;\middle|\; \text{SAT}(d\text{-CNF}) \in \text{DTIME}(2^{\varepsilon \cdot n} \cdot m^{O(1)}) \right\}.$$

It is known that $\varepsilon_d < 1$ for all $d \geq 3$ and that the sequence is $(\varepsilon_d)_{d \geq 3}$ is non-decreasing and not ultimately constant [18]. The latter is a fairly deep result; its proof combines the Sparsification Lemma [19] with techniques for the SAT(d-CNF) algorithm due to Paturi, Pudlak, Saks, and Zane [24].

It is an open problem if $\lim_{d \to \infty} \varepsilon_d = 1$. Of course, if $\lim_{d \to \infty} \varepsilon_d = 1$ then there is no constant $\varepsilon < 1$ satisfying (7.1). It is not known if the converse of this statement also holds.

8. Open Problems

First of all, it would be very nice to prove that the W-hierarchy and the M-hierarchy coincide on each level. In particular, if M[1] = W[1] then the exponential time hypothesis would be equivalent to FPT $\neq$ W[1], which we may interpret as new evidence for the exponential time hypothesis. While the question of whether M[1] = W[1] has received a lot of attention in the parameterized complexity community, the question of whether M[t] = W[t] for $t \geq 2$ has not been looked at very intensely (and may in fact be easier, as the classes get more robust on higher levels). It is also conceivable that M[$t+1$] = W[t] for $t \geq 1$.

A second interesting question is whether the M[1]-completeness of the problem *p-log*-SAT($\Gamma_{1,3}$) (which may be viewed as a normal form result for M[1]) has analogues for higher levels of the hierarchy. The result one would hope for is that *p-log*-SAT($\Gamma_{t,1}$) is M[t]-complete for $t \geq 2$. Essentially the same question is whether (ETH$_t$) is equivalent to the statement

$$\text{SAT}(\Gamma_{t,1}) \notin \text{DTIME}(2^{o^{\text{eff}}(n)} \cdot m^{O(1)}).$$

Proving such a result would probably require some form of a Sparsification Lemma for the higher levels, an interesting problem in itself. Of course one could also try to eliminate the use of the Sparsification Lemma from the proof of the M[1]-completeness of *p-log*-SAT($\Gamma_{1,3}$) and possibly even prove completeness under fpt many-one reductions (instead of Turing reductions).

And finally, it is a notorious open question in parameterized complexity theory if a collapse such as W[t] = FPT on some level t of the W-hierarchy has any implications for the higher levels (ideally, implies W[t'] = FPT for all t'). In view of the entanglement of the W-hierarchy and the M-hierarchy, one possible approach to this question would be to prove a corresponding result for the M-hierarchy. An equivalent formulation of the question for the M-hierarchy is whether $\neg$(ETH$_t$) implies $\neg$(ETH$_{t'}$) for $t' > t$.

References

[1] K.A. Abrahamson, R.G. Downey, and M.R. Fellows. Fixed-parameter tractability and completeness IV: On completeness for W[P] and PSPACE analogs. *Annals of Pure and Applied Logic*, 73:235–276, 1995.

[2] L. Cai, J. Chen, R.G. Downey, and M.R. Fellows. On the structure of parameterized problems in NP. *Information and Computation*, 123:38–49, 1995.

[3] L. Cai and D. Juedes. On the existence of subexponential parameterized algorithms. *Journal of Computer and System Sciences*, 67(4):789–807, 2003.

[4] J. Chen, B. Chor, M. Fellows, X. Huang, D. Juedes, I. Kanj, and G. Xia. Tight lower bounds for certain parameterized NP-hard problems. In *Proceedings of the 19th IEEE Conference on Computational Complexity*, pages 150–160, 2004.

[5] J. Chen, X. Huang, I. Kanj, and G. Xia. Linear fpt reductions and computational lower bounds. In *Proceedings of the 36th ACM Symposium on Theory of Computing*, pages 212–221, 2004.

[6] Y. Chen and J. Flum. On miniaturized problems in parameterized complexity theory. In *Proceedings of the 1st International Workshop on Parameterized and Exact Computation*, 2004.

[7] Y. Chen, J. Flum, and M. Grohe. Bounded nondeterminism and alternation in parameterized complexity theory. In *Proceedings of the 18th IEEE Conference on Computational Complexity*, pages 13–29, 2003.

[8] E. Dantsin, A. Goerdt, E. A. Hirsch, R. Kannan, J. M. Kleinberg, Ch. H. Papadimitriou, P. Raghavan, and U. Schöning. A deterministic $(2-2/(k+1))^n$ algorithm for k-SAT based on local search. *Theoretical Computer Science*, 289(1):69–83, 2002.

[9] R. Downey. Parameterized complexity for the skeptic. In *Proceedings of the 18th IEEE Conference on Computational Complexity*, 2003.

[10] R. Downey, V. Estivill-Castro, M. Fellows, E. Prieto-Rodriguez, and F. Rosamond. Cutting up is hard to do: the parameterized complexity of k-cut and related problems. In J. Harland, editor, *Proceedings of the Australian Theory Symposium*, volume 78 of *Electronic Notes in Theoretical Computer Science*. Elsevier Science Publishers, 2003.

[11] R.G. Downey and M.R. Fellows. Fixed-parameter tractability and completeness I: Basic results. *SIAM Journal on Computing*, 24:873–921, 1995.

[12] R.G. Downey and M.R. Fellows. Fixed-parameter tractability and completeness II: On completeness for $W[1]$. *Theoretical Computer Science*, 141:109–131, 1995.

[13] R.G. Downey and M.R. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.

[14] U. Feige and J. Kilian. On limited versus polynomial nondeterminism. *Chicago Journal of Theoretical Computer Science*, 1997. Available at http://cjtcs.cs.uchicago.edu/.

[15] J. Flum, M. Grohe, and M. Weyer. Bounded fixed-parameter tractability and $\log^2 n$ nondeterministic bits. In *Proceedings of the 31st International Colloquium on Automata, Languages and Programming*, volume 3142 of *Lecture Notes in Computer Science*, pages 555–567. Springer-Verlag, 2004.

[16] J. Goldsmith, M. Levy, and M. Mundhenk. Limited nondeterminism. *SIGACT News*, 1996.

[17] J. Hromkovič. *Algorithmics for Hard Problems*. Springer-Verlag, 2nd edition, 2003.

[18] R. Impagliazzo and R. Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62:367–375, 2001.

[19] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.

[20] K. Iwama and S. Tamaki. Improved upper bounds for 3-sat. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, page 328, 2004.

[21] D.S. Johnson and M. Szegedy. What are the least tractable instances of max independent set? In *Proceedings of the 10th annual ACM-SIAM Symposium on Discrete Algorithms*, pages 927–928, 1999.

[22] C. Kintala and P. Fischer. Refining nondeterminism in relativised polynomial time bounded computations. *SIAM Journal on Computing*, 9:46–53, 1980.

[23] C.H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43:425–440, 1991.

[24] R. Paturi, P. Pudlák, M. E. Saks, and F. Zane. An improved exponential-time algorithm for *k*-SAT. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, pages 628–637, 1998.

[25] U. Schöning. A probabilistic algorithm for *k*-SAT and constraint satisfaction problems. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*, pages 410–414, 1999.

[26] R. E. Stearns and H. B. Hunt III. Power indices and easier hard problems. *Mathematical Systems Theory*, 23:209–225, 1990.

[27] G.J. Woeginger. Exact algorithms for NP-hard problems: A survey. In M. JÃ$\frac{1}{4}$nger, G. Reinelt, and G. Rinaldi, editors, *Combinatorial Optimization - Eureka, You Shrink!, Papers Dedicated to Jack Edmonds, 5th International Workshop*, volume 2570 of *Lecture Notes in Computer Science*, pages 185–208. Springer Verlag, 2001.

The Concurrency Column

BY

Luca Aceto

BRICS, Department of Computer Science
Aalborg University, 9220 Aalborg Ø, Denmark
luca@cs.auc.dk, http://www.cs.auc.dk/~luca/BEATCS

Process calculi like ACP, CCS, CSP and various flavours of the π-calculus are popular specification formalisms for concurrent, distributed and possibly mobile systems. The semantic theory of process calculi has been the subject of extensive investigation for about twenty five years now, and several robust, general principles and results applying to a variety of different formalisms have been isolated in this field of concurrency theory. For instance, structural operational semantics has been successfully applied as a formal tool to establish results that hold for classes of process description languages. This has allowed for the generalization of well-known results in the field of process algebra, and for the development of a meta-theory for process calculi based on the realization that many of the results in this field only depend upon general semantic properties of language constructs. Another approach for the development of a mathematical theory that can cover several key concepts in the theory of process calculi is based on category theory. The main aim of this approach is to develop a general mathematical framework within which one can study notions of behavioural semantics for formalisms that, like process calculi, Petri nets, bigraphs and graph grammars, have an underlying reduction-based operational semantics. This issue of the Concurrency Column is devoted to a paper by Pawel Sobocinski that presents the general agenda of this research programme, puts it in the context of the classic study of behavioural semantics for process calculi, and reports on some of his contributions to this line of research. Enjoy it!

This column will be published soon after CONCUR 2004, the 15th International Conference on Concurrency Theory, that was held in London in the period 31 August–3 September 2004. This was the best attended CONCUR conference to date, and its lively scientific programme witnessed the vitality of

our research field. While waiting for a conference report to appear in a future volume of the Bulletin, I encourage those of you who, like me, could not travel to London for the whole week to check the programme of the main conference and its satellite workshops at the URL http://www.doc.ic.ac.uk/concur2004/.

To have an idea of the difficult choices that the attendees of the pre-conference workshops had to make, it suffices only to note that Rob van Glabbeek, Chris Hankin, Andrew Pitts, Corrado Priami and Julian Rathke were delivering invited talks concurrently in the morning session, and Rocco De Nicola, Andrew Finney, Rob van Glabbeek, Roberto Gorrieri and Uwe Nestmann were speaking at the same time in the afternoon! Which talks would you have chosen? As organizer of one of the workshops, I was left without a choice, and maybe that was just as well.

PROCESS CONGRUENCES FROM REACTION RULES

Paweł Sobociński
IT University, Copenhagen

Abstract

This article is an overview of the recent developments of a theory originally introduced by Leifer and Milner: given a formalism with a reduction semantics, a canonical labelled transition system is derived on which bisimilarity as well as other other equivalences are congruences, provided that the contexts of the formalism form the arrows of a category which has certain colimits. We shall also attempt to provide a context for these developments by offering a review of related work.

1 Introduction

We shall discuss an attempt to develop general mathematical technology for the study of the behavioural theory of computational formalisms with underlying reduction-based operational semantics. Such formalisms include both syntactic models, such as functional programming languages and process-calculi, as well as graphical models such as Petri nets or bigraphs.

The basic technical idea is very simple and can be expressed fairly concisely within a single paragraph: a formalism is equipped with a labelled transition system (lts) semantics where the labels on the transitions out of any particular state

are the smallest contexts which, when instantiated with the term corresponding to that state, can reduce. If the notion of "smallest" is well-behaved enough – in the sense that it is defined via an appropriate universal property – the resulting synthesised lts is very well-behaved. For instance, many popular lts-based equivalences are congruences.

We shall begin by discussing an extension of Leifer and Milner's theory of reactive systems to a 2-categorical setting. This development is motivated by the common situation in which the contexts of a reactive system contain non-trivial algebraic structure with an associated notion of context isomorphism. Forgetting this structure often leads to problems and we shall show that the theory can be extended smoothly, retaining this useful information as well as the congruence theorems. The results reported appeared first in the workshop paper [69] and its journal version [71]. Technically, the generalisation includes defining the central notion of groupoidal-relative-pushout (GRPO) (categorically: a bipushout in a pseudo-slice category), which turns out to provide a suitable generalisation of Leifer and Milner's relative pushout (RPO). The congruence theorems are then reproved in this more general setting. We shall also outline how previously introduced alternative solutions to the problem of forgetting the two-dimensional structure can be reduced to the 2-categorical approach.

Secondly, we shall discuss how GRPOs are constructed in settings which are general enough to allow the theory to be applied to useful, previously studied examples. Indeed, GRPOs were first constructed in a category whose arrows correspond closely to the contexts of a simple process calculus with CCS-style synchronisation. This construction was extended to the category of bunch contexts, studied previously by Leifer and Milner. The constructions use the structure of extensive categories [9]. An account of these translations and constructions appeared first the conference paper [70] and shall appear in the upcoming journal version [73]

Finally, we shall argue that cospans provide an interesting notion of "generalised contexts". In an effort to find a natural class of categories which allows the construction of GRPOs in the corresponding cospan bicategory, we shall consider the class of adhesive categories. As extensive categories have well-behaved coproducts, so adhesive categories have well-behaved pushouts along monomorphisms. Adhesive categories also turn out to be a useful tool in the study and generalisation of the theory of double-pushout graph transformation systems, indeed, such systems have a rich rewriting theory when defined over adhesive categories. Adhesive categories were first introduced in the conference paper [44].

Armed with the theory of adhesive categories, we are able to construct GRPOs in input-linear cospan bicategories. As an immediate application, the construction shed light on as well as extend the theory of rewriting via borrowed contexts, due to Ehrig and König [17]. Secondly, we shall examine the implications of the

construction for Milner's bigraphs [34]. A detailed account of the construction first appeared in the technical report [72].

All the original research mentioned in this article is presented in detail in the author's PhD dissertation [78].

2 Background

Our main source of inspiration shall be the field of process calculus, which is concerned with foundations of concurrent and mobile computation. The field has enjoyed wide popularity over the last 20 years, with several successful depth-first research programs. The usual approach has been to define a relatively simple (compared to industrial programming languages such as ML, Java or C) syntax-based process languages, sometimes referred to as a process algebras or process calculi. These calculi are designed so that they exhibit some fundamental aspect of computation, and research is then devoted to the study of the calculus' behavioural theory, its "expressivity" and decidability aspects. The theory of such calculi is often complicated, perhaps because of the various design decisions involved in the design of a calculus. This fragmented picture makes it difficult to extract generalised principles which are robust, meaning that they apply in several different formalisms. As a result, the field has been described as being in a state of flux [48].

The approach taken up in the research program outlined in this article is breadth-first, in the sense that we are not directly interested in such notions as synchronisation or mobility of code. Rather, we focus on developing a mathematical theory that can, to some extent, cover several basic concepts which have some role to play in many process calculi. Such an approach can be criticised for being too artificial; we are, after all be concerned with "man-made" things like process-calculi, and not "natural" things such as concurrency or mobility. However, while most of the benefits of the (future) full development of the theory discussed here shall be reaped at the meta level (with process-calculus *designers* perhaps benefiting from the insight derived from a general treatment several basic issues common to many calculi) it could be argued that such a general approach may help in isolating robust common principles of important sub-concepts under the umbrella of concurrency or mobility.

In this sense, the approach outlined in this article is related to the development of a domain theory for concurrency [62,59], which advocates the use of mathematics to guide the design of process calculi [60, 61], instead of the, more common, reverse methodology of expending much effort on understanding particular ad-hoc process languages with the use of mathematics. Similarly, the ideas presented share the idea of finding an underlying formalism in which one can study some of

the issues which occur in existing process languages with Milner's work on action calculi [52] and bigraphs [54], as well as with Gadducci and Montanari's work on tile models [26]. Differently from the first two of these, we do not introduce a monolithic model into which we find encodings of other formalisms. The idea is rather to build from bottom-up instead of top-down, i.e. start with basic structures and study their theory instead of starting with a powerful model which is capable of subsuming other formalisms via encodings. In this facet, the approach taken here is consistent with the mathematical tradition of simplifying complex situations into a simple yet rich structure which is amenable to systematic study.

The original material outlined within this article is intended as a contribution in the field of concurrency theory. Since much of it relies on using the language and technology of category theory, parts of it may be considered to be in the field of applied category theory. At all times care is taken to use *standard* and well-studied concepts: 2-categories [40], bicategories [5], bicolimits [79,39] and extensive categories [9]. Indeed, by finding the right mathematical structures to model concurrent (and other) computational phenomena one can use well-understood and elegant tools to solve problems, instead of developing specialised and ad-hoc mathematics from scratch. The only novel categorical concepts discussed are the classes of *adhesive* and *quasiadhesive* categories [44]; we shall argue that they are both natural from a mathematical point of view and useful for computer science.

3 Reaction semantics

By a reaction[1] semantics we mean an unlabelled transition system, usually generated by closing a small set of *reaction rules* under *reactive* (evaluation) contexts. An agents p *reacts* into an agent q when there has been an interaction (specific to the calculus) inside p which, after its application, results in the agent q. The actual technical mechanism of performing a reaction can be seen as an instance of term rewriting; at least in examples where terms are syntactic and not quotiented by exotic structural congruences.

The basic setup involving contexts (which organise themselves as a category, with substitution as composition), rules and reactive contexts corresponds to a mathematical structure: Leifer and Milner's notion of *reactive system* [48]. A reactive system consists of an underlying category $\mathbf{C}$ with a chosen object 0 and a collection $\mathbf{D}$ of arrows of $\mathbf{C}$ called reactive contexts[2]. The arrows with domain 0

[1] Many authors use the term 'reduction' instead of 'reaction'. We shall use 'reaction' because the word 'reduction' is related to the concept of termination, and termination is usually not an interesting notion in concurrency theory.

[2] There are some additional constraints on the set of reactive contexts which we do not specify here.

are usually called terms or agents, other arrows are contexts; composition of arrows is understood as substitution. Thus, for example, a term $a : 0 \to X$ composed with a context $c : X \to Y$ yields a term $ca : 0 \to Y$.

The reaction rules are of the form $\langle l, r \rangle$, where $l : 0 \to C$ is the redex and $r : 0 \to C$ is the reactum. Notice that the rules are *ground* in that they are terms and do not take parameters. One generates a *reaction relation* $\longrightarrow\!\!\triangleright$ by closing the reaction rules under all reactive contexts; we have $p \longrightarrow\!\!\triangleright q$ if, for some $d \in \mathbf{D}$, we have $p = dl$ and $q = dr$. The advantage of a theory at least partly based in the language of category theory is that the constructions and proofs are performed on an abstract level, meaning that they are portable across a range of models.

In many cases, modern presentations of well-known process calculi have their semantics formalised in terms of an underlying rewriting system. This includes the more recent incarnations of CCS [51, 53][3], the Pi-calculus [55, 53, 68][4] and the Ambient Calculus [10][5]. These calculi are all syntax based, but have non-trivial structural congruences associated with the syntax. Taking the terms and contexts up to structural congruence clearly results in a setting where substitution is associative. Moreover, they all have specialised notions of reactive contexts; in CCS for instance, any context which has its hole under a prefix does *not* preserve reaction and thus, in our terminology, is *not* reactive. Thus, all of these calculi can be seen as instances of reactive systems.

4 Process equivalence

There have been various attempts at defining process equivalences starting with the reaction semantics. The notion of process equivalence is of fundamental importance, both theoretically and for practical reasons. For theorists, a natural contextual process equivalence is a starting point in the development of bisimulation-based proof techniques, logical characterisations, model checking of restricted classes and so forth. More practically, process equivalence may be used, for instance, to check that a program adheres to its specification; assuming an a priori encoding of both the program and the specification into a chosen formalism.

The idea of generating a process equivalence using contextual reasoning goes back to the definitions of Morris-style process equivalences of the simply typed and the untyped variants of the lambda calculus [3], as well as other functional formalisms. In the field of process calculus and process algebra, such equivalences are sometimes called *testing* equivalences [29].

[3] fundamental notion: synchronisation on names.

[4] fundamental notion: name passing, with the associated notion of scope extrusion. Early exploratory work in this field was done by Engberg and Nielsen [20].

[5] fundamental notion: spatial mobility of process code.

We shall now discuss some of developments in the quest of finding general techniques for generating equivalences from reaction rules which are relatively robust in that they are not specialised to a single process calculus. The first is the notion of *barbed congruence* by Milner and Sangiorgi [56]. In that article, the authors first study *reduction bisimulation* which involves comparing the internal evolutions of processes. The equivalence this gives is very coarse, and in order to obtain something sensible, one has to close contextually (in one of two possible ways, as we shall discuss later). Milner and Sangiorgi do this in CCS, obtaining *reduction congruence*. The resulting process equivalence is coarser than bisimilarity on the standard labelled transition system semantics, but the correspondence is close. The reason for the mismatch is, essentially, that a congruence built up from reactions does not distinguish certain processes with infinite internal behaviour. To fix the congruence, Milner and Sangiorgi proposed adding an extra ad-hoc notion of observable based on the underlying syntax of CCS. This extra notion of observable is known as a *barb*. Their work has proven very influential and can be repeated for other calculi [10, 84, 11, 28], with the notion of barb chosen ad-hoc in each calculus, using calculus-specific intuition.

An important study which develops a process equivalence based purely on reactions is by Honda and Yoshida [31] who, based on intuitions from the λ-calculus, build equational theories directly from rewrites requiring no a priori specification of observables. They achieve this by using reduction and contextual closure as well as the equating of *insensitive* terms. These are terms which can never interact with their environment or, in other words, can never contribute to a reaction with a context. This elegant characterisation of a useful equivalence which is robust across many formalisms and relies only on the underlying reaction semantics is close in spirit to the aims of the theory presented in this article. The full investigation of the relationship between the two theories is an important direction for future work.

As we've hinted earlier, starting with reduction bisimilarity, one can obtain a sensible congruence in at least two ways which give, in general, different results. First, Honda and Yoshida [31] advocate obtaining a congruence by considering the largest congruence contained in bisimilarity which is also a bisimulation (or, equivalently, postulating congruence in the definition of a bisimulation relation and then considering the resulting bisimilarity). Similarly, an earlier work by Montanari and Sassone [58] obtains a congruence from bisimilarity[6] by considering the largest congruent bisimulation, there called *dynamic bisimilarity*. Alternatively, Milner and Sangiorgi's barbed congruence is defined as follows: two processes p and q are barbed congruent if, given any context c, $c[p]$ and $c[q]$ are barbed bisimilar. This yields the largest congruence contained in bisimilarity.

[6] More precisely, weak bisimilarity on the lts semantics of CCS.

The first approach gives, in general, a finer congruence. This is because any relation which is both a congruence and a barbed bisimulation is clearly included in barbed congruence. On the other hand, the reverse direction is not true in general as barbed congruence may not be a barbed bisimulation.

Fournet and Gonthier [24] have confirmed that the barbed congruence in the style of Milner and Sangiorgi coincides with the barbed congruence in the style of Honda and Yoshida (usually called reduction equivalence) in the setting of the Pi-calculus. In other process calculi, the situation is less clear.

Equivalences which are based on an underlying reduction system and are generated contextually have both advantages and disadvantages. Their chief advantage is their naturality, in the sense that it is often relatively easy to justify their correctness and appropriateness as notions of equivalence. A disadvantage of barbed congruence in particular, is that the barbs, or observables, are a usually of a rather ad-hoc syntactic nature, specific to each calculus. An important common problem of contextually defined equivalences is that it is often very difficult to prove directly that two process terms are equivalent. The main complication follows from the quantification over all contexts, usually an infinite number. Thus, in order to prove equivalence directly, one has to construct a proof based on structural induction; this, when possible, is usually a tedious and a complicated procedure.

We should note that contextually based equivalences based on reduction rules naturally come in *strong* and *weak* variants. A strong equivalence allows one to distinguish processes which vary only in how they react internally, while weak equivalences aim to abstract away from internal reaction. Although weak equivalences are more suitable as a notion of observational equivalence, we shall concentrate our theoretical development on strong equivalences. We shall return to the topic of weak equivalences later in the article.

5 Labelled transition systems

An elegant solution to the problem of universal quantification over the usually infinite set of contexts is to endow a process calculus with an appropriate labelled transition system (lts) semantics. Before we explain what is meant by 'appropriate' in this setting, we shall recall some of the basic theory behind lts semantics. Labelled transition systems have been a very popular tool in theoretical computer science, not least because of their origins in classical automata theory. Indeed, some process calculi, including the earlier variants of the well known CCS [51], have their semantics *a priori* formalised in terms of an lts; the use of reduction based semantics and structural congruence only becoming fashionable after Berry and Boudol's influential work [7] on the chemical abstract machine.

A labelled transition system consist of a set of states S and a set of labelled

transitions T. A transition has a domain state, a codomain state and a label from some, usually fixed, set A of "actions". Technically, the set of transitions is usually considered to be a subset of the cartesian product $S \times A \times S$ which brings with it the usual restriction of there being at most one transition with label a between any two states. Although the intuition may vary between applications, it is often the case that a transition with label a from state s to state s' means that s can participate in an interaction which the symbol a represents, and by doing so, evolve into s'. Although our use of the term "interaction" is intentionally meant to be vague, when there is an underlying reduction semantics such an interaction could be represented by a reaction.

Labelled transition system semantics facilitate a large number of equivalences which vary depending on how much branching structure is taken into consideration. Thus, one of the coarsest (relates most) is the trace preorder and associated equivalence because no branching is taken into consideration. Park's notion of *bisimilarity* [63], adapted for labelled transition systems by Milner [51], is at the other end of the spectrum [83], meaning that it examines all branching structure and is the finest (relates least) of such equivalences. Bisimilarity is often denoted $\sim$.

The notion of bisimilarity has stimulated much research because it is canonical from a number of perspectives. Firstly, it has a elegantly simple coinductive definition, meaning that in order to prove that two states of an lts are bisimilar, it is enough to construct a bisimulation which contains them. Secondly, it has an elegant game-theoretic characterisation in terms of the so-called bisimulation game. Thirdly, there is an elegant and simple logical characterisation in terms of the well-known Hennessy-Milner logic [30]. Finally, there are two, so far largely unrelated general approaches to bisimilarity. The first is usually known as the coalgebraic approach, where a bisimulation is sometimes defined as a spans of coalgebra morphisms for some functor [67]. This is a very general approach which recovers the notion of ordinary bisimulation for a particular endofunctor on the category of sets, namely $\mathcal{P}(A \times X)$ where A is the set of labels of the lts and $\mathcal{P}$ is the power set. In order for the final coalgebra to exist [1,4], one needs to consider the finite power set $\mathcal{P}_f$ functor, which corresponds to the technical assumption of requiring the lts to be *finitely branching*. Observational equivalence, when final coalgebras exist, is sometimes taken to mean equality under the unique mapping to the final coalgebra. Span bisimilarity and observational equivalence via the map to the final coalgebra yield the same equivalence under certain assumptions on the underlying endofunctor. The second general approach to bisimulation is the open map approach [35], where a bisimulation is taken as a span of so called open maps in a category of transition systems and simulations. Open maps are taken with respect to an ad-hoc underlying subcategory of open maps, which led to the study of presheaf categories where such path categories are canonical via

the Yoneda embedding. This approach has lead to research on the aforementioned domain theory for concurrency.

While all of the above form an impressive body of theory on bisimilarity, they all start off with the following assumption: a predefined set of actions A over which the labelled transition systems are built in some, usually unspecified way. Indeed, even the fact that the states of the lts correspond to the terms of some formalism is usually abstracted away.

A work in the general area of combining lts semantics with some notion of syntax is the seminal paper by Turi and Plotkin [81] which combines the coalgebraic approach with structural operational semantics [64] (and in particular the GSOS [8] format) in a comprehensive theory known as *bialgebraic semantics*. Similar ideas have been pursued by Corradini, Heckel and Montanari [13], who used a coalgebraic framework to define labelled transition systems on algebras.

The area of bialgebraic semantics is an exciting field with ongoing research into extending the basic theory with the generation of new names [23, 22] and equivalences other than bisimilarity [42, 41]. Such developments yield insights into labelled transition systems and isolate SOS formats which guarantee congruence properties in such settings. However, even in bialgebraic semantics, the labels of the lts are assumed to come from some fixed ad-hoc set of observable behaviours which one is meant to provide a priori for each setting.

6 Lts for reactive systems

We shall now consider the question of what constitutes an appropriate labelled transition system for a formalism with an underlying reaction semantics and some standard contextually-defined equivalence. Firstly, bisimilarity on such an lts should be at least *sound* with respect to the standard contextually-defined equivalence, meaning that to prove that two terms are contextually equivalent it is enough to show that they are bisimilar. In some cases, bisimilarity is also *complete* (or fully-abstract) with respect to the contextually-defined equivalence, meaning that the two notions of process equivalence – bisimilarity and contextually-defined equivalence – actually coincide, and one can always, in principle, find a bisimulation for any two contextually equivalent processes.

Thus the chief advantage of such a suitable lts is that, in order to prove the equivalence of two processes, one can use the power of coinduction and construct a bisimulation which includes the two processes. This task is usually more attractive and easier then the messy structural inductions involved in proving contextual equivalence defined using quantification over an infinite set of contexts.

There has been much research concerned with finding suitable labelled transition system semantics for different reaction-based formalisms. Unfortunately,

from a theoretical point of view, the labels of such a semantics – if it exists – may seem ad-hoc; they need to be tailored and locally optimised for each process language under consideration. Indeed, the task of identifying a "natural" lts for a particular calculus is often far from obvious, even when its semantics is well understood. On the contrary, labelled transition systems are often intensional: they aim at describing observable behaviours in a compositional way and, therefore, their labels may not be immediately justifiable in operational terms. For example there are two alternative labelled transition system semantics for the Pi-calculus [55], the early and the late version, each giving a different bisimulation equivalence.

An additional benefit of full abstraction and a property of considerable importance in its own right is *compositionality* of lts bisimilarity (and of other useful lts preorders and equivalences). A relation is compositional, in other words a *congruence*, if whenever we have tRu then we have $c[t]Rc[u]$ for any context $c[-]$ of the underlying language. It can be argued that congruence should be a required property of any reasonable notion of observational equivalence – if we prove that a and b are indistinguishable then they certainly should behave equivalently in any given environment.

Compositionality and coinduction work together: compositionality allows one to use modular reasoning to simplify coinductive proofs. Indeed, compositionality is highly desirable because it usually makes equivalence proofs considerably simpler. In particular, it allows the familiar methods of equational reasoning, such as substituting "equals for equals", sound. As an example, consider two nontrivial systems, each of which can be expressed as a parallel composition of two smaller systems, in symbols $p \equiv q \parallel r$ and $p' \equiv q' \parallel r'$. To show that $p \sim p'$, using compositionality it is enough to show that $q \sim q'$ and $r \sim r'$.

It is a serious problem, then, that given an lts designed ad-hoc for a particular calculus, bisimilarity is not automatically a congruence. Even when it *is* a congruence, proving that it is can be a very difficult and technical task. For example, the well-known Howe's method [32] is a technique for proving that lts bisimilarity is a congruence for certain languages with higher-order features. In the field of process calculus, such proofs usually involve finding a close connection between the labels of an lts and the syntactic contexts of the calculus.

Interestingly, from a historical perspective, labelled transition systems as a way of formalising semantics of process calculi actually were used *before* reaction semantics. In particular, the original presentation [51] of Milner's CCS formalised the semantics with a labelled transition system presented with SOS-style rules. An early paper by Larsen [45] identified the importance of congruence results for lts based process equivalences. Starting with an lts, Larsen introduced the notion of a context (itself an lts) which is capable of consuming the actions of a state in the lts. By adding constructors (action prefix and nondeterministic choice) to the set

of contexts, he proved a congruence theorem for bisimilarity. This early work can be seen as related to CCS-like calculi, since Larsen's environments can be otherwise understood as ordinary CCS contexts (with input-actions changed to output-actions and vice-versa) – with the consumption of lts labels by the context being handled by CCS interaction. Even in the basic setting of CCS, it quickly became apparent that the labelled transition systems is not the ideal technology with which to define notions of observational equivalence. For instance, weak bisimilarity in CCS is *not* a congruence. Because, as we have demonstrated, compositionality is a very useful property, Montanari and Sassone [57, 58] considered the largest congruent bisimulation contained in weak bisimilarity. Alternatively, weak observational congruence [51] considers the largest congruence contained in bisimilarity (the difference is similar to the difference between Milner and Sangiorgi's and Honda and Yoshida's approaches). These approaches became for some time accepted techniques for obtaining satisfactory notions of observational equivalence in calculi. The advent of reaction semantics and congruences obtained from reactions have since arguably replaced these approaches as "canonical" methods of obtaining an observational equivalence.

7 Weak equivalences

Another yardstick to measure the appropriateness of an lts for a formalism with reactions is how the lts simulates internal reduction within terms. For example, in CCS and many other calculi, there are "silent" transitions; traditionally labelled τ. Such τ transitions usually correspond closely to the underlying reaction semantics.

Having τ labels as part of an lts allows one to define a notion of *weak* bisimulation and the resulting equivalence: *weak* bisimilarity. Roughly, weak bisimilarity does not distinguish processes which differ only in internal behaviour as represented by the τ-labelled transitions. Such equivalences are considered to be more useful from a practical point of view since it can be argued that any reasonable notion of observational equivalence should not take internal behaviour into consideration.

There are a number inequivalent ways [82] to define precisely what is meant to be a weak equivalence and the appropriateness to any particular application depends on the ad-hoc design of the particular lts. The techniques involved are usually not specialised to bisimilarity and thus one may easily define a notion of weak trace equivalence or a weak failures equivalence. One popular definition pioneered by Milner [51] is allow a (non-τ) label a to be matched by a "weak" a, which means a (possibly empty) sequence of τ labels followed by a and followed again by a (possibly empty) sequence off τs. A τ label is normally allowed to be matched by any (possibly empty) string of τs. As mentioned before, weak

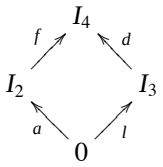


Figure 1: Redex square.

bisimilarity in CCS is not a congruence.

Weak equivalences have traditionally been difficult to handle in general categorical settings. Indeed, there is still no general approach based on coalgebras, although there has recently been an attempt [50] to develop the theory in this direction. The theory has been developed to a more satisfactory level in the field of open maps [21], yet the general approach advocated there is, arguably, quite technical. Surprisingly, the theory of weak bisimulation seems to be quite easily and elegantly handled in the theory of reactive systems, see Jensen's upcoming PhD thesis [33].

8 Deriving bisimulation congruences

We have discussed attempts by Milner and Sangiorgi [56] and by Honda and Yoshida [31] to identify general techniques at arriving at a reasonable notion of process congruence through contextual means. We have also discussed some of the problems inherent in contextual definitions and discussed one solution to the difficulties involved in quantifying over an infinite set of contexts, finding a *suitable* labelled transition system. A third development, which has led in a direct line to the the theory described in this article, is by Sewell [76]. Sewell's idea is to *derive* a labelled transition system directly from the reaction semantics so that useful lts based equivalences, including bisimilarity, are automatically congruences.

Sewell's approach involved a new way of obtaining a labelled transition: the labels of transitions from a particular term should be the contexts which allow the term to react (that is, a rewrite of the term inside the context should be possible in the underlying rewriting semantics). Moreover, the labels should be, in some sense, the *smallest* such contexts. The notion of smallest was elegantly expressed in categorical terms by Leifer and Milner [48].

Leifer and Milner's characterisation of the notion of smallest context utilises the fact that contexts can be organised in a category as part of a reactive system. First, the notion that a term a can be instantiated in a context f and react can be summed up by giving a commutative *redex* square, as illustrated in Figure 1,

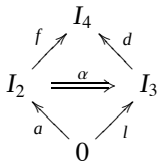


Figure 2: Redex square in a 2-category.

where d is some *reactive* context and l is the redex or a reaction rule.

Using Leifer and Milner's characterisation, the context f is the smallest such context when the diagram is an *idem pushout* (IPO). Categorically, it means that it is a pushout in the slice category over I_4. Starting with an arbitrary redex square, one obtains an IPO by constructing a *relative pushout* (RPO), which amounts to constructing a pushout in the relevant slice category.

The advantages of such a definition is that we have the universal properties of such contexts at our disposal. Indeed, Leifer and Milner [48] showed that a labelled transition system with labels being precisely the contexts which come from IPOs is very well behaved. In particular, bisimilarity is a congruence. In his PhD dissertation, Leifer [47] complemented this result by showing that trace equivalence and failures equivalence are also congruences. In the examples treated by Sewell, Leifer and Milner, bisimilarity on the labelled transition semantics obtained using this approach have corresponded closely to the expected process equivalences.

9 A 2-categorical approach

When applied naively, Leifer and Milner's theory has proven inadequate in reactive systems where contexts have non-trivial algebraic structure. In some cases, IPOs do not give the expected labels in the lts [71], while in others, they do not exist [70]. The troublesome contexts often exhibit non-trivial automorphisms, which naturally form a part of a 2-dimensional structure on the underlying category **C**. It is important to notice that such situations are the norm, rather than the exception. Context isomorphisms arise naturally already in simple process calculi with a parallel composition operator, where terms are considered up to structural congruence which ensures that parallel composition is associative and commutative. In more accessible terms, whereas Leifer and Milner consider categories where the objects are "holes" and arrows are contexts, we shall consider *2-categories* where the intuition for the objects and arrows is the same as for Leifer and Milner, but there is additional structure, the 2-cells. The suggested intuition that the

2-cells is a term isomorphism, in a loose sense, a "derivation" or "proof" of structural congruence. To give a redex square in this setting, it is not enough to say that a in the context of f *equals* a redex l in a reactive context d, one needs to provide an explicit isomorphism α, as illustrated in Figure 2. It turns out that this 2-dimensional structure is crucial and solves many of the problems involved in Leifer and Milner's original theory. The idea of using 2-cells as part of the theory of reactive systems was independently proposed by Sewell [77].

The suitable generalisations of IPO and RPO to this 2-dimensional setting, dubbed GIPO and GRPO, were introduced in [69, 71]. The associated categorical notion is no longer a pushout in a slice category but rather a bipushout [39, 79] in a pseudo-slice category. It turns out, however, that these extra complications do not detract from the good behaviour of the resulting lts; bisimilarity as well as trace and failures equivalences are congruences.

Leifer and Milner, aware of the problems which arise as a consequence of discarding the 2-dimensional structure, have also introduced technology in order to deal with these issues. The main developments have centered around Leifer's theory of *functorial reactive systems* and Milner's S-*precategories* [34]. These solutions have a similar flavour: decorate the contexts by so-called "*support sets*" which identify elements of the contexts so as to keep track of them under arrow composition. This eliminates any confusion about which automorphism to choose since diagrams can now be commutative in only one way. Unfortunately, such supported structures no longer form categories – arrow composition is partial – which has the effect of making the theory laborious and based in part on set theoretical reasoning and principles.

A translation which maps reactive systems on precategories to reactive systems on 2-categories in a way which ensures that the lts generated using the 2-categorical approach is the same as the lts generated using the technology functorial reactive systems or S-precategories was presented in [70, 73]. The translation derives a notion of isomorphism, specific to the particular structure in hand, from the precategory's support information. Such isomorphisms constitute the 2-cells of the derived 2-category. It can be argued that this yields an approach mathematically more elegant and considerably simpler than precategories. Moreover, while subsuming the previous theories, it appears that the 2-categorical theory is more general: there is no obvious way of reversing the translation and obtaining an S-precategory from a general 2-category.

There have been several applications of the theory of 2-categories to computer science, see for example [6, 75, 80, 27, 12]. The 2-dimensional structure has been typically used to model a small-step reduction relation, say in the simply-typed lambda calculus. As in our examples, the objects of the 2-categories are types and the arrows are terms. However, for us the 2-dimensional structure consists of iso-

morphisms between terms, in other words, structural congruence, and the rewrite relation is external to the 2-category. Indeed, there is a fundamental problem in modelling the rewrite relation as 2-cells in our examples, if we allow non-reactive contexts (as, say, prefix in CCS or lambda abstraction in the lazy lambda calculus) as arrows in the category. This is because the axioms of 2-categories ensure that all arrows preserve reaction through horizontal composition with identity 2-cells; otherwise known as "whiskering". In symbols, if $\alpha : f \Rightarrow g : X \to Y$ is a 2-cell then for any $h : Y \to Z$ we have that $h\alpha : hf \Rightarrow hg : X \to Z$ is a 2-cell.

10 Adhesive categories

One approach which aids in understanding constructions on structures such as bigraphs at a general level is: find a natural class of categories which includes many different notions of graphical structures used in computer science and at the same time has enough structure which allows us to derive useful properties. This leads us to the the classes of adhesive and quasiadhesive categories [44].

As is the case with the well-known class of extensive [46, 74, 9] categories, adhesive categories have a simple axiomatic definition as well as an elegant "equivalence" of categories definition. Indeed, the idea behind the development of adhesive categories was to find a class of categories in which pushouts along monomorphisms are "well-behaved" – meaning they satisfy some of the properties of such pushouts in the category of sets and functions **Set** – in much the same way as coproducts are "well-behaved" in extensive categories. Similarly, quasiadhesive categories have well-behaved pushouts along *regular* monos.

Adhesive categories include as examples many of the graphical structures used in computer science. This includes ordinary directed graphs, typed graphs [2] and hypergraphs [16], amongst others. The structure of adhesive category allows us to derive useful properties. For instance, the union of two subobjects is calculated as the pushout over their intersection, which corresponds well with the intuition of pushout as generalised union.

We shall defer the discussion of how adhesive categories fit into the aforementioned 2-categorical theory of process congruences until the next section. Here we shall discuss an immediate application of adhesive categories: one can develop a rich *general* theory of double-pushout (dpo) rewriting [19] within adhesive categories. Dpo *graph* rewriting was first introduced in order to formalise a way of performing rewriting on graphs. It has been widely studied and the field can be considered relatively mature [66, 14, 18].

In dpo rewriting, a rewrite rule is given as a span $L \leftarrow K \rightarrow R$. Roughly, the intuition is that L forms the left-hand side of the rewrite rule, R forms the right-hand side and K, common to both L and R, is the sub-structure to be unchanged

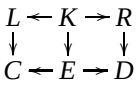


Figure 3: Double pushout.

as the rule is applied. To apply the rule to a structure C, one first needs to find a match $L \to C$ of L within C. The rule is then applied by constructing the missing parts (E, D and arrows), as illustrated in Figure 3, in a way which ensures that the two squares are pushout diagrams. Once such a diagram is constructed we may deduce that $C \Longrightarrow D$, that is, C rewrites to D.

Dpo rewriting is formulated in categorical terms and is therefore portable to structures other than directed graphs. Indeed, there have been several attempts [16, 15] to isolate classes of categories in which one can perform dpo rewriting and in which one can develop the rewriting theory to a satisfactory level. In particular, several axioms were put forward in [16] in order to prove a local Church-Rosser theorem for such general rewrite systems. Additional axioms were needed to prove a general version of the so-called concurrency theorem [43].

Using adhesive categories, one may define *adhesive grammars* which are dpo rewrite systems on adhesive categories. The rewriting theory of such grammars is satisfactory; indeed, one may prove the local Church-Rosser theorem and the concurrency theorem in the general setting without the need for extra axioms. It can thus be argued that adhesive categories provide a natural general setting for dpo rewriting. For further details, the reader is referred to [44].

11 Cospans

Several constructions of RPOs have been proposed in the literature for particular categories of models. For example, Leifer [47] constructed RPOs in a category of action graphs, while Jensen and Milner did so in the precategory of bigraphs [54]. A construction of (G)RPOs in a general setting has so far been missing.

A general construction, provided that it covers several different models and the techniques used are robust, is quite useful. The reasons for this include:

- it provides a general intuition of how to construct GRPOs in many different settings, without having to provide model-specific constrictions and proofs;

- it allows the relating of different models as subcases of a more general setting;

$$I_1 \xrightarrow{\iota} C \xleftarrow{o} I_2$$

Figure 4: Cospan from I_1 to I_2.

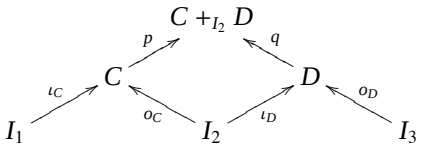


Figure 5: Composition in Cospan$^{\cong}$(**C**)

- it allows one to vary the model within the specified constraints and retain the construction.

We have discussed the modelling of the contexts of a formalism as arrows in an arbitrary category. An interesting question thus arises; what is a reasonable, general and elegant notion of context which nonetheless has more structure than an arrow of an arbitrary category? Secondly, given a category **C**, how can one canonically treat the *objects* as contexts, so that they form the *arrows* of another category? A concrete form of the second question could be: what is a graph context? We argue that the notion of cospan is suitable. Given objects I_1 and I_2 of some category **C**, a cospan from I_1 to I_2 is simply a diagram in **C**, as illustrated in Figure 4, where C is an object of **C** and the arrows are arbitrary. We shall refer to $\iota : I_1 \to C$ and $o : I_2 \to C$ as, respectively, the *input* and *output interface* of the cospan. Note that, as it stands, the notion of cospan is symmetric, and the same diagram forms a cospan from I_2 to I_1 with o forming the input interface and ι the output interface.

The rough intuition is that C corresponds to a "black box" computational environment, with some of its parts available through I_1 to its subcomponents, or variables; and others available publicly through I_2, which can be used to embed C in a larger system.

Given two cospans, $I_1 \xrightarrow{\iota_C} C \xleftarrow{o_C} I_2$ and $I_2 \xrightarrow{\iota_D} D \xleftarrow{o_D} I_3$, one can compose them to obtain a cospan from I_1 to I_3 by constructing the pushout, as illustrated in Figure 5, and letting the input interface be $p\iota_C$ and the output interface be qo_D. Such composition has an identities, the identity cospan on I_1 is $I_1 \xrightarrow{\text{id}} I_1 \xleftarrow{\text{id}} I_1$.

Cospans in **C** actually organise themselves as arrows of another category, or more accurately, the *bi*category Cospan$^{\cong}$(**C**). This bicategory has the same objects as **C** but the arrows from I_1 to I_2 are cospans and the 2-cells are cospan

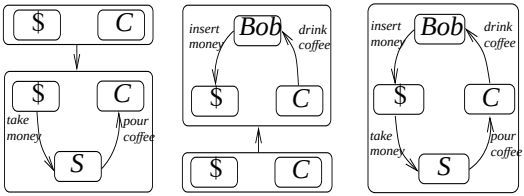


Figure 6: Example of a contextual system.

isomorphisms - isomorphisms $f : C \to C'$ of $\mathbf{C}$ which preserve input and output interfaces, that is $f\iota = \iota'$ and $fo = o'$.

A bicategory [5] can be described roughly as a 2-category where the horizontal composition is associative and has identities up to an isomorphic 2-cell. Composition of cospans is not associative on the nose because composition uses the pushout construction which is defined up to isomorphism. The associativity and identity isomorphisms are required to satisfy the so-called coherence conditions (including the famous Mac Lane pentagon for associativity [49]). It turns out that the canonical isomorphisms obtained using the universal property of pushouts do satisfy these conditions.

As an example of these concepts, consider the simple model of a coffee vending machine, illustrated by the leftmost diagram of Figure 6. It has an output interface consisting of two nodes, \$ and C, which one can think of as a money slot and the coffee out-tray. These are the parts of the coffee machine accessible to the environment, the internal components, represented by S, are invisible. The middle diagram represents a coffee drinker. He expects to see a money slot and a coffee out-tray, which are his input interfaces. As the output interface of the coffee machine and the input interface of the coffee drinker match, one may compose them and obtain the system pictured in the rightmost diagram. (The input interface of the vending machine and the output interface of the coffee drinker have been omitted.)

12 Construction of GRPOs

We shall now discuss a result which ties together the threads which we have discussed so far. It is the central contribution of [78] and appeared first in the technical report [72]: the construction of GRPOs in input-linear cospan bicategories over adhesive categories. By an input linear cospan, we mean a cospan as in Figure 4 but where the input interface ι is mono. Observe that this breaks the symmetry of cospans: to give an input-linear cospan from I_1 to I_2 is not the same thing as

to give an input-linear cospan from I_2 to I_1. When **C** is an adhesive category, the composition of two input-linear cospans in **C** gives an input-linear cospans: they form the bicategory ILC(**C**).

Although technical in nature, the linearity condition does have an intuitive account. As alluded in the coffee drinker example, one can consider a cospan as a "black box," with an input interface and an output interface. The environment cannot see the internals of the system and only interacts with it through the output interface. The fact that the output interface need not be linear means that the system is free to connect the output interface arbitrarily to its internal representation. For example, the coffee machine could have two extra buttons in its output interface; the "café latte" button and the "cappuccino" button. The machine internals could connect both these buttons to the same internal trigger for coffee with milk; the point is that the system controls its output interface and is able to equate parts of it. On the other hand, the system cannot control what is plugged into one of its holes. Thus, an assumption of input-linearity is essentially saying that the system does not have the right to assume that two components coming in through the input interface are equal.

The construction arose from an effort to understand the structure of GRPOs in categories of contexts where the contexts have graphical structure. Incidentally, it is the non-trivial algebraic structure of such contexts that makes it essential to consider 2-dimensional structure in of such categories; it is not enough to deal with the "abstract" versions (where the contexts are quotiented by isomorphism) and consider RPOs. The construction is the first construction of GRPOs for general class of models.

We shall conclude with a discussion of two of the immediate applications of the construction. Firstly, using an insight of Gadducci and Heckel [25] we notice that dpo graph rewriting systems can be seen as certain rewriting systems on cospan categories over the category of directed graphs and homomorphisms **Graph**, and thus can be seen as reactive systems. Since **Graph** is an adhesive category, we are able to derive labelled transition systems for a general class of dpo graph rewriting systems.

One of the advantages of this technology is that it facilitates a transfer of concepts between the theories and technologies of process algebra and graph rewriting. Indeed, it becomes possible to think of graph rewriting systems as certain calculi, with cospans of graphs providing a notion of context. Interestingly, the construction of labelled transition systems captures and extends the borrowed context approach of Ehrig and König [17] who also derive labelled transition systems for double-pushout graph rewriting systems. Indeed, it becomes possible to see their work as part of the framework of reactive systems and GRPOs. The transfer of technology is in both directions, using Ehrig and König's characterisation of labels, we can provide a pleasantly simple characterisation of GIPOs in our setting.

Our second application shall consider Milner's bigraphs [54]. Bigraphs were introduced by Milner in his conference presentation [54] and later in the comprehensive technical report by Jensen and Milner [34]. They aim at modelling systems with two orthogonal modes of connectivity. The first mode is a physical link structure, which may for instance correspond to a physical nesting of systems similar to the nesting of process terms in the ambient calculus [10], or Alastair living next door to Beatrice. The second mode of connectivity is a logical link structure, which may correspond to processes knowing a reference to a resource of an another process, as, for example a process in the Pi-calculus [55] knowing a free name of another process, or Alastair knowing Beatrice's email address. The two sorts of connectivity are orthogonal in the sense that the physical separation of processes should not have an effect on the ability to maintain logical links. Bigraphs are algebraic structures with an underlying carrier set. We shall see how the category of bigraphs can be otherwise defined as a certain cospan bicategory over an adhesive category (which, incidentally, gives an automatic notion of bigraph *homomorphism*).

Considering input-linear cospans allows us to construct GRPOs, allowing the derivation of well-behaved lts for reactive systems over input-linear bigraphs. It turns out that there is a mismatch with Milner's theory of RPOs for bigraphs. Indeed, requiring input-linearity corresponds to taking a different notion of bigraph then the one treated by Milner; it turns out that the category of bigraphs in Milner's sense is actually isomorphic to a certain bicategory of *output*-linear cospans over an adhesive category. As a consequence, it shall be interesting to investigate whether a general construction of GRPOs can be given for output-linear cospans.

Cospans as well as spans have been used in computer science before. As previously mentioned, Gadducci and Heckel [25] have used cospans to shed light on connections between dpo graph rewriting and standard rewriting theory. In an effort to study a general notion of "partial map", Robinson and Rosolini investigated a particular class of span bicategories in [65]. Spans have also been studied by Katis, Sabadini and Walters [37, 38] in an effort to generalise ordinary automata theory in a modular way. Moreover, using the technology of traced monoidal categories [36], they were able to include a "feedback" operation. Thus, as our cospans can be thought of as generalised contexts, their spans can be thought of as generalised automata. It is unclear at this stage what connection can be made between the two theories.

References

[1] P. Aczel and M. P. Mendler. A final coalgebra theorem. In *Category Theory and Computer Science CTCS '89*, volume 389 of *LNCS (Lecture Notes in Computer*

Science). Springer, 1989.

[2] P. Baldan, A. Corradini, H. Ehrig, M. Löwe, U. Montanari, and F. Rossi. Concurrent semantics of algebraic graph transformations. In H. Ehrig, H.-J. Kreowski, U. Montanari, and G. Rozenberg, editors, *Handbook of Graph Grammars and Computing by Graph Transformation*, volume 3, chapter 3, pages 107–187. World Scientific, 1999.

[3] H. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*. North Holland, 1984.

[4] M. Barr. Terminal coalgebras in well-founded set theory. *Theoretical Computer Science*, 114:299–315, 1993.

[5] J. Bénabou. Introduction to bicategories. In *Midwest Category Seminar*, volume 42 of *Lecture Notes in Mathematics*, pages 1–77. Springer-Verlag, 1967.

[6] D. B. Benson. The basic algebraic structures in categories of derivations. *Information and Control*, 28(1):1–29, 1975.

[7] G. Berry and G. Boudol. The chemical abstract machine. *Theoretical Computer Science*, 96:217–248, 1992.

[8] B. Bloom, S. Istrail, and A. Meyer. Bisimulation can't be traced. *Journal of the ACM*, 42:232–268, 1995.

[9] A. Carboni, S. Lack, and R. F. C. Walters. Introduction to extensive and distributive categories. *Journal of Pure and Applied Algebra*, 84(2):145–158, February 1993.

[10] L. Cardelli and A. D. Gordon. Mobile ambients. In *Foundations of Software Science and Computation Structures, FoSSaCS '98*. Springer Verlag, 1998.

[11] G. Castagna and F. Zappa Nardelli. The Seal calculus revisited. In *Foundations of Software Technology and Theoretical Computer Science, FST&TCS '02*, volume 2556 of *LNCS (Lecture Notes in Computer Science)*, pages 85–96. Springer, 2002.

[12] A. Corradini and F. Gadducci. A 2-categorical presentation of term graph rewriting. In *Category Theory and Computer Science, CTCS '97*, volume 1290 of *LNCS (Lecture Notes in Computer Science)*, pages 87–105. Springer, 1997.

[13] A. Corradini, R. Heckel, and U. Montanari. Compositional SOS and beyond: a coalgebraic view of open systems. *Theoretical Computer Science*, 280:163–192, 2002.

[14] H. Ehrig, G. Engels, H.-J. Kreowski, and G. Rozenberg, editors. *Handbook of Graph Grammars and Computing by Graph Transformation, Volume 2: Applications, Languages and Tools*. World Scientific, 1999.

[15] H. Ehrig, M. Gajewsky, and F. Parisi-Presicce. High-level replacement systems with applications to algebraic specifications and Petri Nets. In H. Ehrig, H.-J. Kreowski, U. Montanari, and G. Rozenberg, editors, *Handbook of Graph Grammars and Computing by Graph Transformation*, volume 3, chapter 6, pages 341–400. World Scientific, 1999.

[16] H. Ehrig, A. Habel, H.-J. Kreowski, and F. Parisi-Presicce. Parallelism and concurrency in high-level replacement systems. *Math. Struct. in Comp. Science*, 1, 1991.

[17] H. Ehrig and B. König. Deriving bisimulation congruences in the dpo approach to graph rewriting. In *Foundations of Software Science and Computation Structures FoSSaCS '04*, volume 2987 of *LNCS (Lecture Notes in Computer Science)*, pages 151–166. Springer, 2004.

[18] H. Ehrig, H.-J. Kreowski, U. Montanari, and G. Rozenberg, editors. *Handbook of Graph Grammars and Computing by Graph Transformation, Volume 3: Concurrency, Parallelism and Distribution*. World Scientific, 1999.

[19] H. Ehrig, M. Pfender, and H. Schneider. Graph-grammars: an algebraic approach. In *IEEE Conf. on Automata and Switching Theory*, pages 167–180, 1973.

[20] U. Engberg and M. Nielsen. A calculus of communicating systems with label passing. Technical Report DAIMI PB-208, University of Aarhus, 1986.

[21] M. Fiore, G. L. Cattani, and G. Winskel. Weak bisimulation and open maps. In *Logic in Computer Science, LICS '99*, pages 67–76. IEEE Computer Society Press, 1999.

[22] M. P. Fiore and S. Staton. Comparing operational models of name-passing process calculi. In *Coalgebraic Methods in Computer Science CMCS '04*, ENTCS. Elsevier, 2004. To appear.

[23] M. P. Fiore and D. Turi. Semantics of name and value passing. In *Logic in Computer Science LICS '01*, pages 93–104. IEEE, 2001.

[24] C. Fournet and G. Gonthier. A hierarchy of equivalences for asynchronous calculi. In *International Colloquium on Automata, Languages and Programming, ICALP '98*, volume 1443 of *LNCS (Lecture Notes in Computer Science)*. Springer, 1998.

[25] F. Gadducci and R. Heckel. An inductive view of graph transformation. In *Recent Trends in Algebraic Development Techniques*, volume 1376 of *LNCS (Lecture Notes in Computer Science)*, pages 219–233. Springer, 1998.

[26] F. Gadducci and U. Mon. The tile model. In G. Plotkin, C. Stirling, and M. Tofte, editors, *Proof, Language and Interaction: Essays in honour of Robin Milner*. MIT Press, 2000.

[27] F. Gadducci and U. Montanari. Enriched categories as models of computations, 1996.

[28] J. C. Godskesen, T. Hildebrandt, and V. Sassone. A calculus of mobile resources. In *International Conference on Concurrency Theory, Concur '02*, volume 2421 of *LNCS (Lecture Notes in Computer Science)*, pages 272–287. Springer, 2002.

[29] M. Hennessy. *Algebraic Theory of Process Languages*. MIT Press, 1988.

[30] M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM*, 32(1):137–161, 1985.

[31] K. Honda and N. Yoshida. On reduction-based process semantics. *Theoretical Computer Science*, 151(2):437–486, 1995.

[32] D. J. Howe. Proving congruence of bisimulation in functional programming languages. *Information and Computation*, 124(2):103–112, 1996.

[33] O. H. Jensen. *TBA.* PhD thesis, University of Aalborg, 2004. To appear.

[34] O. H. Jensen and R. Milner. Bigraphs and mobile processes. Technical Report 570, Computer Laboratory, University of Cambridge, 2003.

[35] A. Joyal, M. Nielsen, and G. Winskel. Bisimulation from open maps. *Information and Computation*, 127(2):164–185, 1996.

[36] A. Joyal, R. Street, and D. Verity. Traced monoidal categories. *Mathematical Proceedings of the Cambridge Philosophical Society*, 119(3):447–468, 1996.

[37] P. Katis, N. Sabadini, and R. F. C. Walters. Bicategories of processes. *Journal of Pure and Applied Algebra*, 115:141–178, 1997.

[38] P. Katis, N. Sabadini, and R. F. C. Walters. Span(Graph):an algebra of transition systems. In *International Conference on Algebraic Methodology and Software Technology AMAST '97*, number 1349 in LNCS (Lecture Notes in Computer Science), pages 322–336. Springer, 1997.

[39] G. M. Kelly. Elementary observations on 2-categorical limits. *Bull. Austral. Math. Soc.*, 39:301–317, 1989.

[40] G. M. Kelly and R. H. Street. Review of the elements of 2-categories. *Lecture Notes in Mathematics*, 420:75–103, 1974.

[41] B. Klin. *An Abstract Coalgebraic Approach to Process Equivalence for Well-Behaved Operational Semantics.* PhD thesis, BRICS, University of Aarhus, 2004.

[42] B. Klin and P. Sobociński. Syntactic formats for free: An abstract approach to process equivalence. In *International Conference on Concurrency Theory Concur '03*, volume 2620 of *LNCS (Lecture Notes in Computer Science)*, pages 72–86. Springer, 2003.

[43] H.-J. Kreowski. Transformations of derivation sequences in graph grammars. In *LNCS (Lecture Notes in Computer Science)*, volume 56, pages 275–286, 1977.

[44] S. Lack and P. Sobociński. Adhesive categories. In *Foundations of Software Science and Computation Structures FoSSaCS '04*, volume 2987 of *LNCS (Lecture Notes in Computer Science)*, pages 273–288. Springer, 2004.

[45] K. G. Larsen. A context dependent equivalence between processes. *Theoretical Computer Science*, 49:185–215, 1987.

[46] F. W. Lawvere. Some thoughts on the future of category theory. In *Category Theory*, volume 1488 of *Lecture Notes in Mathematics*, pages 1–13, Como, 1991. Springer-Verlag.

[47] J. Leifer. *Operational congruences for reactive systems.* Phd thesis, University of Cambridge, 2001.

[48] J. Leifer and R. Milner. Deriving bisimulation congruences for reactive systems. In *International Conference on Concurrency Theory Concur '00*, volume 1877 of *LNCS (Lecture Notes in Computer Science)*, pages 243–258. Springer, 2000.

[49] S. Mac Lane and R. Paré. Coherence for bicategories and indexed categories. *Journal of Pure and Applied Algebra*, pages 59–80, 1985.

[50] D. Masulovic and J. Rothe. Towards weak bisimulation for coalgebras. *Electronic Notes in Theoretical Computer Science*, 68(1), 2003.

[51] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.

[52] R. Milner. Calculi for interaction. *Acta Informatica*, 33(8):707–737, 1996.

[53] R. Milner. *Communicating and Mobile Systems: the Pi-calculus*. Cambridge University Press, 1999.

[54] R. Milner. Bigraphical reactive systems. In *International Conference on Concurrency Theory Concur '01*, volume 2154 of *LNCS (Lecture Notes in Computer Science)*, pages 16–35. Springer, 2001.

[55] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, (Parts I and II). *Information and Computation*, 100:1–77, 1992.

[56] R. Milner and D. Sangiorgi. Barbed bisimulation. In *9th Colloquium on Automata, Languages and Programming, ICALP92*, volume 623 of *LNCS (Lecture Notes in Computer Science)*, pages 685–695. Springer, 1992.

[57] U. Montanari and V. Sassone. Dynamic bisimulation. Technical report, Univerisità di Pisa, 1990.

[58] U. Montanari and V. Sassone. Dynamic congruence vs. progressing bisimulation for CCS. *Fundamenta Informaticae*, XVI:171–199, 1992.

[59] M. Nygaard. *Domain Theory for Concurrency*. PhD thesis, BRICS, University of Aarhus, 2003.

[60] M. Nygaard and G. Winskel. HOPLA - A higher-order process language. In *International Conference on Concurrency Theory Concur '02*, volume 2421 of *LNCS (Lecture Notes in Computer Science)*, pages 434–448. Springer, 2002.

[61] M. Nygaard and G. Winskel. Full abstraction for HOPLA. In *International Conference on Concurrency Theory Concur '03*, volume 2761 of *LNCS (Lecture Notes in Computer Science)*, pages 378–392. Springer, 2003.

[62] M. Nygaard and G. Winskel. Domain theory for concurrency. *Theoretical Computer Science*, 316:152–190, 2004.

[63] D. Park. Concurrency on automata and infinite sequences. In P. Deussen, editor, *Conf. on Theoretical Computer Science*, volume 104 of *LNCS (Lecture Notes in Computer Science)*. Springer, 1981.

[64] G. D. Plotkin. A structural approach to operational semantics. Technical Report FN-19, DAIMI, Computer Science Department, Aarhus University, 1981.

[65] E. P. Robinson and G. Rosolini. Categories of partial maps. *Information and Computation*, 79, 1988.

[66] G. Rozenberg, editor. *Handbook of Graph Grammars and Computing by Graph Transformation, Volume 1: Foundations*. World Scientific, 1997.

[67] J. J. M. M. Rutten. Universal coalgebra: a theory of systems. *Theoretical Computer Science*, 249:3–80, 2000.

[68] D. Sangiorgi and D. Walker. *The π-calculus: a Theory of Mobile Processes*. Cambridge University Press, 2001.

[69] V. Sassone and P. Sobociński. Deriving bisimulation congruences: A 2-categorical approach. *Electronic Notes in Theoretical Computer Science*, 68(2), 2002.

[70] V. Sassone and P. Sobociński. Deriving bisimulation congruences: 2-categories vs. precategories. In *Foundations of Software Science and Computation Structures FoSSaCS '03*, volume 2620 of *LNCS (Lecture Notes in Computer Science)*. Springer, 2003.

[71] V. Sassone and P. Sobociński. Deriving bisimulation congruences using 2-categories. *Nordic Journal of Computing*, 10(2):163–183, 2003.

[72] V. Sassone and P. Sobociński. Congruences for contextual graph-rewriting. Technical Report RS-04-11, BRICS, University of Aarhus, June 2004.

[73] V. Sassone and P. Sobociński. Locating reaction with 2-categories. *Theoretical Computer Science*, 2004. To appear.

[74] S. H. Schanuel. Negative sets have Euler characteristic and dimension. In *Category Theory*, volume 1488 of *Lecture Notes in Mathematics*, pages 379–385, Como, 1991. Springer-Verlag.

[75] R. A. G. Seely. Modelling computations: a 2-categorical framework. In *Logic in Computer Science LICS '87*. IEEE Computer Society, 1987.

[76] P. Sewell. From rewrite rules to bisimulation congruences. *LNCS (Lecture Notes in Computer Science)*, 1466:269–284, 1998.

[77] P. Sewell. Working note PS14, March 2000. Unpublished.

[78] P. Sobociński. *Deriving process congruences from reaction rules*. PhD thesis, BRICS, University of Aarhus, 2004. Submitted.

[79] R. H. Street. Fibrations in bicategories. *Cahiers de topologie et géométrie différentielle*, XXI-2:111–159, 1980.

[80] R. H. Street. Categorical structures. In M. Hazewinkel, editor, *Handbook of algebra*, volume vol. 1, pages 529–577. North-Holland, 1996.

[81] D. Turi and G. Plotkin. Towards a mathematical operational semantics. In *Logic in Computer Science, LICS'97*, pages 280–291. IEEE Computer Society Press, 1997.

[82] R. J. van Glabbeek. The linear time - branching time spectrum II: The semantics of sequential processes with silent moves. In *International Conference on Concurrency Theory, Concur '93*, volume 715 of *LNCS (Lecture Notes in Computer Science)*, pages 66–81. Springer, 1993.

[83] R. J. van Glabbeek. The linear time – branching time spectrum I. In J. A. Bergstra, A. Ponse, and S. Smolka, editors, *Handbook of Process Algebra*. Elsevier, 1999.

[84] J. Vitek and G. Castagna. A calculus of secure mobile computations. In *IEEE Workshop on Internet Programming Languages*, 1998.

THE FORMAL LANGUAGE THEORY COLUMN

BY

ARTO SALOMAA

Turku Centre for Computer Science
University of Turku
Lemminkäisenkatu 14, 20520 Turku, Finland
asalomaa@it.utu.fi

QUASIPERIODIC INFINITE WORDS: SOME ANSWERS

F. Levé, G. Richomme
LaRIA, UPJV, France
{Florence.Leve,Gwenael.Richomme}@u-picardie.fr

Abstract

We answer some questions about infinite quasiperiodic words asked by Marcus in Bulletin 82 of the European Assocation of Theoretical Computer Science.

1 Introduction

The notion of repetition in Strings is central in a lot of researches (see for instance [7],[8]). In this vein, Apostolico and Ehrenfeucht introduced the notion of quasiperiodic finite words [2] in the following way: "a string z is quasiperiodic if there is a second string $w \neq z$ such that every position of z falls within some occurrence of w in z". The reader can consult [1] for a short survey of studies concerning quasiperiodicity. In [10], Marcus extends this notion to infinite words and he leaves open six questions. We bring answers to the fourth first questions. The questions will be recalled while treating them.

After some generalities, in Section 3, we recall the notion of quasiperiodic (finite or infinite) words. Section 4 answers the first question providing an example of Sturmian words which is not quasiperiodic. In answer to the second question, Section 5 shows that quasiperiodic words can have an exponential complexity. Answers to the third and fourth questions are given in Section 7. They are based on a characterization of the set of quasiperiods of the Fibonacci word stated in Section 6. In conclusion, we briefly consider the two last questions.

2 Generalities

We assume the reader is familiar with combinatorics on words and morphisms (see, e.g., [8]). We precise our notations.

Given an alphabet A (a non-empty set of letters), A^* is the set of finite words over A including the empty word ε. The length of a word w is denoted by $|w|$. A word u is a *factor* of w if there exist words p and s such that $w = pus$. If $p = \varepsilon$ (resp. $s = \varepsilon$), u is a *prefix* (resp. *suffix*) of w. A word u is a *border* of a word w if u is both a prefix and a suffix of w. A factor u of a word w is said *proper* if $w \neq u$.

Given an alphabet A, a(n endo)*morphism* f on A is an application from A^* to A^* such that $f(uv) = f(u)f(v)$ for any words u, v over A. A morphism on A is entirely defined by the images of elements of A. Given a morphism f, *powers* of f are defined inductively by $f^0 = Id$, $f^i = f f^{i-1}$ for integers $i \geq 1$ (composition of applications is denoted just by juxtaposition). When for a letter a, $f(a) = ax$ with $x \neq \varepsilon$, for all $n \geq 0$, $f^n(a)$ is a prefix of $f^{n+1}(a)$. If moreover, for all $n \geq 0$, $|f^n(a)| < |f^{n+1}(a)|$, the limit $\lim_{n\to\infty} f^n(a)$ is the infinite word denoted $f^\omega(a)$ having all the $f^n(a)$ as prefixes. This limit is also a fixed point of f.

3 Quasiperiodicity

We need both definitions of finite and infinite quasiperiodic words.

Let us take from [3] definitions in the finite case. A string w *covers* another string z if for every $i \in \{1, \ldots, |z|\}$, there exists $j \in \{1, \ldots, |w|\}$ such that there is an occurrence of w starting at position $i - j + 1$ in string z. Alternatively we say that w is a *quasiperiod* of z. If z is covered by $w \neq z$, then z is *quasiperiodic*. A string z is *superprimitive* if it is not quasiperiodic (Marcus [10] calls minimal such words). One can observe that any word of length 1 is not quasiperiodic. The string

$$z = abaababaabaababaaba$$

has *aba*, *abaaba*, *abaababaaba* as quasiperiods. Only *aba* is superprimitive. More generally in [3], it is proved that any quasiperiodic word has exactly one super-

primitive quasiperiod. This is a consequence of the fact that any quasiperiod of a finite word w is a proper border of w.

When defining infinite quasiperiodic words, instead of considering the starting indices of the occurrences of a quasiperiod, for convenience, we choose to consider the words preceding the occurrences of a quasiperiod. An infinite word $\underline{w}$ is *quasiperiodic* if there exist a finite word x and words $(p_n)_{n\geq 0}$ such that $p_0 = \varepsilon$ and, for $n \geq 0$, $0 < |p_{n+1}| - |p_n| \leq |x|$ and $p_n x$ is a prefix of $\underline{w}$. We say that x *covers* $\underline{w}$. The word x is also called a *quasiperiod* and we say that the sequence $(p_n x)_{n\geq 0}$ is *a covering sequence of prefixes of the word* $\underline{w}$. In [9], Marcus proves that any infinite word having all finite words as factors is not quasiperiodic. In [10], several examples of quasiperiodic words are given. They have all the form

$$(rs)^{i_1} r (rs)^{i_2} r \dots$$

for two words $r, s \neq \varepsilon$ and non-zero integers $(i_n)_{n\geq 1}$. Let us give another example of quasiperiodic word which does not follow this form. For this, for $k \geq 2$, we consider the endomorphism φ_k defined on $\Sigma_k = \{a_1, \dots, a_k\}$ by $\varphi_k(a_i) = a_1 a_{i+1}$ if $i \neq k$ and $\varphi_k(a_k) = a_1$. This morphism extends the well-known Fibonacci morphism defined on $\{a, b\}^*$ by $\varphi(a) = ab$ and $\varphi(b) = a$. Marcus mentioned that the Fibonacci word, the fixed point of φ, is quasiperiodic (with quasiperiod $aba = \varphi^2(a)$). More generally, we have:

Lemma 3.1. *For $k \geq 2$, the fixed word $\varphi_k^\omega(a_1)$ is quasiperiodic with quasiperiod $\varphi_k^k(a_1)$.*

Proof. Let $(u_i)_{i\geq 0}$ be the sequence of words defined by $u_0 = \varepsilon$, $u_i = \varphi_k(u_{i-1})a_1$ for $i \geq 1$. The reader can verify the following properties for $1 \leq i \leq k$:

- $u_i = u_{i-1} a_i u_{i-1}$. In particular, u_j is a prefix of u_i when $0 \leq i \leq j \leq k$. (We can also note that u_i is a palindrom.)
- $\varphi_k^i(a_j) = u_i a_{i+j}$ for $1 \leq j \leq k - i$, $\varphi_k^i(a_j) = u_i u_{i-1+j-k}^{-1}$ for $k - i < j \leq k$, where uv^{-1} denotes the word w such that $u = wv$ (this notation can be used only if v is a suffix of u).

Consequently we can observe that $\varphi_k^k(a_1) = u_k u_0^{-1} = u_k$ covers each word $\varphi_k^k(a_j a_1) = u_k u_{j-1}^{-1} u_k$. The word $\varphi_k^\omega(a_1)$ can be decomposed over $\{\varphi_k^k(a_1), \varphi_k^k(a_2 a_1), \dots, \varphi_k^k(a_k a_1)\}$. So $\varphi_k^k(a_1)$ covers $\varphi_k^\omega(a_1)$. □

We end this section with another definitions. We will need to consider infinite words covered by two words and not only one. We say that the set $\{xa, xb\}$ *covers* $\underline{w}$ if xa and xb are factors of w and there exist words $(p_n x a_n)_{n\geq 0}$ with $a_n \in \{a, b\}$ such that $p_0 = \varepsilon$, and, for $n \geq 0$, $0 < |p_{n+1}| - |p_n| \leq |x| + 1$ and $p_n x a_n$ is a prefix of $\underline{w}$. Once again the sequence $(p_n x a_n)_{n\geq 0}$ is called *a covering sequence of prefixes of the word* $\underline{w}$.

4 About Sturmian words

The first question in [10] is: "Is every Sturmian word quasiperiodic?". Proposition 4.1 below provides a negative answer.

Proposition 4.1. *Not all Sturmian words are quasiperiodic.*

Let us recall that there are several equivalent definitions of Sturmian words (see [4] for instance). A convenient tool to deal with Sturmian words is the set of Sturmian endomorphisms $\{\varphi, \tilde{\varphi}, E\}$ where $\tilde{\varphi}$ and E are defined on $\{a, b\}$ by $\tilde{\varphi}(a) = ba$, $\tilde{\varphi}(b) = a$ and $E(a) = b$, $E(b) = a$. The set $\{\varphi, \tilde{\varphi}, E\}^*$ is exactly the set of morphisms that preserves Sturmian words (the image of a Sturmian word is Sturmian) [12]. It is also well-known that if a Sturmian morphism generates an infinite word then this fixed point is a Sturmian word. Let us consider the Sturmian morphism $E\tilde{\varphi}\varphi E$. Proposition 4.1 is a corollary of the following result:

Lemma 4.2. *The infinite word $(E\tilde{\varphi}\varphi E)^\omega(a)$ is not quasiperiodic.*

Proof. Let $f = E\tilde{\varphi}\varphi E$: $f(a) = ab$, $f(b) = abb$. The proof of this lemma holds by contradiction. Assume that x is a quasiperiod of $f^\omega(a)$ of minimal length. Note that x is a prefix of $f^\omega(a) = ababbababbabb\ldots$ We observe that $|x| \geq 5$. By construction of $f^\omega(a)$, x ends with abb, ab or a. If x ends with abb, for each word p such that px is a prefix of $f^\omega(a)$, there exist words p', x' such that $p = f(p')$, $x = f(x')$ and $|x'| < |x|$. In fact x' does not depend on p. Consequently we can verify that x' is a quasiperiod of $f^\omega(a)$. This contradicts the choice of x. If x ends with a then xb is also a quasiperiod of $f^\omega(a)$. So we can assume x ends with ab and $f^\omega(a)$ has no quasiperiod of length less than or equal to $|x| - 2$. Let $(p_n)_{n\geq 0}$ be a covering sequence of prefixes of $f^\omega(a)$. Since x starts with a, there exist words $(p'_n)_{n\geq 0}$ and a unique word x' such that $p_n = f(p'_n)$ and $x = f(x')ab$. Moreover from $|x| \geq 5$, we deduce $|x'a| < |x| - 2$. Consequently $x'a$ cannot be a quasiperiod of $f^\omega(a)$. It follows that xa and xb are factors of $f^\omega(a)$ (otherwise $xb = f(x'b)$ or $x = f(x'a)$ and we can deduce that $x'b$ or $x'a$ is a quasiperiod of $f^\omega(a)$). So $\{xa, xb\}$ covers $f^\omega(a)$.

Let y be a non-empty word such that $\{ya, yb\}$ covers $f^\omega(a)$. Note that y must end with ab. Let $(p_n y a_n)_{n\geq 0}$ (with $a_n \in \{a, b\}$ for all $n \geq 0$) be a covering sequence of prefixes of $f^\omega(a)$. Since $|y| \geq 2$, y starts with ab. Consequently there exist words $(p'_n)_{n\geq 0}$ and y' such that $p_n = f(p'_n)$ and $y = f(y')ab$. Moreover $(p'_n y' a_n)_{n\geq 0}$ is a covering sequence of prefixes of $f^\omega(a)$. From what precedes, it follows that the word y is one of the words x_n defined by $x_0 = \varepsilon$ and $x_n = f(x_{n-1})$ for $n \geq 1$. Note that we can see by induction that for all $n \geq 1$, $x_n b$ has no proper suffix which is a prefix of x_n.

Let us consider again the quasiperiod x. The word xb is a factor of $f^\omega(a)$. Since x covers $f^\omega(a)$ and since x starts with a, the word xb has a proper suffix which is a prefix of x. Since $x \neq \varepsilon$, this contradicts what was said about the x_n's.

So $(E\tilde{\varphi}\varphi E)^\omega(a)$ has no quasiperiod. □

Let us observe that the word $\underline{w}$ such that $(E\tilde{\varphi}\varphi E)^\omega(a) = a\underline{w}$ starts with ba and can be decomposed over $\{ba, bba\}$. So it is quasiperiodic with quasiperiod bab.

5 Complexity

The second question of [10] is "What about the complexity function of a quasiperiodic infinite word?". Let recall that the complexity function $p_{\underline{w}}(n)$ of an infinite word $\underline{w}$ is the function which associates to each integer $n \geq 0$ the number of factors of length n of $\underline{w}$. Our aim is to show that there exists no relation between quasiperiodic words and complexities.

First we consider words with lowest complexity. It is well known (see [8] for instance) that a word $\underline{w}$ has a bounded complexity if and only if $\underline{w} = uv^\omega$ for words $u, v \neq \varepsilon$. When $u = \varepsilon$, v is a quasiperiod of $\underline{w}$. When $u \neq \varepsilon$, $\underline{w}$ can be quasiperiodic as for instance $ab(aba)^\omega$ or it can be non-quasiperiodic as for instance ab^ω.

In [11], it is shown that Sturmian words are the words with lowest unbounded complexity. We know there exist quasiperiodic (the Fibonacci word [10]) and non-quasiperiodic Sturmian words (Section 4).

In [5], Cassaigne characterizes couples of integers (α, β) for which there exists an integer $n_0 \geq 0$ and an infinite word over $\{a, b\}$ having complexity $\alpha n + \beta$ for all $n \geq n_0$. They are the couples in $\{0, 1\} \times (\mathbb{N} \setminus \{0\}) \cup (\mathbb{N} \setminus \{0, 1\}) \times \mathbb{Z}$. When $\alpha \geq 1$, the word given for example by Cassaigne is a quasiperiodic word. More precisely, it is the word $g_{l,j}(\varphi_k^\omega(a_1))$ where j, l, k are suitable integers, $g_{l,j}$ is the morphism defined by $g_{l,j}(a_i) = a^l b^{i+j}$ and φ_k is the morphism defined in Section 3. We leave open the question to find non-quasiperiodic words with these complexities when $\alpha \notin \{0, 1\}$.

We end this section showing that there exist quasiperiodic words with exponential complexity. As already said, in [9], it is shown that all words having all finite words as factors are not quasiperiodic. These words are those with complexity $p(n) = 2^n$ for all $n \geq 0$.

Let $\underline{w}$ be such a word over $\{a, b\}$. Since $\varphi^2(a) = aba$ and $\varphi^2(b) = ab$, the word $\varphi^2(\underline{w})$ is quasiperiodic with quasiperiod aba.

We now evaluate the complexity of the word $\varphi^2(\underline{w})$. For this, let a_n (resp. b_n, c_n) be the number of factors of $\varphi^2(\underline{w})$ ending with b (resp. ba, aa). We have $p(0) = 1$, $p(1) = 2$, $p_{\underline{w}}(n) = a_n + b_n + c_n$ for $n \geq 2$. Since a^3 and b^2 are not factors of

$\varphi^2(\underline{w})$, we have $a_2 = b_2 = c_2 = 1$ and for $n \geq 2$, $a_{n+1} = b_n + c_n$, $b_{n+1} = a_n$, $c_{n+1} = b_n$. Consequently for $n \geq 3$, $a_{n+1} = a_{n-1} + a_{n-2}$, $b_{n+1} = b_{n-1} + b_{n-2}$, $c_{n+1} = c_{n-1} + c_{n-2}$. So $p(2) = 3$ and for $n \geq 3$, $p(n+1) = p(n-1) + p(n-2)$.

The first values of the sequence $(p(n))_{n\geq 1}$ are:

$$2, 3, 4, 5, 7, 9, 12, 16, 21, 28, 37, 49, \ldots$$

This sequence is part of the Padovan sequence (see sequence A000931 in [13]) defined by $a_0 = 1$, $a_1 = 0$, $a_2 = 0$ and for $n \geq 3$ $a_n = a_{n-2} + a_{n-3}$ (more precisely $p(n) = a_{n+8}$ for $n \geq 1$). It is known (see [13] for instance) that a_n is asymptotic to $r^n/(2 * r + 3)$ where $r = 1.3247179572447\ldots$, is the real root of $x^3 = x + 1$ (r is called the plastic constant [14]). So $p(n) = \theta(r^n)$.

To end with complexity, let us quote a new question: what is the maximal complexity of a quasiperiodic infinite word?

6 Quasiperiods of the Fibonacci word

In order to answer other questions of [10], we characterize the quasiperiods of the Fibonacci word (Proposition 6.5). In particular, we show that this word has an infinite number of superprimitive quasiperiods (Proposition 6.6). We start with a useful lemma:

Lemma 6.1. *Let $\underline{w}$ be an infinite word over $\{a, b\}$ and let $x \in \{a, b\}^*$.*

If x is a quasiperiod of $\varphi(\underline{w})$ then x verifies one of the three following properties:

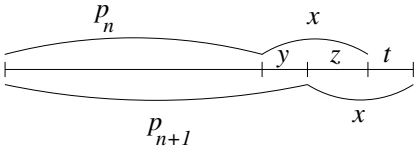
a) $x = \varphi(x')$ where x' is a quasiperiod of $\underline{w}$.

b) $x = \varphi(x')a$ where x' is a quasiperiod of $\underline{w}$.

c) $x = \varphi(x')a$ for a word x' such that $\{x'a, x'b\}$ covers $\underline{w}$.

Proof. Assume that x is a quasiperiod of $\varphi(\underline{w})$. Since $\varphi(\underline{w})$ starts with the letter a and since x is a prefix of $\varphi(\underline{w})$, x starts with a.

First we consider the case where x ends with the letter b. Since x is a factor of $\varphi(\underline{w})$, x ends with ab. Let $(p_n x)_{n\geq 0}$ be a covering sequence of prefixes of $\varphi(\underline{w})$. Let $n \geq 0$. We have $|p_n| < |p_{n+1}| \leq |p_n x|$. So $|p_{n+1}| \leq |p_n x| < |p_{n+1} x|$ and the following situation holds.



There exist words y, z, t such that $x = yz = zt$. Note $|y| = |p_{n+1}| - |p_n| \neq 0$. Since x starts with a and ends with b, there exist words y', z', t', p'_n such that $y = \varphi(y')$, $z = \varphi(z')$, $t = \varphi(t')$, $p_n = \varphi(p'_n)$ and $p_{n+1} = \varphi(p'_n y')$. Since p_n is a prefix of p_{n+1} which is itself a prefix of $p_n x$, we deduce $|p'_n| < |p'_{n+1}| \leq |p'_n x'|$ where $x' = y'z'$. Since φ is injective, we also have $x' = z't'$.

What precedes is valid for all $n \geq 0$ and due to injectivity of φ, the words $(p'_n)_{n\geq 0}$ and x' are defined uniquely. So $(p'_n x')_{n\geq 0}$ is a covering sequence of $\underline{w}$, that is, x' is a quasiperiod of $\underline{w}$. Moreover $x = \varphi(x')$.

(Note that since x ends with b, x' ends with a.)

Now we consider the case where x ends with the letter a.

If xa is a quasiperiod of $\varphi(\underline{w})$, one can see as previously that $x = \varphi(x')$ for a quasiperiod x' of $\underline{w}$.

If each occurrence of x is followed by an occurrence of the letter b, then x ends with the letter a. Since it also starts with a, there exists a word x' such that $x = \varphi(x')a$. Moreover in this current case, xx cannot be a factor of $\varphi(\underline{w})$. So $\varphi(x')$ is a quasiperiod of $\varphi(\underline{w})$. We can deduce that x' is a quasiperiod of $\underline{w}$.

Finally we have to consider the case where x ends with the letter a and some occurrences of x are followed by a and others by b. Thus we cannot say that xa or xb is a quasiperiod of $\underline{w}$. Let $(p_n x)_{n\geq 0}$ be a covering sequence of prefixes of $\varphi(\underline{w})$. For $n \geq 0$, let a_n be the letter such that $p_n x a_n$ is a prefix of $\varphi(\underline{w})$. Let b_n be the letter in $\{a, b\} \setminus \{a_n\}$. There exist a word x' and prefixes p'_n of $\underline{w}$ such that $p_n = \varphi(p'_n)$, $x = \varphi(x')a$. Moreover $xa_n = \varphi(x'b_n)$ when $a_n = b$ and $x = \varphi(x'b_n)$ when $a_n = a$. We can deduce that $(p'_n x' b'_n)_{n\geq 0}$ is a covering sequence of prefixes of w, that is, $\{x'a, x'b\}$ covers $\underline{w}$. □

The converse of Lemma 6.1 partially holds. We let the reader verify that:

Lemma 6.2. *If a word x verifies Case a or b in Lemma 6.1 then x is a quasiperiod of $\underline{w}$.*

This does not hold if x fulfils Case c. We can only deduce that $\{xa, xb\}$ covers $\varphi(\underline{w})$. Indeed there exist words $\underline{w}$, x' such that $x = \varphi(x'a)$ does not cover $\varphi(\underline{w})$. For instance if $x' = ab$ and $\underline{w} = abaaba(bba)^\omega$ then $\{x'a, x'b\} = \{aba, abb\}$ covers $\underline{w}$. But $\varphi(x')a = abaa$ does not cover $\varphi(\underline{w}) = abaabaabaaba(baaa)^\omega$.

In order to characterize the quasiperiods of the Fibonacci word $\underline{F}$, we need to study what happens when Case c of Lemma 6.1 occurs.

Lemma 6.3. *The words y such that $\{ya, yb\}$ covers $\underline{F}$ are the words $(u_n)_{n\geq 0}$ defined by $u_0 = \varepsilon$, $u_{n+1} = \varphi(u_n)a$ for $n \geq 1$.*

Moreover for $n \geq 2$, u_n is a quasiperiod of $\underline{F}$.

The proof of this lemma is a direct consequence of the following result and of the fact that $u_2 = aba$ is a quasiperiod of $\underline{F}$.

Lemma 6.4. *Let $\underline{w}$ be an infinite word over $\{a, b\}$ that does not contain bb as factor. Let $y \in \{a, b\}^*$ such that $|y| \geq 1$.*

The set $\{ya, yb\}$ covers $\varphi(\underline{w})$ if and only if $y = \varphi(z)a$ and $\{za, zb\}$ covers w.

Proof. First assume that $\{za, zb\}$ covers w. Since each occurrence of $\varphi(zb)$ in $\varphi(\underline{w})$ is followed by the letter a, the set $\{\varphi(za), \varphi(zb)a\} = \{\varphi(z)ab, \varphi(z)aa\}$ covers $\varphi(\underline{w})$.

Assume now that $\{ya, yb\}$ covers $\varphi(w)$ and $|y| \geq 1$. The word y starts and ends with the letter a. Let $(p_n ya_n)_{n\geq 0}$ be a sequence of prefixes covering $\varphi(\underline{w})$ with $a_n \in \{a, b\}$. There exist words p'_n and z such that $p_n = \varphi(p'_n)$ and $y = \varphi(z)a$. For $n \geq 0$, let $b_n = a$ if $a_n = b$ and $b_n = b$ if $a_n = a$. Let $n \geq 0$. By definition of a covering sequence of prefixes, $|p_n| < |p_{n+1}| \leq |p_n ya_n|$. If $a_n = b$, $|\varphi(p'_n)| < |\varphi(p'_{n+1})| \leq |\varphi(p'_n zb_n)|$. Since p'_n is a prefix of p_{n+1} itself a prefix of $p'_n zb_n$, $|p'_n| < |p'_{n+1}| \leq |p'_n zb_n|$. Observe now that *yaya* is not a factor of $\varphi(\underline{w})$ (Indeed since y starts and ends with a, this would imply that aaa is a factor of $\varphi(\underline{w})$ but this is not possible since bb is not a factor of $\underline{w}$). Thus when $a_n = a$, $|p_n| < |p_{n+1}| < |p_n ya_n|$, that is, $|p_n| < |p_{n+1}| \leq |p_n y| = |\varphi(p'_n zb_n)|$. Once again $|p'_n| < |p'_{n+1}| \leq |p'_n zb_n|$. So the sequence $(p'_n zb_n)_{n\geq 0}$ is a covering sequence of prefixes of $\underline{w}$, that is, $\{za, zb\}$ covers $\underline{w}$. □

Now we can describe the set of quasiperiods of the Fibonacci word. Let $Q_0 = \{aba\}$ and for $n \geq 1$ $Q_n = \{\varphi(u), \varphi(u)a \mid u \in Q_{n-1}\}$.

Proposition 6.5. *The set of quasiperiods of the Fibonacci word is* $\bigcup_{n\geq 0} Q_n$.

The proof of this proposition is a consequence of Lemmas 6.1 and 6.3.

To illustrate the previous proposition, let us note that the first quasiperiods of $\underline{F}$ are: *aba*, *abaab*, *abaaba*, *abaababa*, *abaababaa*, *abaababaab*, *abaababaaba*.

Let $f_n = \varphi^n(a)$ for $n \geq 0$. We can see by induction that, for any integer $n \geq 2$, $Q_n = \{f_{n+2} f_{i_1} f_{i_2} \dots f_{i_k} \mid 0 \leq k \leq n, n-1 \geq i_1 > i_2 > \dots i_k = 0\}$. So Q_n contains 2^n distinct elements. Moreover for $x \in Q_n$, $|f_{n+2}| \leq |x| < |f_{n+3}|$. It follows that $\bigcup_{i=0}^{n} Q_i$ has $2^{n+1} - 1$ distinct elements each of length less than $|f_{n+3}|$.

The reader interested by similar results can consult [6] that provides a description of the quasiperiods of the words f_k considered as circular.

Proposition 6.5 shows in particular that the Fibonacci word has an infinite number of quasiperiods. This could have been obtained observing the particular quasiperiods $\varphi^n(aba)$ for $n \geq 0$.

We now want to state a great difference between quasiperiodic infinite words and quasiperiodic finite words. We have already recalled that any quasiperiodic finite word has a unique superprimitive quasiperiod. We show that the Fibonacci word has an infinite number of superprimitive quasiperiods.

Proposition 6.6. *The set of quasiperiods of $\underline{F}$ are the words $(q_n)_{n\geq 0}$ defined (for $n \geq 0$) by:*

$$q_{2n} = f_{2n+1} \prod_{i=0}^{n} f_{2(n-i)}$$

$$q_{2n+1} = f_{2n+2} \prod_{i=0}^{n} f_{2(n-i)+1}$$

Before proving this proposition let us give the first superprimitive quasiperiods: $q_0 = f_1 f_0 = aba$, $q_1 = f_2 f_1 = abaab$, $q_2 = f_3 f_2 f_0 = abaababaa$, $q_3 = f_4 f_3 f_1 = abaababaabaabab$.

Proof of Proposition 6.6. First we note that all the words q_n are quasiperiods of $\underline{F}$. Indeed $q_0 = aba \in Q_0$ and for $n \geq 0$, $q_{2n+1} = \varphi(q_{2n}) \in Q_{2n+1}$ and $q_{2n+2} = \varphi(q_{2n+1})a \in Q_{2n+2}$. Observe that the sequence $(q_n)_{n\geq 0}$ is a sequence of length increasing words.

We now want to prove that q_n does not cover q_m for any $n < m$. For this note that if n is odd and m is even, q_n ends with b and q_m ends with a. So q_n does not cover q_m. Case n even and m odd is similar. Assume now $n = 2p$ and $m = 2q$. The word q_{2n} ends with $ab \prod_{i=0}^{n} f_{2(n-i)}$ whereas the word q_{2m} ends with $\prod_{i=0}^{n+1} f_{2(n+1-i)}$ and so with $ba \prod_{i=0}^{n} f_{2(n-i)}$. Once again q_{2n} cannot cover q_{2m}. Case n and m both odd is similar.

To end the proof of Proposition 6.6, we need to see that $\underline{F}$ does not have a quasiperiod x that covers q_n for an integer $n \geq 0$ with $|x| < |q_n|$. This can be stated showing by induction that the set of superprimitive quasiperiods of $\underline{F}$ that belong to Q_n is $\{q_i \mid 0 \leq i \leq n\}$. This is a consequence of Lemma 6.2. □

7 About set of quasiperiods

In this section, we consider the third and four questions in [10]. The third one is an open question: "What about the set of quasiperiods of an infinite word?".

As seen in previous section, there exists at least one word (the Fibonacci word) which has an infinite number of quasiperiods. It is easy to construct other such

examples. Indeed taking two infinite words u and v over $\{a, b\}$ having the same quasiperiod z starting with the letter a, considering the morphism f defined by $f(a) = u$, $f(b) = v$, the fixed point $f^\omega(a)$ has the quasiperiods $f^n(z)$ for any $n \geq 0$. Indeed for any word w having a quasiperiod x, $f(w)$ has $f(x)$ as quasiperiod.

We now want to show that there are a lot of intermediate cases between infinite words having no quasiperiod [10] and infinite words having an infinite number of superprimitive quasiperiods (as the Fibonacci word).

Lemma 7.1. *Let $\underline{w} = abaaba(bba)^\omega$. For $n \geq 0$, the word $\varphi^{n+1}(\underline{w})$ has $\bigcup_{i=0}^{n} Q_i$ as set of quasiperiods and $q_0, \ldots, q_n$ as superprimitive quasiperiods.*

We let the reader prove this result using lemmas of the previous section.

In [10], Marcus defines "the quasiperiodicity of order p that where the intersection of two different occurrences of a superprimitive quasiperiod is never larger than p, but sometimes it is equal to p". He provides examples of words for each positive order. He asks: "Does there exist a quasiperiodic infinite word which is of no order p ($p = 1, 2, 3, \ldots$)? The Fibonacci word provides a positive answer.

Lemma 7.2. *The Fibonacci word has no order.*

Proof. The covering sequences of prefixes of $\underline{F}$ associated with the quasiperiod $q_0 = aba$ starts with $\varepsilon, aba, abaab$. In particular the second and third occurrences of aba overlap and the overlap has length $1 = |f_0|$. We have seen in the proof of Proposition 6.6 that $q_{2n} = \varphi(q_{2n-1})$ and $q_{2n+1} = \varphi(q_{2n})a$. Thus by induction on $n \geq 0$, one can see that q_n has two occurrences overlapping with an overlap of length at least $|f_n|$. So the Fibonacci word has no order. □

8 Conclusion

The two last questions of [10] concerns infinite words such that all their factors are also quasiperiodic. These questions should certainly be precised or transformed since any prefix of length 1 of a non-empty word is not quasiperiodic. One possible way to transform Question 6 (Does there exist a non-quasiperiodic infinite word such that all its factors are quasiperiodic?) is to search for a non-quasiperiodic infinite word having an infinity of quasiperiodic prefixes. We provide such an example.

Let $(u_n)_{n\geq 0}$, $(v_n)_{n\geq 0}$ be the sequences of words defined by $u_0 = aa$, and for $n \geq 0$, $v_n = u_n b$, $u_{n+1} = v_n^2$. So $u_0 = aa$, $v_0 = aab$, $u_1 = aabaab$, $v_1 = aabaabb$, $u_2 = aabaabbaabaabb$, $v_2 = aabaabbaabaabbb$. Of course, since each u_n is a square it is a quasiperiodic word. By induction one can see that for any $n \geq 0$, the word

b^{n+1} occurs only once in v_n as a suffix. So v_n is superprimitive. Consequently the word $\lim_{n\to\infty} u_n = \lim_{n\to\infty} v_n$ is a non-quasiperiodic infinite word having infinitely many quasiperiodic prefixes.

Acknowledgments. The authors thank Francis Wlazinski for fruitful discussions.

References

[1] A. Apostolico and M. Crochemore. String pattern matching for a deluge survival kit. In J. Abello, P.M. Pardalos, and M.G.C. Resende, editors, *Handbook of Massive Data Sets*. Kluwer Academic Publishers, 2001.

[2] A. Apostolico and A. Ehrenfeucht. Efficient detection of quasiperiodicities in strings. *Theoretical Computer Science*, 119:247–265, 1993.

[3] A. Apostolico, M. Farach, and C. S. Iliopoulos. Optimal superprimitivity testing for strings. *Information Processing Letters*, 39(1):17–20, 1991.

[4] J. Berstel and P. Séébold. *Algebraic Combinatorics on Words (*M. Lothaire), volume 90, chapter 2. Sturmian words. Cambridge Mathematical Library, 2002.

[5] J. Cassaigne. Complexité et facteurs spéciaux. *Bulletin of the Belgian Mathematical Society*, 4:67–88, 1997.

[6] C. S. Iliopolos, D. Moore, and W. F. Smyth. The covers of a circular Fibonacci string. *J. Combinatorial Math. and Combinatorial Computing*, 26:227–236, 1998.

[7] M. Lothaire. *Algebraic Combinatorics on words*, volume 90 of *Encyclopedia of Mathematics*. Cambridge University Press, Cambridge, UK, 2002.

[8] M. Lothaire. *Applied Combinatorics on Words*. To appear. (see http://www-igm.univ-mlv.fr/˜berstel).

[9] S. Marcus. Bridging two hierarchies of infinite words. *Journal of Universal Computer Science*, 8:292–296, 2002.

[10] S. Marcus. Quasiperiodic infinite words. *Bulletin of the European Association for Theoretical Computer Science (EATCS)*, 82:170–174, 2004.

[11] M. Morse and G.A. Hedlund. Symbolic Dynamics II: Sturmian trajectories. *Amer. J. Math.*, 61:1–42, 1940.

[12] P. Séébold. Fibonacci morphisms and Sturmian words. *Theoretical Computer Science*, 88:365–384, 1991.

[13] N. J. A. Sloane. The on-line encyclopedia of integer sequences. http://www.research.att.com/˜njas/sequences/index.html.

[14] E. W Weisstein. Plastic constant. From *MathWorld* – A Wolfram Web resource. http://mathworld.wolfram.com/PlasticConstant.html.

The Logic in Computer Science Column

by

Yuri Gurevich

Microsoft Research
One Microsoft Way, Redmond WA 98052, USA
gurevich@microsoft.com

Why Sets?

Andreas Blass* Yuri Gurevich

Abstract

Sets play a key role in foundations of mathematics. Why? To what extent is it an accident of history? Imagine that you have a chance to talk to mathematicians from a far away planet. Would their mathematics be set-based? What are the alternatives to the set-theoretic foundation of mathematics? Besides, set theory seems to play a significant role in computer science, in particular in database theory and formal methods. Is there a good justification for that? We discuss these and some related issues.

1 Sets in Computer Science

Quisani: I wonder why sets play such a prominent role in foundations of mathematics. To what extent is it an accident of history? And I have questions about the role of sets in computer science.

*Partially supported by NSF grant DMS–0070723 and by a grant from Microsoft Research. Address: Mathematics Department, University of Michigan, Ann Arbor, MI 48109–1109, U.S.A., ablass@umich.edu. This paper was written during a visit to Microsoft Research.

Authors: Have you studied set theory?

Q: Not really but I came across set theory when I studied discrete mathematics and logic, and I looked into Lévy's book [38] years ago. I remember that ZFC, first-order Zermelo-Fraenkel set theory with the axiom of choice, became for all practical purposes the foundation of mathematics. I can probably reconstruct the ZFC axioms.

A: Do you remember the intuitive model for ZFC.

Q: Let me see. You consider the so-called cumulative hierarchy of sets. It is a transfinite hierarchy, so that you have levels $0, 1, \ldots, \omega, \omega + 1, \ldots$ On the level zero, you have the empty set and possibly some atoms. On any other level α you have the sets of objects that occur on levels $< \alpha$. Intuitively the process never ends. To model ZFC, you just go far enough in this hierarchy so that all axioms are satisfied. Is that correct, more or less?

A: Yes.

Q: This morning I read at the Z users website [50] the following: "The formal specification notation Z (pronounced "zed"), useful for describing computer-based systems, is based on Zermelo-Fraenkel set theory and first order predicate logic." And I was somewhat surprised.

A: Were you surprised that they use the ZF system rather than ZFC, the Zermelo-Fraenkel system with the axiom of choice? As long as we consider only finite families of sets, the axiom of choice is unnecessary. That is, one can prove in ZF that, if X is a finite family of nonempty sets, then there is a function assigning to each set $S \in X$ one of its members. Furthermore, there is a wide class of statements, which may involve infinite sets, but for which one can prove a metatheorem saying that any sentence in this class, if provable in ZFC, is already provable in ZF; see [44, Section 1] for details. This class seems wide enough to cover anything likely to arise in computer science, even in its more abstract parts.

Q: That is an interesting issue in its own right, but I was surprised by something else. Set theory wasn't developed to compute with. It was developed to be a foundation of mathematics.

A: There are many things that were developed for one purpose and are used for another.

Q: Sure. But, because set theory was so successful in foundations of mathematics, there may be an exaggerated expectation of the role that set theory can play in computing. Let me try to develop my thought. What makes set theory so useful in foundations of mathematics? I see two key aspects. One aspect is that the notion of set is intuitively simple.

A: Well, it took time and effort to clarify our intuition about sets and to deal with set-theoretic paradoxes; see for example [21] and [30]. But we agree that the notion of set is intuitively simple.

Q: The other aspect is that set theory is very expressive and succinct: mathematics can be faithfully and naturally translated into set theory. This is extremely important. Imagine that somebody claims an important theorem but you don't understand some notions involved. You can ask the claimer to define the notions more and more precisely. In the final account, the whole proof can be reduced to ZFC, and then the verification becomes mechanical.

Can sets play a similar role in computing? I see a big difference between the reduction to set theory in mathematics and in computing. The mathematicians do not actually translate their stuff into set theory. They just convince themselves that their subject is translatable.

A: Bourbaki [10] made a serious attempt to actually translate a nontrivial portion of mathematics into set theory, but it is an exception.

Q: Right. In computing, such translations have to be taken seriously. If you want to use a high-level language that is compiled to some set-theoretic engine, then a compiler should exist in real life, not only in principle. I guess all this boils down to the question whether the datatype of sets can be the basic datatype so that everything else is interpreted in set theory.

A: There has been an attempt made in this direction [49].

Q: Yes, and most people remained unconvinced that this was the way to go. Sequences, or lists, are appropriate as the basic datastructure.

A: We know one example where sets turned out to be more succinct than sequences as the basic datastructure.

Q: Tell me about it.

A: OK, but bear with us as we explain the background. We consider computations where inputs are finite structures, for example graphs, rather than strings.

Q: Every such structure can be presented as a string.

A: That is true. But we restrict attention to computing properties that depend only on the isomorphism type of the input structure. For example, given a bipartite graph, decide whether it has a matching. Call such properties *invariant queries*.

Q: Why the restriction?

A: Because we are interested in queries that are independent from the way the input structure is presented. Consider, for example, a database query. You want that the result depends on the database only and not on how exactly is it stored.

Q: Fine, what is the problem?

A: The original problem was this: Does there exist a query language L such that

(Restrained) every query that can be formulated in L is an invariant query computable in polynomial time, and

(Maximally expressive) every polynomial-time computable invariant query can be formulated in L.

Q: How can one ensure that all L-queries are invariant?

A: Think about first-order logic as a query language. Every first-order sentence is a query. First-order queries are *pure* in the sense that they give you no means to express a property of the input structure that is not preserved by isomorphisms. Most restrained languages in the literature are pure in that same sense.

Q: But, in principle, can a restrained language allow you to have non-invariant intermediate results? For example, can you compute a particular bipartite matching, throw away the matching and return "Yes, there is a bipartite matching"?

A: Yes, a restrained language may have non-invariant intermediate results. In fact, Ashok Chandra and David Harel, who raised the original problem in [11], considered Turing machines M that are invariant in the following sense: If M accepts one string representation of the given finite structure then it accepts them all. They asked whether there is a decidable set L of invariant polynomial time Turing machines such that, for every invariant polynomial time Turing machine T_1, there is a machine $T_2 \in L$ that computes the same query as T_1 does. In the case of a positive answer, such an L would be restrained and maximally expressive.

Q: Hmm, a decidable set of Turing machines does not look like a language.

A: One of us conjectured [25] that there is no query language, even as ugly a decidable set of Turing machines, that is restrained and maximally expressive.

Q: But one can introduce, I guess, more and more expressive restrained languages.

A: Indeed. In particular, the necessity to deal with invariant database queries led to the introduction of a number of restrained query languages [1] including the polynomial-time version of the language $\texttt{while}_{\texttt{new}}$. In [8], Saharon Shelah and the two of us proposed a query language, let us call it BGS, that is based on set theory. BGS is pure in the sense discussed above. A polynomial time bounded version of BGS, let us call it Ptime BGS, is a restrained query language.

Q: In what sense is BGS set-theoretic?

A: It is convenient to think of BGS as a programming language. A state of a BGS program includes the input structure A, which is finite, but the state itself is an

infinite structure. It contains, as elements, all hereditarily finite sets built from the elements of A. These are sets composed from the elements of A by repeated use of the pairing operation $\{x, y\}$ and the union operation $\bigcup(x) = \{y : \exists z(y \in z \in x)\}$. BGS uses standard set theoretic operations and employs comprehension terms $\{t(x) : x \in r \wedge \varphi(x)\}$. In any case, to make a long story short, it turned out that Ptime BGS was more expressive than the Ptime version of the language $\texttt{while}_{\text{new}}$ that works with sequences; see [9] for details. For the purpose at hand, sets happened to be more efficient than sequences.

Q: I don't understand this. A set s can be easily represented by a sequence of its elements?

A: Which sequence?

Q: Oh, I see. You may have no means to define a particular sequence of the elements of s and you cannot pick an arbitrary sequence because this would violate the purity of BGS.

A: Right. You may want to consider all $|s|!$ different sequences of the elements of s. This does not violate the purity of BGS. But, because of the polynomial time restriction, you may not have the time to deal with $|s|!$ sequences.

On the other hand, a sequence $[a_1, a_2, \ldots, a_k]$ can be succinctly represented by a set $\{[i, a_i] : 1 \leq i \leq k\}$. Ordered pairs have a simple set-theoretic representation due to Kuratowski: $[a, b] = \{\{a, b\}, \{a\}\}$.

Q: I agree that, in your context, sets are more appropriate than sequences.

A: It is also convenient to have the datatype of sets available in software specification languages.

Q: But closer to the hardware level, under the hood so to speak, we cannot deal with sets directly. They have to be represented e.g. by means of sequences.

A: You know hardware better than we do. Can one build computers that deal with sets directly?

Q: A good question. The current technology would not support a set oriented architecture.

A: What about quantum or DNA-based computing?

Q: I doubt that these new paradigms will allow us to deal with sets directly but your guess is as good as mine.

2 Sets in Mathematics

Q: Let me return to the question why sets play such a prominent role in the foundation of mathematics. But first, let me ask a more basic question: Why do we

need foundations at all? Is mathematics in danger of collapsing? Most mathematicians that I know aren't concerned with foundations, and they seem to do OK.

A: Well, you already mentioned the fact that an alleged proof can be made more and more detailed until it becomes mechanically verifiable.

Q: Yes, but I'd hope that this could be done with axioms that talk about all the different sorts of objects mathematicians use — real numbers, functions, sequences, Hilbert spaces, etc. — and that directly reflect the facts that mathematicians routinely use. What's the advantage of reducing everything to sets?

A: We see three advantages. First, people have already explicitly written down adequate axiomatizations of set theory. The same could probably be done for the sort of rich theory that you described, but it would take a nontrivial effort. Second, the reduction of mathematics to set theory means that the philosopher who wants to understand the nature of mathematical concepts needs only to understand one concept, namely sets. Third, when proving that a statement is consistent with ordinary mathematics, one only has to produce a model of set theory in which the statement is true. Without the set theoretic foundation, one would have to construct a model of a much richer theory.

Q: These advantages make sense but they also show why a typical mathematician never has to use the reduction to set theory. Actually, the second advantage is not entirely clear to me; it seems that by reducing mathematics to set theory the philosopher can lose some of its semantic or intuitive content. Consider a proof that complex polynomials have roots, and imagine a set-theoretic formalization of it.

A: It's not a matter of the philosopher's understanding particular mathematical results or the intuition behind them, but rather understanding the general nature of abstract, mathematical concepts.

Q: Anyway, granting the value of a reduction of mathematics to a simple foundation, why should it be set theory? For example, since sequences are so important in computing, it's natural to ask whether they could replace sets in the foundations of mathematics.

A: We don't know of any attempts in this direction. Transfinite sequences are a messy business. Nor do we know of attempts to use multisets, which are also computationally useful, for foundational purposes.

2.1 Non-ZF sets

Q: Concerning the intuitive idea of sets, is ZFC still the only game in town?

A: It's the biggest game, but there are others. For example, there are theories of sets and proper classes which extend ZFC. The most prominent ones are the von Neumann-Bernays-Gödel theory (NBG) and the Morse-Kelley theory (MK). In both cases the idea is to continue the cumulative hierarchy for one more step. The collections created at that last step are called proper classes.

Q: Wait a minute! The cumulative hierarchy is supposed to continue forever. How can there be another step? And if there is one more step, why not two or many?

A: We admit that this extra step doesn't quite make sense philosophically, but it is convenient technically. Consider some property of sets, for example the property of having exactly three members. It is convenient to refer to the multitude of the sets with this property as a single object. If this object isn't a set then it is a proper class.

There is also a less known but rather elegant extension of ZFC due to Ackermann [2]. It uses a distinction between sets and classes, but not the same distinction as in NBG or MK. For Ackermann, what makes a class a set is not that it is small but rather that it is defined without reference to the totality of all sets. It turns out [37, 46] that, despite the difference in points of view, Ackermann's set theory plus an axiom of foundation is equivalent to ZF in the sense that they prove the same theorems about sets. Lévy [37] showed how to interpret Ackermann's axioms by taking an initial segment of the cumulative hierarchy as the domain of sets and a much longer initial segment as the domain of classes.

Q: Are there set theories that contradict ZFC?

A: Yes. One is Quine's "New Foundations" (NF), named after the article [45] in which it was proposed. Another is Aczel's set theory with the anti-foundation axiom [3, 6].

Quine's NF is axiomatically very simple. It has the axiom of extensionality (just as in ZF) and an axiom schema of comprehension, asserting the existence of $\{x : \varphi(x)\}$ whenever $\varphi(x)$ is a stratified formula. "Stratified" means that one can attach integer "types" to all the variables so that, if $v \in w$ occurs in $\varphi(x)$, then $\text{type}(v) + 1 = \text{type}(w)$, and if $v = w$ occurs then $\text{type}(v) = \text{type}(w)$.

Q: This looks just like simple type theory.

A: Yes, but the types aren't part of the formula; stratification means only that there exist appropriate types. The point is that this restriction of comprehension seems sufficient to avoid the paradoxes.

Q: I see that it avoids Russell's paradox, since $\neg(x \in x)$ isn't stratified, but how do you know that it avoids all paradoxes?

A: We only said it seems to avoid paradoxes. Nobody has yet deduced a contradiction in NF, but nobody has a consistency proof (relative to, say, ZFC or even

ZFC with large cardinals). But Jensen [28] has shown that NF becomes consistent if one weakens the extensionality axiom to allow atoms. Rosser [47] has shown how to develop many basic mathematical concepts and results in NF. For lots of information about NF and (especially) the variant NFU with atoms, see Randall Holmes's web site [26].

Q: How does NF contradict the idea of the cumulative hierarchy?

A: The formula $x = x$ is stratified, so it is an axiom of NF that there is a universal set, the set of all sets. No such thing can exist in the cumulative hierarchy, which is never completed.

Q: And what about anti-foundation?

A: This theory is similar to ZFC, but it allows sets that violate the axiom of foundation. For example, you can have a set x such that $x \in x$; you can even have $x = \{x\}$.

Q: And you could have $x \in y \in x$ and even $x = \{y\} \wedge y = \{x\}$, right?

A: Yes, but the anti-foundation axiom imposes tight controls on these things. There is only one x such that $x = \{x\}$. Using that x as the value of both x and y you get $x = \{y\} \wedge y = \{x\}$, and this pair of equations has no other solutions. The axiom says, very roughly, that if you propose some binary relation to serve (up to isomorphism) as the membership relation in a transitive set, then, as long as it's consistent with the axiom of extensionality, it will be realized exactly once. It turns out that this axiomatic system and ZFC, though they prove quite different things, are mutually interpretable. That is, one can define, within either of the two theories, strange notions of "set" and "membership" that satisfy the axioms of the other theory.

2.2 Categories

Q: What about possible replacements for sets as the fundamental concept for mathematics? For example, I've heard people say that category theory could replace set theory as a foundation for mathematics. But I don't understand them. A category consists of a set (or class) of objects, plus morphisms and additional structure. So category theory presupposes the notion of set. How can it serve as a foundation by itself?

A: The idea that the objects (and morphisms) of a category must be viewed as forming a set seems to be an artifact of the standard, set-theoretic way of presenting general structures, namely as sets with additional structure. One can write down the axioms of category theory as first-order sentences and then do proofs from these axioms without ever mentioning sets (or classes).

Q: Sure, but unless you're a pure formalist, you have to wonder what these first-order sentences mean. How can you explain their semantics without invoking the traditional notion of structures for first-order logic, a notion that begins with "a non-empty *set* called the universe of discourse (or base set) . . . "?

A: This seems like another artifact of the set-theoretic mind-set, insisting that the semantics of first-order sentences must be expressed in terms of sets. People understood first-order sentences long before Tarski introduced the set-theoretic definition of semantics. Think of that set-theoretic definition as representing, within set theory, a pre-existing concept of meaning, just as Dedekind cuts or Cauchy sequences represent in set theory a pre-existing concept of real number.

Q: Hmmm. I'll have to think about that. It still seems hard to imagine the meaning of a first-order sentence without a set for the variables to range over. But let's suppose, at least for the sake of the discussion, that the axioms of category theory make sense without presupposing sets. Those axioms seem much too weak to serve as a foundation; after all, they have a model with one object and one morphism.

A: That's right. For foundational purposes, one needs axioms that describe not just an arbitrary category but a category with additional structure, so that its objects can represent the entities that mathematicians study.

Q: That sounds reasonable but vague. What sort of axioms are we talking about here?

A: There have been two approaches. One is to axiomatize the category of categories and the other is to axiomatize a version of the category of sets.

Q: The first of these sounds more like a genuinely category-theoretic foundation; the second mixes categories and sets.

A: Yes, but the first has had relatively little success.

Q: Why? What's its history?

A: The idea was introduced by Lawvere in [34]. He proposed axioms, in the first-order language of categories, to describe the category of categories, and to provide tools adequate for the formalization of mathematics. But three problems arose. First, as pointed out by Isbell in his review [27], the axioms didn't quite accomplish what was claimed for them. That could presumably be fixed by modifying the axioms. But there was a second problem: Although some of the axioms were quite nice and natural, others were rather unwieldy, and there were a lot of them. As a result, it looked as if the axioms had just been rigged to simulate what can be done in set theory. That's related to the third problem: The representation of some mathematical concepts in terms of categories was done by, in

effect, representing them in terms of sets and then treating sets as discrete categories (categories in which the only morphisms are the identity morphisms, so the category is essentially just its set of objects). This third point should not be overemphasized; some concepts were given very nice category-theoretic definitions. For example, the natural number system is the so-called coequalizer of a pair of morphisms between explicitly described finite categories. But the use of discrete categories for some purposes made the whole project look weak.

Q: So what about the other approach, axiomatizing the category of sets?

A: That approach, also pioneered by Lawvere [33], had considerably more success, for several reasons. First, many of the basic concepts and constructions of set theory (and even of logic, which underlies set theory) have elegant descriptions in the language of categories; specifically, they can be described as so-called adjoint functors. In the category of sets, adjoint functors provide definitions of disjoint union, cartesian product, power set, function set (i.e., the set of all functions from X to Y), and the set of natural numbers, as well as the logical connectives and quantifiers.

Q: That covers quite a lot. What other advantages does the category of sets have — or provide?

A: There is a technical advantage, namely that the axioms admit a natural weakening that describes far more categories than just the category of sets. These categories, called topoi or toposes, resemble the category of sets in many ways (including the availability of all of the constructions listed above, except that the existence of the set of natural numbers is usually not included in the definition of topos) but also differ in interesting ways (for example, the connectives and quantifiers may obey intuitionistic rather than classical logic), and there are many topoi that look quite different from the category of sets (not only non-standard models of set theory but also categories of sheaves, categories of sets with a group acting on them, and many others). As a result, set-theoretic arguments can often be applied in topoi in order to obtain results about, for example, sheaves. These ideas were introduced by Lawvere and Tierney in [36]; see [29] and [39] for further information.

Q: I don't know what sheaves are. In any case, I care mostly about foundations, so this technical advantage doesn't do much for me. What more can the category of sets do for the foundations of mathematics?

A: One can argue that the notion of abstract set described in this category-theoretic approach is closer to ordinary mathematical practice than the cumulative hierarchy described by the Zermelo-Fraenkel axioms.

Q: What is this notion of abstract set? The ZF sets look pretty abstract to me.

A: The phrase "abstract set" refers (in this context) to abstracting from any internal structure that the elements of a set may have. A typical set in the cumulative hierarchy has, as elements, other sets, and there may well be membership relations (or more complicated set-theoretic relations) between these elements. Abstract set theory gets rid of all this. As described in [35], an abstract set "is supposed to have elements, each of which has no structure, and is itself to have no internal structure, except that the elements can be distinguished as equal or unequal, and to have no external structure except for the number of elements."

Q: How is this closer to ordinary mathematical practice than the cumulative hierarchy view of sets?

A: One way to describe the difference is that the abstract view gets rid of unnecessary structure. For example, in any of the usual set-theoretic representations of the real numbers, the basic facts about $\mathbb{R}$ depend on information about, say, members of members of real numbers — information that mathematicians would never refer to except when giving a lecture on the set-theoretic representation of the real numbers. The abstract view discards this sort of information. Of course, some structural information is needed — unlike abstract sets, the real number system has internal structure. But the relevant structure is postulated directly, say by the axioms for a complete ordered field, not obtained indirectly as a by-product of irrelevant structure.

Q: So if an abstract-set theorist wanted to talk about a set from the cumulative hierarchy, with all the structure imposed by that hierarchy, he would include that structure explicitly, rather than relying on the hierarchy to provide it.

A: Exactly. If x is a set in the cumulative hierarchy, then one can form its transitive closure t, the smallest set containing x and containing all members of its members. Then t with the membership relation $\in$ (restricted to t) is an abstract representation of t. It no longer matters what the elements of t were, because any isomorphic copy of the structure $(t, \in)$ contains the same information and lets you recover x.

Q: Well if this category-theoretic view of abstract sets is so wonderful, why isn't everybody using it?

A: There are (at least) four answers to your question. One is a matter of history. The cumulative hierarchy view of sets has been around explicitly at least since 1930 [52], and Zermelo's part of ZFC (all but the replacement and foundation axioms) goes back to 1908 [51]. ZFC has had time to demonstrate its sufficiency as a basis for ordinary mathematics. People have become accustomed to it as the foundation of mathematics, and that includes people who don't actually know what the ZFC axioms are. There is, however, a chance that the abstract view of sets will gain ground if students learn basic mathematics from books like [35].

A second reason is the simplicity of the primitive notion of set theory, the membership predicate. Perhaps, we should say "apparent simplicity," in view of the complexity of what can be coded in the cumulative hierarchy. But still, the idea of starting with just $\in$ and defining everything else is philosophically appealing. Another way to say this is that, in developing mathematics, one certainly needs the concepts of "set" and "membership"; if everything else can be developed from just an iteration of these (admittedly a transfinite iteration), why not take advantage of it?

Third, there is a technical reason. Although topos theory provides an elegant view of the set-theoretic constructions commonly used in mathematics, serious uses of the replacement axiom don't look so nice in category-theoretic terms. (By serious uses of replacement, we mean something like the proof of Borel determinacy [40], which provably [22] needs uncountably many iterations of the power set operation.) But such serious uses are still quite rare.

Q: OK, what's the fourth answer to why people aren't using the category-theoretic view of abstract sets?

A: The fourth answer is that they *are* using this point of view but just don't realize it. Mathematicians talk about ZFC as the foundation of what they do, but in fact they rarely make explicit use of the cumulative hierarchy. That hierarchy enters into their work only as an invisible support for the structures they really use — like the complete ordered field $\mathbb{R}$. When you look at what these people actually say and write, it is entirely consistent with the category-theoretic viewpoint of abstract sets equipped with just the actually needed structure.

2.3 Functions

Q: The discussion of categories, with their emphasis on morphisms alongside objects, reminds me of a way in which functions could be considered more basic than sets.

A: More basic? "As basic" seems reasonable, if one doesn't insist on representing functions set-theoretically (using ordered pairs), but in what sense do you mean "more basic"?

Q: This came up when I was a teaching assistant for a discrete mathematics class. Sets were one of the topics, and several students had trouble grasping the idea that, for example, a thing a and the set $\{a\}$ are different, or that the empty set is one thing, not nothing. They thought of a set as a physical collection, obtained by bringing the elements together, not as a separate, abstract entity.

A: Undergraduate students aren't the only people who had such difficulties; see [30] for some relevant history. But what does this have to do with functions?

Q: Well, I found that I could clarify the problem for these students by telling them to think of a set S as a black box, where you can put in any potential element x and it will tell you "yes" if $x \in S$ and "no" otherwise. So I was explaining the notion of set in terms of functions, essentially identifying a set with its characteristic function. The black-box idea, i.e., functions, seemed to be something the students could understand directly, whereas sets were best understood via functions.

A: It seems that functions are obviously abstract, so the students aren't tempted to identify them with some concrete entity, whereas they are tempted to do that with sets.

Q: That may well explain what happened with my students.

If one takes seriously the idea of functions being more basic than sets, then it seems natural to develop a theory of functions as a foundation for mathematics. Has that been tried?

A: Yes, although sometimes the distinction between using sets and using functions as the basic notion is rather blurred.

Q: Blurred how?

A: Well, the set theory now known as von Neumann-Bernays-Gödel (NBG) was first introduced by von Neumann [42, 43] in terms of functions. But he minimizes the significance of using functions rather than sets. Not only do the titles of both papers say "Mengenlehre" (i.e., "set theory") with no mention of functions, but von Neumann explicitly writes that the concepts of set and function are each easily reducible to the other and that he chose functions as primitive solely for technical simplicity.[1] And when Bernays [7] recast the theory in terms of sets and classes (the form in which NBG is known today), he described his work as "a modification of a system due to von Neumann," the purpose of the modification being "to remain nearer to the structure of the original Zermelo system and to utilize at the same time some of the set-theoretic concepts of the Schröder logic and of *Principia Mathematica*." Bernays doesn't mention that the primitive concept has been changed from function to set (and class). The tone of Bernays's introduction gives the impression that the change is not regarded as a significant change in content but rather as a matter of connecting with earlier work (Zermelo, Schröder, Russell, and Whitehead) and of technical convenience (Bernays mentions a "considerable simplification" vis à vis von Neumann's system).

Q: Von Neumann claimed that functions were technically simpler than sets, and Bernays claimed the opposite?

[1] Wir haben statt dem Begriffe der Menge hier den Begriff der Funktion zum Grundbegriffe gemacht: die beiden Begriffe sind ja leicht aufeinander zurückzuführen. Die technische Durchführung gestaltet sich jedoch beim Zugrundelegen des Funktionsbegriffes wesentlich einfacher, allein aus diesem Grunde haben wir uns für denselben entschieden. [43, page 676]

A: Yes. Of course, the set-based system that von Neumann had in mind for his comparison may have been more complex than Bernays's system. Presumably part of Bernays's work was to make the set-based approach simpler.

By the way, Gödel [24] modified Bernays's formulation slightly; in particular, he used a single membership relation, whereas Bernays had distinguished between membership in sets and membership in classes. Gödel describes his system as "essentially due to P. Bernays and ... equivalent to von Neumann's system ...." In the announcement [23], Gödel stated his consistency result in terms of von Neumann's system.

Q: So it seems we can think of von Neumann's function-based axiom system as being in some sense the same as the set-based system now known as NBG. But are there function-based foundations that aren't just variants of more familiar set-based systems?

A: The lambda calculus [4, 5] and its variations fit that description. The idea here is that one works in a world of functions, with application of a function to an argument as a primitive concept. There is also the primitive notion of lambda-abstraction; given a description of a function using a free variable v, say some meaningful expression A involving v, one can produce a term $\lambda v\, A$ (which most mathematicians would write as $v \mapsto A$), denoting the function whose value at any v is given by A. In the untyped lambda calculus, one takes the functions to be defined at all arguments. That way, one doesn't need to specify sets as the domains of the functions; every function has universal domain. The typed lambda calculus is less antagonistic to sets; its functions have certain types as their domains and codomains.

Q: I've seen the lambda calculus mentioned in two places in computer science. First, Church's original statement of his famous thesis identified the intuitive concept of computability with definability in the lambda calculus. Second, lambda calculus plays a major role in the domain-theoretic approach to denotational semantics. But how does it relate to foundations of mathematics?

A: Church [12] originally intended the lambda calculus as an essential part (the other part being pure logic) of a foundational system for mathematics. The other pioneers of lambda calculus, albeit in the equivalent formulation using combinators, were Schönfinkel [48] and Curry [15, 16, 17], and they also had foundational objectives. Unfortunately, Church's system turned out to be inconsistent [31], and the system proposed by Curry was not strong enough to serve as a general foundation for mathematics. (Schönfinkel's system was also weak, being intended just as a formulation of first-order logic.)

Q: So this approach to foundations was a dead end.

A: Not really; the task is neither dead nor ended. The original plans didn't succeed, but there has been much subsequent work, which has succeeded to a considerable extent, and which may have more successes ahead of it. Church himself developed not only the pure lambda calculus [14] (essentially the lambda part of his earlier inconsistent system, but without the logical apparatus that led to the inconsistency) but also a typed lambda calculus [13] that is essentially equivalent to the simple theory of types but expressed in terms of functions and lambda abstraction instead of sets and membership. The typed lambda calculus also provides a good way to express the internal logic of topoi (and certain other categories) [32]. It forms the underlying framework of the system developed by Martin-Löf [41] as a foundation for intuitionistic mathematics. There is also a considerable body of work by Feferman (for example [18, 19, 20]) on foundational systems that incorporate versions of the lambda calculus and that have both constructive and classical aspects.

Q: So if you meet mathematicians from a far away planet, would you expect their mathematics to be set-based?

A: Not necessarily but we wouldn't be surprised if their mathematics is set-based. We would certainly expect them to have a set theory, but it might be quite different from the ones we know, and it might not be their foundation of mathematics.

Acknowledgment

We thank Akihiro Kanamori and Jan Van den Bussche for promptly reading a draft of this paper and making helpful remarks.

References

[1] Serge Abiteboul, Richard Hull, and Victor Vianu, *Foundations of Databases*, Addison-Wesley (1995).

[2] Wilhelm Ackermann, "Zur Axiomatik der Mengenlehre," *Math. Ann.* 131 (1956) 336–345.

[3] Peter Aczel, *Non-Well-Founded Sets*, CSLI Lecture Notes 14, Center for the Study of Language and Information, Stanford Univ. (1988).

[4] Henk Barendregt, *The Lambda Calculus. Its Syntax and Semantics*, Studies in Logic and the Foundations of Mathematics 103, North-Holland (1984).

[5] Henk Barendregt, "The impact of the lambda calculus in logic and computer science," *Bull. Symbolic Logic* 3 (1997) 181–215,

[6] Jon Barwise and Lawrence Moss, *Vicious Circles. On the mathematics of non-wellfounded phenomena*, CSLI Lecture Notes 60, Center for the Study of Language and Information, Stanford Univ. (1996).

[7] Paul Bernays, "A system of axiomatic set theory — Part I," *J. Symbolic Logic* 2 (1937) 65–77.

[8] Andreas Blass, Yuri Gurevich, and Saharon Shelah, "Choiceless Polynomial Time", *Annals of Pure and Applied Logic* 100 (1999) 141–187.

[9] Andreas Blass, Yuri Gurevich, and Jan Van den Bussche, "Abstract State Machines and Computationally Complete Query Languages", *Information and Computation* 174 (2002) 20-36.

[10] Nicolas Bourbaki, *Elements of mathematics: Theory of sets*, Translation from French, Addison-Wesley (1968).

[11] Ashok Chandra and David Harel, "Structure and Complexity of Relational Queries", *J. Comput. and System Sciences* 25 (1982) 99–128.

[12] Alonzo Church, "A set of postulates for the foundation of logic," *Ann. Math. (2)* 33 (1932) 346–366 and 34 (1933) 839–864.

[13] Alonzo Church, "A formulation of the simple theory of types," *J. Symbolic Logic* 5 (1940) 56–68.

[14] Alonzo Church, *The Calculi of Lambda-Conversion*, Annals of Mathematics Studies 6, Princeton Univ. Press (1941).

[15] Haskell Curry, "Grundlagen der kombinatorischen Logik," *Amer. J. Math.* 52 (1930) 509–536 and 789–834.

[16] Haskell Curry, "The combinatory foundations of mathematical logic," *J. Symbolic Logic* 7 (1942) 49–64.

[17] Haskell Curry and Robert Feys, *Combinatory Logic. Vol. I*, North-Holland (1958).

[18] Solomon Feferman, "A language and axioms for explicit mathematics," in *Algebra and Logic*, ed. J. Crossley, Lecture Notes in Mathematics 450, Springer-Verlag (1975) 87–139.

[19] Solomon Feferman, "Constructive theories of functions and classes," in *Logic Colloquium '78*, ed. M. Boffa, D. van Dalen, and K. McAloon, Studies in Logic and the Foundations of Mathematics 97, North-Holland (1980) 159–224.

[20] Solomon Feferman, "Toward useful type-free theories, I," *J. Symbolic Logic* 49 (1984) 75–111.

[21] Abraham Fraenkel, Yehoshua Bar-Hillel, and Azriel Lévy, *Foundations of Set Theory*, Studies in Logic and the Foundations of Mathematics 67, North-Holland (1973).

[22] Harvey Friedman, "Higher set theory and mathematical practice," *Ann. Math. Logic* 2 (1970/71) 325–357.

[23] Kurt Gödel, "The consistency of the axiom of choice and of the generalized continuum hypothesis," *Proc. Nat. Acad. Sci. U.S.A.* 24 (1938) 556–557.

[24] Kurt Gödel, *The Consistency of the Axiom of Choice and the Generalized Continuum Hypothesis with the Axioms of Set Theory*, Annals of Mathematics Studies 3, Princeton Univ. Press (1940).

[25] Yuri Gurevich, "Logic and the Challenge of Computer Science", in *Current Trends in Theoretical Computer Science*, ed. E. Börger, Computer Science Press (1988) 1–57.

[26] Randall Holmes, *New Foundations Home Page*, `http://math.boisestate.edu/~holmes/holmes/nf.html`.

[27] John Isbell, Review of [34], *Mathematical Reviews* 34 (1967) #7332.

[28] Ronald Jensen, "On the consistency of a slight(?) modification of Quine's NF," *Synthèse* 19 (1969) 250–263.

[29] Peter T. Johnstone, *Topos Theory*, London Math. Soc. Monographs 10, Academic Press (1977).

[30] Akihiro Kanamori, "The empty set, the singleton, and the ordered pair," *Bull. Symbolic Logic* 9 (2003) 273–298.

[31] Stephen Kleene and J. Barkley Rosser, "The inconsistency of certain formal logics," *Ann. Math. (2)* 36 (1935) 630–636.

[32] Joachim Lambek and Philip Scott, *Introduction to Higher Order Categorical Logic*, Cambridge Studies in Advanced Mathematics 7, Cambridge Univ. Press (1986).

[33] F. William Lawvere, "An elementary theory of the category of sets," *Proc. Nat. Acad. Sci. U.S.A.* 52 (1964) 1506–1511.

[34] F. William Lawvere, "The category of categories as a foundation for mathematics," in *Proc. Conf. Categorical Algebra (La Jolla, CA, 1965)*, Springer-Verlag (1966) 1–20.

[35] F. William Lawvere and Robert Rosebrugh, *Sets for Mathematicians*, Cambridge Univ. Press (2003).

[36] F. William Lawvere and Myles Tierney, "Quantifiers and sheaves," in *Actes du Congrès International des Mathématiciens (Nice, 1970), Tome 1*, Gauthier-Villars (1971) 329–334.

[37] Azriel Lévy, "On Ackermann's set theory," *J. Symbolic Logic* 24 (1959) 154–166.

[38] Azriel Lévy, *Basic Set Theory*, Perspectives in Mathematical Logic, Springer-Verlag (1979).

[39] Saunders Mac Lane and Ieke Moerdijk, *Sheaves in geometry and logic. A first introduction to topos theory*, Universitext, Springer-Verlag (1994).

[40] Donald A. Martin, "Borel determinacy," *Ann. Math. (2)* 102 (1975) 363–371.

[41] Per Martin-Löf, "An intuitionistic theory of types: predicative part," in *Proceedings of the Logic Colloquium (Bristol, July, 1973)*, ed. H. E. Rose and J. C. Shepherdson, Studies in Logic and the Foundations of Mathematics 80, North-Holland (1975) 73–118.

[42] Johann von Neumann, "Eine Axiomatisierung der Mengenlehre," *J. Reine Angew. Math.* 154 (1925) 219–240. English translation in *From Frege To Gödel. A Source Book in Mathematical Logic, 1879–1931*, ed. J. van Heijenoort, Harvard University Press (1967) 393–413.

[43] Johann von Neumann, "Die Axiomatisierung der Mengenlehre," *Math. Z.* 27 (1928) 669–752

[44] Richard Platek, "Eliminating the continuum hypothesis," *J. Symbolic Logic* 34 (1969) 219–225.

[45] Willard Van Orman Quine, "New foundations for mathematical logic," *Amer. Math. Monthly*, 44 (1937) 70–80

[46] William Reinhardt, "Ackermann's set theory equals ZF," *Ann. Math. Logic* 2 (1970) 189–249.

[47] J. Barkley Rosser, *Logic for Mathematicians*, McGraw-Hill (1953).

[48] Moses Schönfinkel, "Über die Bausteine der mathematischen Logik," *Math. Ann.* 92 (1924) 305–316.

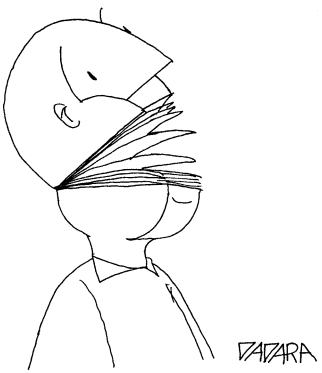
[49] Jacob T. Schwartz, Robert B. K. Dewar, Ed Dubinsky, and Edmond Schonberg, *Programming with Sets: An Introduction to SETL*, Springer-Verlag (1986).

[50] Z users website `http://vl.zuser.org`.

[51] Ernst Zermelo, "Untersuchungen über die Grundlagen der Mengenlehre I," *Mathematische Annalen* 65 (1908) 261–281.

[52] Ernst Zermelo, "Über Grenzzahlen und Mengenbereiche, Neue Untersuchungen über die Grundlagen der Mengenlehre," *Fundamenta Mathematicae* 16 (1930) 29–47.

TECHNICAL CONTRIBUTIONS



Some Preliminary Results on Three Combinatorial Board Games

Samee Ullah Khan* Ishfaq Ahmad†

Abstract

This paper analyzes a certain class of combinatorial board games that includes Ayo, Tchoukaillon and Modular *N*–queen. We refine the existing results by providing simple and intuitive proofs using abstract combinatorics, and also reveal some interesting game positions.

1 Introduction

Chess and its variants, are pure strategic games without any random moves. It should, at least in principle, be possible to decide whether there is a winning strategy for the first or the second player, or whether the game always ends with a draw (assuming a perfect play). Such games are called combinatorial games, and combinatorial game theory (CGT) is a branch of mathematics devoted to their analysis [2]. A rich theory on how to evaluate game positions has been developed in recent years, and it has been successfully applied to analyze certain endgame positions of Chess [6] and Go [1], for example. Unfortunately, CGT cannot directly be applied to Chess, Shogi and Xiangqi, due to the fact that draws do not qualify as a game in CGT [2, 4]. Moreover, CGT also fails to help analyze various other board games such as *Ayo*, *Tchoukaillon* and *Modular N–queen* which are classified as pure combinatorial games [7, 9, 12]. In this paper we analyze these three board games with the help of abstract combinatorics and report some preliminary results.

2 Ayo

There are various kinds of Mancala games that date back to the early years of the great Egyptian Empire [10]. Mancala games mostly played in Nigeria are a group

*Department of Computer Science and Engineering, University of Texas at Arlington, sakhan@cse.uta.edu. †Department of Computer Science and Engineering, University of Texas at Arlington, iahmad@cse.uta.edu.

of games, with certain common characteristics. They all involve cup–shaded depressions called pits that are filled with stones or seeds. Players take turns and maneuver the stones, by various rules, which govern them. Ayo is one such two–player game played mostly in the western part of Nigeria. With predefined rules, Ayo players follow their respective strategies and do not depend on chance moves (dice).

Ayoyayo (Ayo) is played over a wooden board 20 inches long, 8 inches wide, and 2 inches thick. This board accommodates two rows of six pits each 3 inches in diameter. The pits are filled with either stones or dried palm nuts [3].

Ayo is played with 48 stones with 4 stones placed in each of the 12 pits. Two players alternatively move the stones, each controlling 6 pits. Their objective is to capture their opponent's stones (as many as possible). A move consists of a player choosing a non–empty pit on his side of the board and removing all of the stones contained in that pit. The stones are redistributed (*sown*) one stone per pit. The pits are sown in counter–clockwise direction from the pit that has been just emptied. A pit containing 12 or more stones is called an *Odu*. If the chosen pit is an *Odu*, the same redistribution continues, but the emptied pit is skipped on each circuit of the board [10]. A capture is made, when the last pit sown is on the opponent's side, and contains after the addition of the sowing stone either two or three stones. Thus, the stones in the pit are captured and removed from the game. Also are captured the immediately preceding pits which meet the same conditions. One important feature of this game is that each player has to make a move such that his opponent has a legal move to play. If this does not happen, then the opponent is rewarded with all the remaining stones on the board. If during the game, it is found that there are not enough stones to make a capture, but both the players can always proceed with a legal move, the game is stopped and the players are awarded stones that reside on their respective side of the board. The initial game is rapid and much more interesting, where both the players capture stones in quick succession. To determine the optimal strategy during the initial play is hard, and thus has not yet been studied. It involves planning at least 2–3 moves in advance, and remembering the number of stones in every pit [3, 10].

To the best of the authors' knowledge, the only work reported on Ayo can be found in [3]. They generalized Ayo such that: 1) Each player got n pits, 2) the *Odu* rule was overridden, and 3) the Ayo board was numbered in clockwise $-n+2, -n+1, \cdots, -1, 0, 1, \cdots, n, n+1$. Their analysis was only confined to the Ayo endgames. They defined a determinable position as an arrangement of stones where it is possible for a player to move such that:

1. A player captures at every turn.

2. No move is allowed from *Odu*.

3. After a player has moved, his opponent has only one stone on his side of the board.

4. Every stone is captured except the one which is award to his opponent.

Based on these assumptions, Broline and Loeb, also showed a small endgame with nine stones. Thus they provided us with the following lemma.

Lemma 1. *If a player has to move in a determined Ayo position, his stone has to be in pit* 1, *else in pit* 0 *if his opponent has to move.*

The proof presented in [3] is very complicated and tedious. Here we present a simple and intuitive proof (the pit positions are illustrated in Figure 1).

Proof: From the predefined set of four legal moves, if a player's opponent has to make the move, he must capture (step 1) and leave only one stone (step 3). With simple combinatorics it can be shown that the stone has to be in pit 0 (step 4). Thus before the player's move, the stone has to be in pit 1.

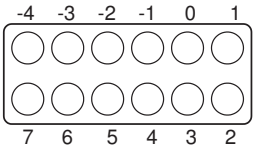


Figure 1: Ayo board labelled with pit numbers.

3 Tchoukaillon

Tchoukaillon is a Russian game, played with pits dug into sand and the pits filled with stones. The game is the modern variation of *Tchouka* which was mostly played in central Europe. In [5] the authors developed Tchoukaillon as the modern variant of Tchouka. This game also falls in the category of Mancala games [9]. The modern version of Tchoukaillon involves playing over a strip of wood (also possible is the circular arrangement of the pits). These pits contain a certain number of stones, with one empty pit called *Rouma*, *Cala* or *Roumba*.

There is no limit as to how many stones can to be used, and neither is a limit on the number of pits [12]. The objective of the game is to put the stones in *Roumba*. The maneuvering of the stones is called *sow*. Thus, like a solitaire game stones are sown into the empty pit. The sowing takes place as a constant one stone per

pit at a time in the direction of *Roumba*, but it can also be in the opposite direction of *Roumba*. Therefore, there can be only three possibilities during the game:

1. If the last stone drops into *Roumba*, the player has a choice to start sowing another pit of his choice.

2. If the last stone drops in an occupied non-*Roumba* pit, this pit is to be sown immediately.

3. If the last pit drops in an empty non-*Roumba* pit, the game is over and the player who does this losses.

The objective in a two–player Tchoukaillon game is to play the last stone in an empty pit so that the next player takes the turn. While doing so he has to sow as many stones in *Roumba* as possible. The winner of the game is the player with the largest number of stones in *Roumba*. The Tchoukaillon board is shown in Figure 2.

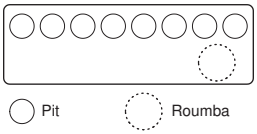


Figure 2: Tchoukaillon board.

To the best of the authors' knowledge, the work reported in [9] and [3] are the only ones that address Tchoukaillon as a combinatorial game. In [3] the authors investigated some winning positions, but disregarded a number of other possibilities of winning and confined themselves to just the objective of sowing stones in *Roumba* only. Their strategy is based on the results based on the work of [5]. They stated the winning move as:

If a win is possible from a given Tchoukaillon position, the unique winning move must be to harvest the smallest harvestable pit.

Although the authors in [3] identified a winning strategy, yet they did not provide any proof of the strategy. Here we state a simple proof.

Lemma 2. *The unique winning move in Tchoukaillon is to harvest the smallest harvestable pit.*
Proof: Imagine if there are only two pits on the board, with one pit having less

number of stones than the other. If a stone is taken from the pit having more stones, it will increase the number of stones in the smaller pit. Thus the pit at some moment will overflow if a play is continued in this fashion, and an indefinite play will continue.

From Lemma 1, one can always argue that there should be a way to backtrack all possible positions till the initial setup. Based on this argument one can always make the following claim.

Conjecture 1. *For all $s = 0$ (initial setup of the game), there is exactly one winning position involving a total of s stones.*

4 Modular N–queen

Consider an *nn* ordinary chessboard. It is always possible to find *n* queens positioned in such a way, that no two attack each other. This is although only true, when $n = 4$. There are a number of ways to pose the modular *n*–queen problem, e.g., How many such placements can be found, when there are no such two queens who share a row, column or a diagonal? The original problem was only for 8 queens on a regular chessboard. For the original 8 queens' problem, 92 solutions to this date have been identified. Of these 92, there are 12 distinct patterns. Thus all of the 92 solutions can be transformed into the 12 unique patterns, using reflection and rotation. These 12 patterns are show in Table 1. For instance, if one is to constructing solution number 1, then the queen for chessboard row 1 should be placed in column 1, the queen for row 2 should be placed in column 5, and so on.

A modular chess board is a one where the diagonals run on the other side of the board. Thus a queen can still be under attack, even if it is not directly under attack from another queen on the diagonal. This fact is illustrated in Figure 3. The basic question asked for a modular chessboard is: What is the maximum number of queens that can be accommodated on a modular chessboard, such that no two queens attack each other? In other words if an *nn* chessboard is transformed into a torus by identifying the opposite side, we want to place *n* queens in such a way that none attack each other. This number is denoted as $M(n)$. The basic modular chessboard *n*–queen problem is accredited to Pölya [11].

There has been a lot of work done on the modular *n*–queen problem. The first major result was reported in [11]. There the author proved that if and only if $gcd(n, 6) = 1$ then there are *n* queens on the modular board. The same result was proved by many others, e.g. see [7]. Later Klöve improved the result and gave the following theorem [8].

Theorem 1. *The modular chessboard has solutions of the following form:*

Sol.	Row 1	Row 2	Row 3	Row 4	Row 5	Row 6	Row 7	Row 8
1	1	5	8	3	3	7	2	4
2	1	6	8	7	7	4	2	5
3	2	4	6	3	3	1	7	5
4	2	5	7	3	3	8	6	4
5	2	5	7	1	1	8	6	3
6	2	6	1	4	4	8	3	5
7	2	6	8	1	1	4	7	5
8	2	7	3	8	8	5	1	4
9	2	7	5	8	1	4	6	3
10	3	5	2	8	1	7	4	6
11	3	5	8	4	1	7	2	6
12	3	6	2	5	8	1	7	4

Table 1: 12 unique patterns for the 8–queen problem.

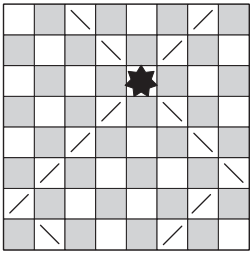


Figure 3: Modular chessboard.

1. $M(n) = n$ *if* $gcd(n, 6) = 1$
2. $M(n) = n - 2$ *if* $gcd(n, 6) = 3$
3. $n - 3 = M(n) = n - 1$ *if* $gcd(n, 6) = 2$
4. $n - 5 = M(n) = n - 1$ *if* $gcd(n, 6) = 6$

Proof: [8]

Later, Heden further improved Klöve's result by providing a simpler proof of the original work reported in [11]. The following theorem is deduced from the work reported in [7].

Theorem 2. *The modular chessboard has solutions of the following form:*

1. $M(n) = n$ *if* $gcd(n, 6) = 1$
2. $M(n) = n - 1$ *if* $gcd(n, 12) = 2$
3. $M(n) = n - 2$ *if* $gcd(n, 6) = 3$ *or* 4
4. $n - 4 = M(n) = n - 2$ *if* $gcd(n, 12) = 6$
5. $n - 5 = M(n) = n - 2$ *if* $gcd(n, 12) = 12$

Proof: [7]

Heden used the concept of chains of queens and colored queens to prove his results. A chain is closed, if $Q_1 = Q_k$. Similarly, chaining can be defined for rows and columns. Heden defined four colors for queens as *A*, *B*, *C* and *D*. A queen colored in color *A* is called *A*–queen. A queen colored in either *A* or *D* is called *AD*–queen, and so on. This helped in defining the colorings for the diagonal and bi–diagonal, which are the bases of his proof. This approach is so effective, that he was able to give a partial solution without the aid of computer towards the modular 12–double queen problem, with 22 queens. A modular double queen problem is an extension of the normal modular queen problem, where there can be at most two queens on a single row, column, or a diagonal.

Here, we generalize the chains of queens concepts and give the following necessary and sufficient conditions.

Conjecture 2. *A chain of queens on a diagonal is a set of queens* (Q_1, Q_2, Q_3, $\cdots$, Q_k) *such that the following two conditions hold:*

1. *No three queens are on the same diagonal or bi–diagonal.*
2. *Two queens marked consecutively are always in the same diagonal or bi–diagonal.*

5 Concluding Remarks and Some Open Problems

In this paper we provided the readers with some preliminary results on a class of combinatorial board games that includes Ayo, Tchoukaillon and Modular *N*–queen. These games require abstract combinatorial analysis and cannot be analyzed by pure combinatorial game theoretical methods. Thus, the combinatorial game theory lacks some fine tuning especially in case of combinatorial board games (loopy games). We now pose the following open problems:

Problem 1: Is it possible to identify a winning sequence of moves provided an arbitrary board position in Tchoukaillon?

We conjecture that this is possible, but it is to be noted that the game played with n number of pits might involve enormous amount of moves (Conjecture 1). It is thus, *also* possible that the problem might not be intractable to begin with.
Problem 2: Are there more than two necessary and sufficient conditions to form a chain of queens on a diagonal?

References

[1] E. R. Berlekamp and D. Wolfe. *Mathematical Go–Chilling Gets the Last Point*. A K Peters, Natick, MA, 1994.

[2] E. R. Berlekamp, J. H. Conway, and R. K. Guy. *Winning Ways for your Mathematical Plays*, volume I & II. Academic Press, London, 1982. 2nd edition of vol. 1 (of four volumes), 2001, A. K. Peters, Natick, MA.

[3] D. M. Brolin and D. E. Loeb. The combinatorics of mancala-type games. *UMAP*, 10(1), 1995.

[4] J. H. Conway. *On Numbers and Games*. Academic Press, London, 1976.

[5] A. Deledicq and A. Popova. Wari et solo: Le jeu de calculs africain. *Collection Les Distracts*, 3:318–359, 1977.

[6] N. D. Elkies. On numbers and endgames: combinatorial game theory in chess endgames. In R. J. Nowakowski, editor, *Games of No Chance*, Proc. MSRI Workshop on Combinatorial Games, July, 1994, Berkeley, CA, MSRI Publ., volume 29, pages 135–150. Cambridge University Press, Cambridge, 1996.

[7] O. Heden. On the modular n–queen problem. *Discrete Mathematics*, 102:155–161, 1992.

[8] T. Klöve. The modular n–queen problem II. *Discrete Mathematics*, 36:33–48, 1981.

[9] D. E. Loeb. Combinatorial properties of mancala. *Abstratcs of AMS*, 96:471, 1994.

[10] A. O. Odeleya. *Ayo: A popular Yoruba Game*. Oxford University Press, Ibadan, Nigeria, 1977.

[11] G. Polya. Uber die “doppelt–periodishen” lösungen des n–damen–problem. In W. Ahrens, editor, *Mathematische Unterhaltungen and Spiele*, volume 2. 1921.

[12] M. A. Saint-Laguë. Géométrie de situation et jeux. *Mémorial des Sciences Mathématiques*, Fascicule XLI, 1929.

PASSAGES OF PROOF

Cristian S. Calude[1], Elena Calude[2], Solomon Marcus[3]

[1]University of Auckland, New Zealand

cristian@cs.auckland.ac.nz

[2]Massey University at Albany, New Zealand

e.calude@massey.ac.nz

[3]Romanian Academy, Mathematics, Bucharest, Romania

Solomon.Marcus@imar.ro

1 To Prove or Not to Prove–That Is the Question!

Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune,
Or to take arms against a sea of troubles
And by opposing end them?
Hamlet 3/1, by W. Shakespeare

In this paper we propose a new perspective on the evolution and history of the idea of mathematical proof. Proofs will be studied at three levels: syntactical, semantical and pragmatical. Computer-assisted proofs will be give a special attention. Finally, in a highly speculative part, we will anticipate the evolution of proofs under the assumption that the quantum computer will materialize. We will argue that there is little 'intrinsic' difference between traditional and 'unconventional' types of proofs.

2 Mathematical Proofs: An Evolution in Eight Stages

Theory is to practice as rigour is to vigour. D. E. Knuth

Reason and *experiment* are two ways to acquire knowledge. For a long time mathematical proofs required only reason; this might be no longer true. We can distinguish eight periods in the evolution of the idea of mathematical proof. The first period was that of pre–Greek mathematics, for instance the Babylonian one, dominated by observation, intuition and experience.

The second period was started by Greeks such as Pythagoras and is characterized by the discovery of deductive mathematics, based on theorems. Pythagoras proved his theorem, but the respective statement was discovered much earlier. Deductive mathematics saw a culminating moment in Euclid's geometry. The importance of abstract reasoning to ancient Greeks can be illustrated by citing Aristophanes's comedy *The Birds* which includes a cameo appearance of Meton, the astronomer, who claims that he had squared the circle. Knuth [44] rhetorically asked: "Where else on earth would a playwright think of including such a scene?" Examples would have been difficult to produce in 1985, but today the situation has changed. Take for example, the movie *Pi* written and directed by Darren Aronofsky Starring Sean Gullette or Auburn's play *Proof* [2] originally produced by the Manhattan Theatre Club on 23rd May 2000.

In a more careful description, we observe that deductive mathematics starts with Thales and Pythagoras, while the axiomatic approach begins with Eudoxus and especially with Aristotle, who shows that a demonstrative science should be based on some non–provable principles, some common to all sciences, others specific to some of them. Aristotle also used the expression "common notions" for axioms (one of them being the famous principle of non–contradiction). Deductive thinking and axiomatic thinking are combined in Euclid's *Elements* (who uses, like Aristotle, "common notions" for "axioms"). The great novelty brought by Euclid is the fact that, for the first time, mathematical proofs (and, through them, science in general) are built on a long distance perspective, in a step by step procedure, where you have to look permanently to previous steps and to fix your aim far away to the hypothetical subsequent steps. Euclid became, for about two thousands years, a term of reference for the axiomatic–deductive thinking, being considered the highest standard of rigour. Archimedes, in his treatise on static equilibrium, the physicists of the Middle Age (such as Jordanus de Nemore, in *Liber de ratione ponderis*, in the 13th century), B. Spinoza in *Ethics* (1677) and I. Newton in *Principia* (1687) follow Euclid's pattern. This tradition is continued in many more recent works, not only in the field of mathematics, but also in physics, computer science, biology, linguistics, etc.

However, some shortcomings of Euclid's approach were obstacles for the development of mathematical rigour. One of them was the fact that, until Galilei, the mathematical language was essentially the ordinary language, dominated by imprecision resulting from its predominantly spontaneous use, where emotional factors and lack of care have an impact. In order to diminish this imprecision and make the mathematical language capable to face the increasing need of precision and rigour, the ordinary language had to be supplemented by an artificial component of symbols, formulas and equations: with Galilei, Descartes, Newton and Leibniz, the mathematical language became more and more a mixed language, characterized by a balance between its natural and artificial components. In this

way, it was possible to pack in a convenient, heuristic way, previous concepts and results, and to refer to them in the subsequent development of mathematical inferences. To give only one example, one can imagine how difficult was to express the *n*th power of a binomial expression in the absence of a symbolic representation, i.e., using only words of the ordinary language. This was the third step in the development of mathematical proofs.

The fourth step is associated with the so–called epsilon rigour, so important in mathematical analysis; it occurred in the 19th century and it is associated with names such as A. Cauchy and K. Weierstrass. So, it became possible to renounce the predominantly intuitive approach via the infinitely small quantities of various orders, under the form of functions converging in the limit to zero (not to be confused with the Leibnizian infinitely small, elucidated in the second half of the 20th century, by A. Robinson's non–standard analysis). The epsilon rigour brought by the fourth step created the possibility to cope in a more accurate manner with processes with infinitely many steps such as limit, continuity, differentiability and integrability.

The fifth period begun with the end of the 19th century, when Aristotle's logic, underlining mathematical proofs for two thousands years, entered a crisis with the challenge of the principle of non–contradiction. This crisis was already announced by the discovery of non–Euclidean geometries, in the middle of the 19th century. Various therapies were proposed to free the mathematical proof of the dangerous effects of paradoxes (Russell–Whitehead, Hilbert, Brouwer, etc). This period covers the first three decades of the 20th century and is dominated by the optimistic view stating the possibility to arrange the whole mathematics as a formal system and to decide for any possible statement whether it is true or false. However, even during this period mathematicians were divided with respect to the acceptance of non–effective (non–constructive) entities and proofs (for example, Brouwer's intuitionism rejects the principle of excluded middle in the case of infinite sets). Intuitionism was a signal for the further development of constructive mathematics, culminating with the algorithmic approach leading to computer science.

The sixth period begins with Gödel's incompleteness theorem (1931), for many meaning the unavoidable failure of any attempt to formalise the whole of mathematics. Aristotle's requirement of complete lack of contradiction can be satisfied only by paying the price of incompleteness of the working formal system. Chaitin (1975) has continued this trend of results by proving that from N bits of axioms one cannot prove that a program is the smallest possible if it is more than N bits long; he suggested that complexity is a source of incompleteness because a formal system can capture only a tiny amount of the huge information contained in the world of mathematical truth. This principle has been proved in Calude and Jürgensen [19]. Hence, incompleteness is natural and inevitable rather then mys-

terious and esoteric. This raises the natural question (see Chaitin [26]): *How come that in spite of incompleteness, mathematicians are making so much progress?*

The seventh period belongs to the second half of the 20th century, when algorithmic proofs become acceptable only when their complexities were not too high. Constructiveness is no longer enough, a reasonable high complexity (cost) is mandatory. We are now living in this period. An important event of this period was the 1976 proof of the Four–Colour Problem (4CP): it marked the reconciliation of empirical–experimental mathematics with deductive mathematics, realized by the use of computer programs as pieces of a mathematical proof. Computer refers to classical von Neumann computer. At the horizon we can see the (now hypothetical) quantum computer which may modify radically the relation between empirical–experimental mathematics and deductive mathematics ...

With the eighth stage, proofs are no longer exclusively based on logic and deduction, but also empirical and experimental. On the other hand, in the light of the important changes brought by authors like Hilbert, already at the beginning of the 20th century, primitive terms became to have an explicit status, axioms show their dependence on physical factors and the axiomatic–deductive method displays its ludic dimension, being a play with abstract symbols. Hilbert axiomatization of geometry is essentially different from Euclid's geometry and this fact is well pointed out by Dijkstra in [31] where he considers that, by directing their attention towards provability, formalists circumvented the vague metaphysical notion of "truth". Dijkstra qualifies as "philosophical pollution" the mentality which pushed Gauss not to publish his ideas related to non–Euclidean geometry. Contrary to appearances, believes Dijkstra, Euclidean geometry is not a prototype of a deductive system, because it is based to a large extent on pictures (so–called definitions of points and lines, for instance) motivated by the need of geometric intuition. For Dijkstra, the claim that the Euclidean geometry is a model of deductive thinking, is a big lie. As a matter of fact, the shortcomings to which Dijkstra refers were well-known, as can be seen in Morris Kline's book [43], pp. 86–88. In contemporary mathematics we are facing a change of perspective, a change of scenario, replacing the old itinerary definition–theorem–proof by another one (see, for instance, W. Thurston), based on ideas, examples and motivations. The interesting fact is that the gap created between proof and intuition by Hilbert prepared the way for a new marriage between deduction and experiment, made possible by the computational revolution, as it was shown by the latest step in the evolution of proofs.

3 Proofs, Theorems and Truths

Depuis les Grecs, qui dit Mathématique,
dit démonstration. Bourbaki

What is a mathematical proof? At a first glance the answer seems obvious: a proof is a series of logical steps based on some axioms and deduction rules which reaches a desired conclusion. Every step in a proof can be checked for correctness by examining it to ensure that it is logically sound. In David Hilbert's words: "The rules should be so clear, that if somebody gives you what they claim is a proof, there is a mechanical procedure that will check whether the proof is correct or not, whether it obeys the rules or not." By making sure that every step is correct, one can tell once and for all whether a theorem has been proved. Simple! A moment of reflection shows that the problem may not be so simple. For example, what if the "agent" (human or computer) checking a proof for correctness makes a mistake (agents are fallible)? Obviously, another agent has to check that the agent doing the checking did not make any mistakes. Some other agent will need to check that agent, and so on. Eventually one runs out of agents who could check the proof and, in principle, they could all have made a mistake!

The mistake is the neighbour and the brother of proof, it is both an opponent and a stimulus. An interesting analysis, responding to Joseph L. Doob's challenge, of various possible mistakes in the proof of the 4CT can be found in the work of Schmidt [58]. In 1976, Kenneth Appel and Wolfgang Haken proved the 4CT. They used some of Alfred Kempe's ideas, but avoided his mistake.[1] They showed that if there is a map which needs five colours, then a contradiction follows. If there are several five–colour maps, they have chosen one with the smallest number of countries and proved that this map must contain one of 1,936 possible configurations; they also proved that every one of these possible configurations can be reduced into a smaller configuration which also needs five colours. This is a contradiction because we assumed that we already started with the smallest five–colour map. The reduction step, i.e., the step in which one shows that the 1,936 configurations could be reduced was actually done by brute force computer search through every configuration. No human being could ever actually read the entire proof to check its correctness. For Ron Graham, "The real question is this: If no human being can ever hope to check a proof, is it really a proof?"

In 1996 Robertson, Sanders, Seymour and Thomas [52] offered a simpler proof involving only 633 configurations. The paper [52] concludes with the following interesting comment (p. 24): "We should mention that both our programs use only integer arithmetic, and so we need not be concerned with round–off errors and similar dangers of floating point arithmetic. However, an argument can be made that our "proof" is not a proof in the traditional sense, because it contains steps that can never be verified by humans. In particular, we have not proved the

[1] In 1879 Kempe announced his 'proof' of the 4CT in both the magazine *Nature* and the *American Journal of Mathematics*. Eleven years later, Percy Heawood found an error in the proof which nobody had spotted, despite careful checking.

correctness of the compiler we compiled our programs on, nor have we proved the infallibility of the hardware we ran our programs on. These have to be taken on faith, and are conceivably a source of error. However, from a practical point of view, the chance of a computer error that appears consistently in exactly the same way on all runs of our programs on all the compilers under all the operating systems that our programs run on is infinitesimally small compared to the chance of a human error during the same amount of case–checking. Apart from this hypothetical possibility of a computer consistently giving an incorrect answer, the rest of our proof can be verified in the same way as traditional mathematical proofs. We concede, however, that *verifying a computer program is much more difficult than checking a mathematical proof of the same length*."[2]

According to Vladimir Arnold, "Proofs are to mathematics what spelling (or even calligraphy) is to poetry. Mathematical works do consist of proofs, just as poems do consist of characters." These analogies point out both the necessity and the insufficiency of proofs in the development of mathematics. Indeed, spelling is the way poetry takes expression, but it is equally the tool used by the common everyday language, in most cases devoid of any poetic effect. What should be added to spelling in order to get a piece of poetry remains a mystery. A poem consists of characters, but it is much more than a meaningful concatenation of characters.

Mathematics cannot be conceived in the absence of proofs. According to Foiaş [37], "the theorem is the brick of mathematics". Obviously, "proof" and "theorem" go together; the object of a proof is to reach a theorem, while theorems are validated by proofs. Theorems are, for the construction of mathematics, what bricks are for the construction of a building. A building is an articulation of bricks and, analogically, a mathematical work is an articulation of theorems. Motivated by a similar view, Jean Dieudonné [34] defines a mathematician as a person who has proved at least one theorem. In contrast, Arnold's analogies point out the fact that mathematics is much more than a chain of theorems and proofs, so implicitly a mathematician should be much more than the author of a theorem. Probably the best example is offered by Bernhard Riemann whose lasting fame does not come (in the first instance) from his theorems or proofs, but from his conjectures, definitions, concepts and examples (see for example, the discussion in Hersh [42], pp. 50–51). Srinivasa Ramanujan is another famous example of a mathematician who produced more results than proofs. What the mathematical community seems to value most are "ideas". "The most respected mathematicians are those with strong 'intuition' " (Harris [39], p. 19).

[2] Our Italics.

4 Mathematical Proofs: The Syntactic Dimension

Cum Deus calculat, fit mundus. Leibniz

Of course, the first thing to be discussed is Gödel's incompleteness theorem (GIT) which says that *every formal system which is (1) finitely specified, (2) rich enough to include the arithmetic, and (3) consistent, is incomplete.* That is, there exists an arithmetical statement which (A) can be expressed in the formal system, (B) is true, but (C) is unprovable within the formal system. All conditions are necessary. Condition (1) says that there is an algorithm listing all axioms and inference rules (which could be infinite). Taking as axioms all true arithmetical statements will not do, as this set is not finitely listable. But what does it mean to be a "true arithmetical statement"? It is a statement about non-negative integers which cannot be invalidated by finding any combination of non-negative integers that contradicts it. In Alain Connes terminology (see [30], p. 6), a true arithmetical statement is a "primordial mathematical reality". Condition (2) says that the formal system has all the symbols and axioms used in arithmetic, the symbols for 0 (zero), S (successor), + (plus), $\times$ (times), = (equality) and the axioms making them work (as for example, $x + S(y) = S(x + y)$). Condition (2) cannot be satisfied if you do not have individual terms for $0, 1, 2, \ldots$; for example, Tarski proved that Euclidean geometry, which refers to points, circles and lines, is complete. Finally (3) means that the formal system is free of contradictions. The essence of GIT is to distinguish between truth and provability. A closer real life analogy is the distinction between truths and judicial decisions, between what is true and what can be proved in court.[3] How large is the set of true and unprovable statements? If we fix a formal system satisfying all three conditions in GIT, then the set of true and unprovable statements is topologically "large" (constructively, a set of second Baire category, and in some cases even "larger"), cf. Calude, Jürgensen, Zimand [20]; because theorems proven in such a system have bounded complexity, the probability that an n-bit statement is provable tends to zero when n tends to infinity (see Calude and Jürgensen [19]).

There is a variety of reactions in interpreting GIT, ranging from pessimism to optimism or simple dismissal (as irrelevant for the practice of mathematics). For pessimists, this result can be interpreted as the final, definite failure of any attempt to formalise the whole of mathematics. For example, Hermann Weyl acknowledged that GIT has exercised a "constant drain on the enthusiasm" with which he has engaged himself in mathematics and for Stanley Jaki, GIT is a fundamental barrier in understanding the Universe. In contrast, scientists like Freeman Dyson acknowledge the limit placed by GIT on our ability to discover the truth in math-

[3] The Scottish judicial system which admits three forms of verdicts, guilty, not–guilty and not–proven, comes closer to the picture described by GIT.

ematics, but interpret this in an optimistic way, as a guarantee that mathematics will go on forever (see Barrow [6], pp. 218–221).

In modern times a penetrating insight into the incompleteness phenomenon has been obtained by an information–theoretic analysis pioneered by Chaitin in [24]. Striking results have been obtained by studying the Chaitin's Omega Number, Ω, the halting probability of a self-delimiting universal Turing machine. This number is not only uncomputable, but also (algorithmically) random. Chaitin has proven the following important theorem: *If ZFC (Zermelo set theory with the Axiom of Choice) is arithmetically sound, that is, any theorem of arithmetic proved by ZFC is* true, *then, ZFC can determine the value of only finitely many bits of* Ω*, and one can give a bound on the number of bits of* Ω *which ZFC can determine.* Robert Solovay [56] (see more in [17, 15, 16, 14]) has constructed *a self-delimiting universal Turing machine such that ZFC, if arithmetically sound, cannot determine any single bit of its halting probability* (Ω). Re–phrased, the most powerful formal axiomatic system is powerless when dealing with the questions of the form "is the mth bit of Ω 0?" or "is the mth bit of Ω 1?".

Chaitin has constructed an exponential Diophantine equation $F(t; x_1, \ldots, x_n) = 0$ with the following property: the infinite binary sequence whose mth term is 0 or 1 depending whether the equation $F(m; x_1, \ldots, x_n) = 0$ has finitely or infinitely many solutions is exactly the digits of Ω, hence it is random; its infinite amount of information is algorithmically incompressible. The importance of exponential Diophantine equations comes from the fact that most problems in mathematics can be formulated in terms of these type of equations; Riemann's Conjecture is one such example. Manin [46], p. 158, noticed that "The epistemologically important point is the discovery that randomness can be defined without any recourse to physical reality ... in such a way that the necessity to make an infinite search to solve a parametric series of problems leads to the technically random answers. Some people find it difficult to imagine that a rigidly determined discipline like elementary arithmetic may produce such phenomena".

Last but not least, is the truth achieved through a formal proof the ultimate expression of knowledge? Many (mathematicians) will give a positive answer, but perhaps not all. For the 13th century Oxford philosopher Roger Bacon, "Argument reaches a conclusion and compels us to admit it, but it neither makes us certain nor so it annihilates doubt that the mind rests calm in the intuition of truth, unless it finds this certitude by way of experience." More recently, I. J. Schoenberg[4] is cited by Epstein ([38]) as saying that Edmund Landau kept in his desk drawer for years a manuscript proving what is now called the two constants theorem: he had the complete proof but could not believe it until his intuition was ready to accept it. Then he published it. A "proof is only one step in the direction of confidence"

[4]Landau's son-in-law.

argued De Millo, Lipton and Perlis in a classical paper on proofs, theorems and programs [33]. Written in the same spirit is Don Knuth's warning: "Beware of bugs in the above code: I have only proved it correct, not tried it."

5 Mathematical Proofs: The Semantic Dimension

If one must choose between rigour and meaning,
I shall unhesitatingly choose the latter. R. Thom

The above quotation turned slogan as "more rigour, less meaning", or better still, "less rigour, more meaning" (Chaitin [27]) points out the necessity to distinguish between the syntactic and the semantic aspects of proofs. Should proofs belong exclusively to logic, according to the tradition started by Greeks such as Pythagoras and Euclid? Or should they also be accepted as a cocktail of logical and empirical–experimental arguments, as in the proof of the 4CT (1976)? Mathematicians are now divided into those giving an affirmative answer to the first question and implicitly a negative answer to the second question and those giving a negative answer to the first question and an affirmative one to the second question. Computationally oriented mathematicians usually belong to the second category, while many other mathematicians (as, for instance, the Fields medalist William Thurston) belong to the first, so for them, the 4CT is not yet proved! Meaning is a key distinction. For mathematicians such as René Thom, Daniel Cohen and William Thurston, correctness by itself does not validate a proof; it is also necessary to "understand" it. "The mission of mathematics is understanding" says Cohen. Paul Halmos has also insisted on the "conceptual understanding". For him a "good" proof of a theorem is one that sheds light on why it is true. It is just the process of understanding which is in question with proofs like that given to the 4CT. Referring to the proof of the 4CT, Halmos says: "I do not find it easy to say what we learned from all that. ... The present proof relies in effect on an Oracle, and I say down with Oracles! They are not mathematics!" In contrast with Halmos, who hopes that "100 years from now the map theorem will be ... an exercise in a first–year graduate course, provable in a couple of pages by means of appropriate concepts, which will be completely familiar by then" (see [42], p. 54), R. Hersh thought that the problem itself might be responsible for the way it was solved: he is cited by saying dejectedly "So it just goes to show, it wasn't a good problem after all" (see [23] p. 73).

We will return later to these issues. For the moment we make the following two observations.

A) Not only the hybrid proofs obtained as a combination of logical and empirical–experimental arguments might be hard/impossible to be understood in their

"globality"; this happens also for some pure deductive proofs. An example is the proof of the classification of finite simple groups called by Danny Gorenstein the "Thirty Years War" (for the classification battles were fought mostly in the decades 1950–1980), a work which comprises about 10,000–15,000 pages scattered in 500 journal articles by some 100 authors.[5]

According to Knuth [44] p. 18, "... program–writing is substantially more demanding than book–writing". "Why is this so? I think the main reason is that a larger attention span is needed when working on a large computer program than when doing other intellectual tasks. ... Another reason is ... that programming demands a significantly higher standard of accuracy. Things don't simply have to make sense to another human being, they must make sense to a computer." Knuth compares his TEX compiler (a document of about 500 pages) with Feit and Thompson [35] theorem that all simple groups of odd order are cyclic. He lucidly argues that the program might not incorporate as much creativity and "daring" as the proof of the theorem, but they come even when compared on depth of details, length and paradigms involved. What distinguishes the program from the proof is the "verification": convincing a couple of (human) experts that the proof *works in principle* seems to be easier than making sure that the program *really works*. A demonstration that *there exists a way to compile TEX* is not enough! Another example, which will be discussed later in this section, is the proof of Fermat's Last Theorem (FLT).

B) Without diminishing in any way the "understanding" component of mathematics we note that the idea of distinguishing between "good" and "bad" proofs on the light they shed on their own truth seems to be, at least to some extent, relative and subjective.

Thom's slogan 'more rigour, less meaning' was the main point in his controversy with Jean Dieudonné (as a representative of the Bourbaki group). Taking rigour as something that can be acquired only at the expense of meaning and conversely, taking meaning as something that can be obtained only at the expense of rigour, we oblige mathematical proof to have the status of what was called in physics a "conjugate (complimentary) pair", i.e., a couple of requirements, each of them being satisfied only at the expense of the other (see [47]). Famous prototypes of conjugate pairs are (position, momentum) discovered by W. Heisenberg in quantum mechanics and (consistency, completeness) discovered by K. Gödel in logic. But similar warnings come from other directions. According to Einstein (see, for instance, [53] p. 195), "in so far as the propositions of mathematics

[5] Still, there is a controversy in the mathematical community on whether these articles provide a complete and correct proof. For a recent account see Aschbacher [1].

are certain, they do not refer to reality, and in so far as they refer to reality, they are not certain", hence (certainty, reality) is a conjugate pair. Obviously, reality is here understood as an empirical entity, hence mixed with all kinds of imprecision, ranging from obscurity and disorder to randomness, ambiguity and fuzziness [48]. Pythagoras' theorem is certain, but its most empirical tests will fail. There are some genuine obstacles in our attempts to eliminate or at least to diminish the action of various sources of imprecision. Einstein implicitly calls our attention on one of them. Proof, to the extent to which it wants to be rigorous, to give us the feeling of certainty, should be mathematical; but satisfying this condition, means failing to reach reality. In other words, the price we have to pay to obtain proofs giving us the feeling of total confidence is to renounce to be directly connected to reality. There is a genuine tension between certainty and reality, they form a conjugate pair, which is the equivalent of what in the field of humanities is an oxymoronic pair. However, there is an essential difference between Gödel's conjugate pair (consistency, completeness) and Einstein's conjugate pair (certainty, reality). While consistency and completeness are binary logical predicates, certainty and reality are a matter of degree, exactly like the terms occurring in Thom's conjugate pair: rigour and meaning. In last two situations there is room for manipulation and compromise.

Near to the above conjugate pairs is a third one: (rigour, reality), attributed to Socrates (see [51]). A price we have to pay in order to reach rigour is the replacement of the real world by a fictional one. There is no point and no line in the real world, if we take them according to their definitions in Euclid's *Elements*. Such entities belong to a fictional/virtual universe, in the same way in which the characters of a theatrical play are purely conventional, they don't exist as real persons. The rules of deduction used in a mathematical proof belong to a game in the style they are described in the scenario of a Hilbert formal system, which is, as a matter of fact, a machine producing demonstrative texts. A convention underlines the production of theorems and again a convention is accepted in a theatrical play. In the first case, the acceptance of the convention is required from both the author of the proof and its readers; in the second case all people involved, the author, the actors and spectators, have to agree the proposed convention. Since many proofs, if not most of them, are components of a modeling process, we have to add the unavoidable error of approximation involved in any cognitive model. The model should satisfy opposite requirements, to be as near as possible to the phenomenon modelled, in order to be relevant; to be as far as possible from the respective phenomenon, in order to useful, to make possible the existence of at least one method or tool that can be applied to the model, but not to the original (see [49]). Theorems are discovered, models are invented. Their interaction leads to many problems of adequacy, relevance and correctness, i.e., of syntactic, semantic and pragmatic nature.

In the light of the situations pointed out above, we can understand some ironical comments about what a mathematician could be. It is somebody who can prove theorems, as Dieudonné claimed. But what kind of problems are solved in this way? "Any problem you want, ... except those you need", said an engineer, disenchanted by his collaboration with a mathematician. Again, what is a mathematician? "It is a guy capable to give, after a long reflection, a precise, but useless answer", said another mathematician with a deep feeling of self irony. Remember the famous reflection by Goethe: "Mathematicians are like French people, they take your question, they translate it in their language and you no longer recognize it".

But things are controversial even when they concern syntactic correctness. In this respect, we should distinguish two types of syntactic mistakes: benign and malign. Benign mistakes have only a local, not global effect: they can be always corrected. Malign mistakes, on the contrary, contaminate the whole approach and invalidate the claim formulated by the theorem. When various authors (including the famous probabilist J. L. Doob, see [58]) found some mistakes in the proof of the 4CT, the authors of the proof succeeded in showing that all of them were benign and more than this, *any other possible mistake, not yet discovered, should be benign*. How can we accept such arguments, when the process of global understanding of the respective proof is in question? The problem remains open. A convenient, but fragile, solution is to accept Thom's pragmatic proposal: a theorem is validated if it has been accepted by a general agreement[6] of the mathematical community (see [54, 55]).

The problems raised by the 4CT were discussed by many authors, starting with Tymoczko [57] and Swart [59] (more recent publications are D. MacKenzie [45], J. Casti [23], A.S. Calude [13]). Swart proposed the introduction of a new entity called *agnogram*, which is "a theorem–like statement that we have verified as best we could, but whose truth is not known with the kind of assurance we attach to theorems and about which we must thus remain, to some extent, agnostic." There is however the risk to give the status of agnogram to any property depending on a natural number n and verified only for a large, but finite number of values of n. This fact would be in conflict with Swart's desire to consider an agnogram less than a theorem, but more than a conjecture. Obviously, the 4CT is for Swart an agnogram, not a theorem. What is missing from an agnogram to be a theorem? A theorem is a statement which could be checked individually by a mathematician and confirmed also individually by at least two or three more mathematicians, each of them working independently. But already here we can observe the weakness of the criterion: how many mathematicians are to check individually and independently the status of an agnogram to give it the status of theorem?

[6] Perhaps "general" should be replaced here by "quasi–general".

The seriousness of this objection can be appreciated by examining the case of Andrew Wiles' proof of FLT—a challenge to mathematics since 1637 when Pierre de Fermat wrote it into the margin of one of his books. The proof is extremely intricate, quite long (over 100 printed pages[7]), and only a handful of people in the entire world can claim to understand it.[8] To the rest of us, it is utterly incomprehensible, and yet we all feel entitled to say that "the FLT has been proved". On which grounds? We say so because *we believe the experts* and *we cannot tell for ourselves*. Let us also note that in the first instance the original 1993 proof seemed accepted, then a gap was found, and finally it took Wiles and Richard Taylor another year to fix the error.[9]

According to Hunt [41], "In no other field of science would this be good enough. If a physicist told us that light rays are bent by gravity, as Einstein did, then we would insist on experiments to back up the theory. If some biologists told us that all living creatures contain DNA in their cells, as Watson and Crick did in 1953, we wouldn't believe them until lots of other biologists after looking into the idea agreed with them and did experiments to back it up. And if a modern biologist were to tell us that it were definitely possible to clone people, we won't really believe them until we saw solid evidence in the form of a cloned human being. Mathematics occupies a special place, where we believe anyone who claims to have proved a theorem on the say—so of just a few people—that is, until we hear otherwise."

Suppose we loosely define a religion as any discipline whose foundations rest on an element of faith, irrespective of any element of reason which may be present. Quantum mechanics, for example, would qualify as a religion under this definition. Mathematics would hold the unique position of being a branch of theology possessing a "proof" of the fact that it should be so classified. "Where else do you have absolute truth? You have it in mathematics and you have it in religion, at least for some people. But in mathematics you can really argue that this is as close to absolute truth as you can get" says Joel Spencer.

[7] Probabilists would argue that very long proofs can at best be viewed as only probably correct, cf. [33], p. 273. In view of [19], the longer the statement, the lesser its chance is to be proved.

[8] Harris [39] believes that no more than 5% of mathematicians have made the effort to work through the proof. Does this have anything to do with what George Hardy has noted in his famous *Apology*: "All physicists and a good many quite respectable mathematicians are contemptuous about proof."?

[9] According to Wiles, "It was an error in a crucial part of the argument, but it was something so subtle that I'd missed it completely until that point. The error is so abstract that it can't really be described in simple terms. Even explaining it to a mathematician would require the mathematician to spend two or three months studying that part of the manuscript in great detail."

6 Mathematical Proofs: The Pragmatic Dimension

Truth is not where you find it, but where you put it. A. Perlis

In the second half of the 20th century, theorems together with their proofs occur with increasing frequency as components of some cognitive models, in various areas of knowledge. In such situations we are obliged to question the theorems not only with respect to their truth value, but also in respect to their adequacy and relevance within the framework of the models to which they belong. We have to evaluate the explanatory capacity of a theorem belonging to a model B, concerning the phenomenon A, to which B is referring. This is a very delicate and controversial matter, because adequacy, relevance and explanatory capacity are a matter of degree and quality, which cannot be settled by binary predicates. Moreover, there is no possibility of optimization of a cognitive model. Any model can be improved, no model is the best possible. This happens because, as we have explained before, a cognitive model B of an entity A has simultaneously the tendency to increase its similarity with A and stress its difference from A. To give only one example in this respect, we recall the famous result obtained by Chomsky [29], in the late 1950s, stating that context–free grammars are not able to generate the English language. This result was accepted by the linguistic and computer science communities until the eighties, when new arguments pointed out the weakness of Chomsky's argument; but this weakness was not of a logical nature, it was a weakness in the way we consider the entity called "natural language". As a matter of fact, the statement "English is a context–free language" is still controversial.

Mathematical proofs are "theoretical" and "practical". Theoretical proofs (formal, ideal, rigorous) are models for practical proofs (which are informal, imprecise, incomplete). "Logicians don't tell mathematicians what to do. They make a theory out of what mathematicians actually do", says Hersh [42], p. 50. According to the same author, logicians study what mathematicians do the way fluid dynamicists study water waves. Fluid dynamicists don't tell water how to wave, so logicians don't tell mathematicians what to do. The situation is not as simple as it appears. Logical restrictions and formal models (of proof) can play an important role in the practice of mathematics. For example, the key feature of constructive mathematics is the identification "existence = computability" (cf. Bridges [12]) and a whole variety of constructive mathematics, the so–called Bishop constructive mathematics, is mathematics with intuitionistic rather than classical underlying logic.

7 Quasi–Empirical Proofs: From Classical to Quantum

Truth does not change because it is, or is not, believed by a majority of the people. Giordano Bruno

The use of large–scale programs, such as Mathematica, Maple or MathLab is now widespread for symbolical and numerical calculations as well as for graphics and simulations. To get a feeling of the extraordinary power of such programs one can visit, for example, the Mathematica website `http://www.wolfram.com`. New other systems are produced; "proofs as programs", "proof animation" or "proof engineering" are just a few examples (see [40]). In some cases an experiment conveys an aesthetic appreciation of mathematics appealing to a much broader audience (cf. [7, 8, 21]). A significant, but simple example of the role an experiment may play in a proof is given by Beyer [9]. He refers to J. North who asked for a computer demonstration that the harmonic series diverges. We quote Beyer: "His example illustrates the following principle: Suppose that one has a computer algorithm alleged to provide an approximation to some mathematical quantity. Then the algorithm should be accompanied by a theorem giving a measure of the distance between the output of the algorithm and the mathematical quantity being approximated. For the harmonic series, one would soon find that the sum was infinite." It is interesting to mention that in 1973 Beyer made together with Mike Waterman a similar attempt to compute Euler's constant; their experiment failed, but the error was discovered later by Brent [11].

New types of proofs motivated by the experimental "ideology" have appeared. For example, rather than being a static object, the *interactive proof* (see Goldwasser, Micali, Rackoff [36], Blum [10]) is a two–party protocol in which the *prover* tries to prove a certain fact to the *verifier*. During the interactive proof the *prover* and the *verifier* exchange messages and at the end the *verifier* produces a verdict "accept" or "reject". A holographic (or probabilistic checkable) proof (see Babai [4]) is still a static object but it is verified probabilistically. Errors become almost instantly apparent after a small part of the proof was checked.[10] The transformation of a classical proof (which has to be self-contained and formal) into a holographic one requires super-linear time.

The blend of logical and empirical–experimental arguments ("quasi–empirical mathematics" for Tymoczko [57], Chaitin [25, 26, 28] or "experimental mathematics" for Bailey, Borwein [5], Borwein, Bailey [7], Borwein, Bailey, Girgensohn [8]) may lead to a new way to understand (and practice) mathematics. For example, Chaitin argued that we should introduce the Riemann hypothesis as an

[10] More precisely, a traditional proof of length l is checked in time a constant power of l while a holographic proof requires only constant power of $\log_2 l$. To appreciate the difference, the binary logarithm of the number of atoms in the known Universe is smaller than 300.

axiom: “I believe that elementary number theory and the rest of mathematics should be pursued more in the spirit of experimental science, and that you should be willing to adopt new principles. I believe that Euclid’s statement that an axiom is a self–evident truth is a big mistake. The Schrödinger equation certainly isn’t a self–evident truth! And the Riemann hypothesis isn’t self–evident either, but it’s very useful. A physicist would say that there is ample experimental evidence for the Riemann hypothesis and would go ahead and take it as a working assumption." Classically, there are two equivalent ways to look at the mathematical notion of proof: *logical*, as a finite sequence of sentences strictly obeying some axioms and inference rules, and *computational*, as a specific type of computation. Indeed, from a proof given as a sequence of sentences one can easily construct a Turing machine producing that sequence as the result of some finite computation and, conversely, given a machine computing a proof we can just print all sentences produced during the computation and arrange them into a sequence.

This gives mathematics an immense advantage over any science: a proof is an explicit sequence of reasoning steps that can be inspected at *leisure*. *In theory*, if followed with care, such a sequence either reveals a gap or mistake, or can convince a sceptic of its conclusion, in which case the theorem *is considered proven*. The equivalence between the logical and computational proofs has stimulated the construction of programs which play the role of *“artificial" mathematicians*. The “theorem provers" have been very successful as “helpers" in proving many results, from simple theorems of Euclidean geometry to the computation of a few digits of a Chaitin Omega Number [18]. “Artificial" mathematicians are far less ingenious and subtle than human mathematicians, but they surpass their human counterparts by being infinitely more patient and diligent.

If a conventional proof is replaced by an “unconventional" one (that is a proof consisting of a sequence of reasoning steps obeying axioms and inference rules which depend not only on some logic, but also on the external physical medium), then the conversion from a computation to a sequence of sentences may be impossible, e.g. due to the size of the computation. An extreme, and for the time being hypothetical example, is the proof obtained as a result of a quantum computation (see Calude and Păun [22]). The quantum automaton would say “your conjecture is true", but (due to quantum interference) there will be no way to exhibit all trajectories followed by the quantum automaton in reaching that conclusion. The quantum automaton has the ability to check a proof, but it may fail to reveal any “trace" of the proof for the human being operating the quantum automaton. Even worse, any attempt to *watch* the inner working of the quantum automaton (e.g. by “looking" inside at any information concerning the state of the ongoing proof) may compromise forever the proof itself! We seem to go back to Bertrand Russell who said that “mathematics may be defined as the subject in which we never know what we are talking about, nor whether what we are saying is true",

and even beyond by adding *and even when it's true we might not know why.*

Speculations about quantum proofs *may not affect* the essence of mathematical objects and constructions (which, many believe, have an autonomous reality quite independent of the physical reality), but they seem to *have an impact* on how we *learn/understand mathematics,* which is through the physical world. Indeed, our glimpses of mathematics are revealed only through physical objects, human brains, silicon computers, quantum automata, etc., hence, according to Deutsch [32], they have to obey not only the axioms and the inference rules of the theory, but the *laws of physics* as well. To complete the picture we need to take into account also the *biological* dimension. No matter how precise the rules (logical and physical) are, we need human consciousness to apply the rules and to understand them and their consequences. Mathematics is a human activity.

8 Knowledge Versus Proof

The object of mathematical rigour is to sanction and legitimize the conquests of intuition. J. Hadamard

Are there intrinsic differences between traditional and 'unconventional' types of proofs? To answer this question we will consider the following interrelated questions:

1. Do 'unconventional' methods supply us with a proof in some formal language?
2. Do 'unconventional' methods supply us with a mathematical proof?
3. Do 'unconventional' methods supply us with knowledge?
4. Does mathematics require knowledge or proof?

A blend of mathematical reasoning supported by some silicon or quantum computation or a classical proof of excessive length and complexity (for example, the classification of finite simple groups) are examples of "unconventional" proofs. The ultimate goal of the mathematical activity is the *advance human understanding of mathematics* (whatever this means!).

The answer to the first two question is affirmative. Indeed, computations are represented in the programming language used by the computer (the 'unconventional' computer too), even if the whole proof cannot be globally 'visualized' by a human being. The proof can be checked by any other mathematician having the equipment used in the 'unconventional' proof. A proof provides knowledge only to the extent that its syntactic dimension is balanced by the semantic one; any gap between them makes the proof devoid of knowledge and paves the way for the proof to become a ritual without meaning. Proofs generating knowledge, quite

often produce much more, for example, 'insight' (think of the insight provided by understanding the algorithm used in the proof).

A misleading analogy would be to replace, in the above questions, *'unconventional' methods* with *"testimony from a respected and (relevantly) competent mathematician"*. Certainly, such testimony provides knowledge; it does not qualify as a mathematical proof (even less as a formalized proof), but the result is a "mathematical activity" because it advances our knowledge of mathematics. The difference between 'unconventional' methods and 'relevant testimony' can be found in the mechanisms used to produce outputs: a 'relevant testimony' is the gut feeling of a respected, relevant, competent mathematician, by and large based on a considerable mathematical experience, while an 'unconventional' method produces an objective argument.

There is little 'intrinsic' difference between traditional and 'unconventional' types of proofs as i) first and foremost, *mathematical truth* cannot always be certified by proof, ii) correctness is not absolute, but almost certain, as mathematics advances by making mistakes and correcting and re–correcting them (mathematics fallibility was argued by Lakatos), iii) non–deterministic and probabilistic proofs do not allow mistakes in the applications of rules, they are just indirect forms of checking (see Pollack [50], p. 210) which correspond to various degrees of rigour, iv) the explanatory component, the understanding 'emerging' from proofs, while extremely important from a cognitive point of view, is subjective and has no bearing on formal correctness. As Hersh noticed, mathematics like music exists by some logical, physical and biological manifestation, but "it makes sense only as a mental and a cultural activity" ([42], p. 22).

How do we continue to produce rigorous mathematics when more research will be performed in large computational environments where we might or might not be able to determine what the system has done or why[11] is an open question. The blend of logical and empirical–experimental arguments are here to stay and develop. Of course, some will continue to reject this trend, but, we believe, they will have as much effect as King Canute's royal order to the tide. There are many reasons which support this prediction. They range from economical ones (powerful computers will be more and more accessible to more and more people), social ones (skeptical oldsters are replaced naturally by youngsters born with the new technology, results and success inspire emulation) to pure mathematical (new challenging problems, wider perspective) and philosophical ones (note that incompleteness is based on the analysis of the computer's behaviour). The picture holds marvelous promises and challenges; it does not eliminate the continued importance of extended personal interactions in training and research.

[11] Metaphorically described as "relying on proof by 'Von Neumann says'".

Acknowledgements

This paper is based on a talk presented at the Workshop *Truths and Proofs*, a satellite meeting of the *Annual Conference of the Australasian Association of Philosophy (New Zealand Division)*, Auckland, New Zealand, December 2001. We are most grateful to Andreea Calude, Greg Chaitin, Sergiu Rudeanu, Karl Svozil, Garry Tee, and Moshe Vardi for inspired comments and suggestions.

References

[1] M. Aschbacher. The status of the classification of finite simple groups, *Notices of the Amer. Math. Soc.*51, 7 (2004), 736–740.

[2] D. Auburn. *Proof. A Play*, Faber and Faber, New York, 2001.

[3] K. Appel, W. Haken. *Every Planar Graph is Four Colorable*, Contemporary Mathematics 98, AMS, Providence, 1989.

[4] L. Babai. Probably true theorems, cry wolf? *Notices of the Amer. Math. Soc.* 41 (5) (1994), 453–454.

[5] D. H. Bailey, J. M. Borwein. Experimental mathematics: Recent developments and future outlook, in B. Engquist, W. Schmid (eds.). *World Mathematical Year Mathematics Unlimited—2001 and Beyond*, Springer-Verlag, Berlin, 2001, 51–66.

[6] J. Barrow. *Impossibility. The Limits of Science and the Science of Limits*, Oxford University Press, Oxford, 1998.

[7] J. M. Borwein, D. H. Bailey, *The Experimental Mathematician. Plausible Reasoning in the 21st Century*, A. K. Peters, Natick, Ma., 2003.

[8] J. M. Borwein, D. H. Bailey, R. Girgensohn. *Experimentation in Mathematics. Computational Paths to Discovery*, A. K. Peters, Natick, Ma., 2004.

[9] W. A. Beyer. The computer and mathematics. *Notices of the Amer. Math. Soc.* 48, 11 (2001), 1302.

[10] M. Blum. How to prove a theorem so no one else can claim it, *Proceedings of the International Congress of Mathematicians*, Berkeley, California, USA, 1986, 1444–1451.

[11] R. P. Brent. Computation of the regular continued fraction for Euler's constant, *Math. of Computation* 31, 139 (1977), 771–777.

[12] D. S. Bridges. Constructive truth in practice, in H. G. Dales and G. Oliveri (eds.). *Truth in Mathematics*, Clarendon Press, Oxford, 1998, 53–69.

[13] A. S. Calude. The journey of the four colour theorem through time, *The NZ Mathematics Magazine* 38, 3 (2001), 27–35.

[14] C. S. Calude. *Information and Randomness: An Algorithmic Perspective*, 2nd Edition, Revised and Extended, Springer Verlag, Berlin, 2002.

[15] C. S. Calude. Chaitin Ω numbers, Solovay machines and incompleteness, *Theoret. Comput. Sci.*, 284 (2002), 269–277.

[16] C. S. Calude. Incompleteness, complexity, randomness and beyond, *Minds and Machines: Journal for Artificial Intelligence, Philosophy and Cognitive Science*, 12, 4 (2002), 503–517.

[17] C. S. Calude, G. J. Chaitin. Randomness everywhere, *Nature*, 400, 22 July (1999), 319–320.

[18] C. S. Calude, M. J. Dinneen and C.-K. Shu. Computing a glimpse of randomness, *Experimental Mathematics* 11, 2 (2002), 369–378.

[19] C. S. Calude, H. Jürgensen. *Is Complexity a Source of Incompleteness?*, *CDMTCS Research Report* 241, 2004, 15 pp.

[20] C. Calude, H. Jürgensen, M. Zimand. Is independence an exception ?, *Appl. Math. Comput.* 66 (1994), 63–76.

[21] C. S. Calude, S. Marcus. Mathematical proofs at a crossroad? in J. Karhumäki, H. Maurer, G. Păun, G. Rozenberg (eds.). *Theory Is Forever*, Lectures Notes in Comput. Sci. 3113, Springer Verlag, Berlin, 2004, 15–28.

[22] C. S. Calude, G. Păun. *Computing with Cells and Atoms*, Taylor & Francis Publishers, London, 2001.

[23] J. L. Casti. *Mathematical Mountaintops*, Oxford University Press, Oxford, 2001.

[24] G. J. Chaitin. Randomness and mathematical proof, *Scientific American*, 232 (5) (1975), 47–52.

[25] G. J. Chaitin. *Exploring Randomness*, Springer Verlag, London, 2001.

[26] G. J. Chaitin. Computers, paradoxes and the foundations of mathematics, *American Scientist*, 90 March–April (2002), 164–171.

[27] G. J. Chaitin. Personal communication to C. S. Calude, 5 March, 2002.

[28] G. J. Chaitin. On the intelligibility of the universe and the notions of simplicity, complexity and irreducibility, `http://www.cs.auckland.ac.nz/CDMTCS/chaitin/bonn.html`, September 2002.

[29] N. Chomsky. *Syntactic Structures*, Mouton, The Hague, 1957.

[30] A. Connes, A. Linchnerowicz, M. P. Schützenberger. *Triangle of Thoughts*, AMS, Providence, 2001.

[31] E. W. Dijkstra. Real mathematicians don't prove, *EWD1012*, University of Texas at Austin, 1988, `http://www.cs.utexas.edu/users/EWD/EWD1012.pdf`.

[32] D. Deutsch. *The Fabric of Reality*, Allen Lane, Penguin Press, 1997.

[33] R. A. De Millo, R. J. Lipton, A. J. Perlis. Social processes and proofs of theorems and programs, *Comm. ACM* 22, 5 (1979), 271–280.

[34] J. Dieudonné. *Pour L'honneur de l'Esprit Humain*, Gallimard, Paris, 1986.

[35] W. Feit, J. G. Thomson. Solvability of groups of odd order, *Pacific J. Math.* 13 (1963), 775–1029.

[36] S. Goldwasser, S. Micali, C. Rackoff. The knowledge complexity of interactive proof–systems, *SIAM Journal of Computing*, 18(1) (1989), 186–208.

[37] C. Foiaş. Personal communication to S. Marcus (about 20 years ago).

[38] L. E. Hahn, B. Epstein. *Classical Complex Analysis*, Sudury, Mass., Jones and Barlettt, 1996.

[39] M. Harris. Contexts of justification, *The Mathematical Intelligencer* 23, 1 (2001), 10–22.

[40] S. Hayashi, R. Sumitomo, K. Shii. Towards the animation of proofs–testing proofs by examples, *Theoret. Comput. Sci.* 272 (2002), 177–195.

[41] R. Hunt. The philosophy of proof, `http://plus.maths.org/issue10/features/proof4/`.

[42] R. Hersh. *What Is Mathematics, Really?*, Vintage, London, 1997.

[43] M. Kline. *Mathematical Thought from Ancient to Modern Times*, Oxford University Press, Oxford, Vol. 1, 1972.

[44] D. E. Knuth. Theory and practice, *EATCS Bull.* 27 (1985), 14–21.

[45] D. MacKenzie. Slaying the kraken. The sociohistory of a mathematical proof, *Social Studies of Science* 29, 2 (1999), 7–60.

[46] Yu. I. Manin. Truth, rigour, and common sense, in H. G. Dales and G. Oliveri (eds.). *Truth in Mathematics*, Clarendon Press, Oxford, 1998, 147–159.

[47] S. Marcus. No system can be improved in all respects, in G. Altmann and W. A. Koch (eds.). *Systems. New Paradigms for the Human Sciences*, Walter de Gruyter, Berlin, 1998, 143–164.

[48] S. Marcus. Imprecision, between variety and uniformity: the conjugate pairs, in J. J. Jadacki and W. Strawinski (eds.). *The World of Signs*, Poznan Studies in the Philosophy of Sciences and the Humanities 62 Rodopi, Amsterdam, 1998 59–72.

[49] S. Marcus. Metaphor as dictatorship, in J. Bernard, J. Wallmannsberger and G. Withalm (eds.). *World of Signs. World of Things*, Angewandte Semiotik 15, OGS, Wien, 1997, 87–108.

[50] R. Pollack. How to believe a machine–checked proof, in G. Sambin and J. M. Smith (eds.). *Twenty–five Years of Constructive Type Theory*, Clarendon Press, Oxford, 1998, 205–220.

[51] A. Rényi. *Dialogues on Mathematics*, Holden Day, San Francisco, 1967.

[52] N. Robertson, D. Sanders, P. Seymour, R. Thomas. A new proof of the four–colour theorem, *Electronic Research Announcements of the AMS* 2,1 (1996), 17–25.

[53] R. R. Rosen. Complementarity in social structures, *Journal of Social and Biological Structures* 1 (1978), 191–200.

[54] R. Thom. Modern mathematics: does it exist?, in A. G. Howson (ed.). *Developments in Mathematical Education*, Cambridge University Press, 1973, 194–209.

[55] R. Thom. Topologie et linguistique, in A. Haefliger and R. Nerasimham (eds.). *Essays on Topology and Related Topics. Memoires Dédiés à Georges de Rham*, Springer Verlag, New York, 1970, 226–248.

[56] R. M. Solovay. A version of Ω for which *ZFC* can not predict a single bit, in C. S. Calude and G. Păun (eds.). *Finite Versus Infinite. Contributions to an Eternal Dilemma*, Springer Verlag, London, 2000, 323–334.

[57] T. Tymoczko. The four–colour problem and its philosophical significance, *J. Philosophy* 2,2 (1979), 57–83.

[58] U. Schmidt. *Überprüfung des Beweises für den Vierfarben Satz*, Diplomarbeit Technische Hochschule, Aachen, 1982.

[59] E. R. Swart. The philosophical implications of the four–colour problem, *American Mathematical Monthly* 87, 9 (1980), 697–702.

MONOTONE ALGEBRAS, $\mathcal{R}$-TRIVIAL MONOIDS AND A VARIETY OF TREE LANGUAGES

Ville Piirainen
Turku Centre for Computer Science
Lemminkäisenkatu 14 A
FIN-20520 Turku, Finland
visapi@utu.fi.

Abstract

In this paper we re-examine a characterization of monotone algebras by Gécseg and Imreh [3]. We prove that the class of finite algebras whose translation monoids are $\mathcal{R}$-trivial is exactly the class of finite monotone algebras. As a result we obtain an example of a family of algebras which is defined by a simple property of algebras, but also has a characterization by a variety of finite monoids. Finally, we discuss some connections of the result to tree language theory.

1 Introduction

Several different frameworks have been proposed for the classification of regular tree languages. In this paper we follow the ideas from [7] and refer to it for the notation and concepts that are not presented here.

Every $*$-variety of string languages can be characterized by a variety of finite monoids (VFM) [2]. For tree languages the varieties of finite monoids do not work as well. In [7], a correspondence theorem between generalized varieties of finite algebras (GVFAs), general varieties of tree languages (GVTLs) and varieties of finite g-congruences (GVFCs) is presented, and it generalizes the classification theory on string languages very naturally. Indeed, many interesting classes of regular tree languages are GVTLs that arise as natural generalizations of varieties of string languages. However, few of these GVTLs can be characterized by syntactic monoids. On the other hand, for every variety of finite monoids there exists a unique corresponding general variety of tree languages, but these are not always very interesting as tree language families. In many cases, even if we have a natural

definition for some GVTL, GVFA, GVFC or VFM, finding a good characterization for even one of the other corresponding classes seems to be quite difficult. In this paper we provide one example where at least a VFM and the corresponding GVFA both have natural and seemingly independent defining properties.

In [3], Gécseg and Imreh characterize the classes of monotone automata, monotone top-down tree recognizers and monotone bottom-up tree recognizers by their syntactic monoids. However, they do not state the fact that the class of monoids they are considering actually forms a variety of finite monoids, namely that of $\mathcal{R}$-trivial monoids. In this paper we prove this fact about the monotone algebras (bottom-up tree recognizers) with basically the same proofs. Since finite monotone algebras correspond exactly to finite $\mathcal{R}$-trivial monoids, we get many closure properties of the classes of monotone algebras and monotone tree languages for free from the general theory.

It is quite obvious that these same proofs can be used also to characterize monotone string automata by $\mathcal{R}$-trivial monoids, but this characterization is already known (cf. [1] for example).

2 Preliminaries

In the following, Σ denotes a finite ranked alphabet and Σ_m, for every $m \geq 0$, is the set of m-ary symbols in Σ. We use the symbol X to denote a finite leaf alphabet and by $T_\Sigma(X)$ we denote the set of ΣX-trees defined in the usual way.

A Σ-algebra $\mathcal{A} = (A, \Sigma)$ consists of the set A equipped with operations $f^{\mathcal{A}} : A^m \to A$ for all $m \geq 0$ and $f \in \Sigma_m$.

Let $\mathcal{A} = (A, \Sigma)$ be an algebra. A unary map $p : A \to A$ is an *elementary translation* of $\mathcal{A}$, if there exist $m > 0$, $f \in \Sigma_m$, $i \in [1, m]$ and $a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_m \in A$ such that for each $a \in A$,

$$p(a) = f^{\mathcal{A}}(a_1, \ldots, a_{i-1}, a, a_{i+1}, \ldots, a_m).$$

The set $\mathrm{Tr}(\mathcal{A})$ of all *translations* of $\mathcal{A}$ consists of the identity map, all elementary translations and all possible compositions of these.

We define a product operation $\cdot : A \times \mathrm{Tr}(\mathcal{A}) \to A$ such that for any $a \in A$ and $p \in \mathrm{Tr}(\mathcal{A})$, $a \cdot p = p(a)$. We also use a similar product on the set of translations, namely if $p, q \in \mathrm{Tr}(\mathcal{A})$, then $p \cdot q$ is the composed translation such that for any $a \in A$, $a \cdot (p \cdot q) = (p \cdot q)(a) = q(p(a))$. These definitions stem from tree language theory and corresponding products between trees and contexts.

It is clear that the products are associative operations (in a generalized sense). This also implies that the translations of an algebra $\mathcal{A}$ form a monoid $\mathrm{Tr}(\mathcal{A}) = (\mathrm{Tr}(\mathcal{A}), \cdot)$. For the sake of brevity, the product signs are sometimes omitted.

Definition 1. An algebra $\mathcal{A} = (A, \Sigma)$ is *monotone* if there exists an order $\leq$ on A such that for every $n \geq 0$, $a_1, \ldots, a_n \in A$ and $f \in \Sigma_n$

$$a_1, \ldots, a_n \leq f(a_1, \ldots, a_n).$$

It is easy to see that an algebra $\mathcal{A} = (A, \Sigma)$ is monotone if and only if there exists an order $\leq$ on A such that $a \leq p(a)$ for every (elementary) translation $p \in \mathrm{Tr}(\mathcal{A})$ and every $a \in A$.

The well-known *Green's $\mathcal{R}$-relation* for semigroups is defined using right ideals of semigroups (cf. for example [4]), but the following characterization of the relation is more useful here.

Lemma 1. *Let S be a semigroup. Then, for every $a, b \in S$, $a \mathcal{R} b$ if and only if there exist $c, d \in S^1$ such that $a = bc$ and $b = ad$.*

A semigroup is called *$\mathcal{R}$-trivial* if $a \mathcal{R} b$ implies $a = b$.

3 The characterization

Before presenting the main theorem, we give a characterization for monotone algebras, introduced in both [1] and [3].

Lemma 2. *An algebra $\mathcal{A}$ is monotone if and only if for every $a \in A$ and all $p, q \in \mathrm{Tr}(\mathcal{A})$,*

$$a \cdot q \cdot p = a \ \textit{ implies } \ a \cdot q = a.$$

Proof. If $\mathcal{A}$ is monotone, then for any $a \in A$ and $p, q \in \mathrm{Tr}(\mathcal{A})$, $a \leq aq \leq aqp$. So, if $aqp = a$, then $a = aq$.

Assume the condition in the lemma and define a relation $\leq_{\mathcal{A}}$ on A such that for every $a, b \in A$, $a \leq_{\mathcal{A}} b$ if and only if $b = p(a)$ for some $p \in \mathrm{Tr}(\mathcal{A})$. It is easy to see, that $\leq_{\mathcal{A}}$ is an order and by its definition it is monotone. □

The proof of the following theorem is also from [3] with small modifications to suit our needs here.

Theorem 3. *A finite algebra A is monotone if and only if $\mathrm{Tr}(\mathcal{A})$ is $\mathcal{R}$-trivial.*

Proof. Assume that $\mathcal{A}$ is monotone, let $u, v \in \mathrm{Tr}(\mathcal{A})$ and assume that $u \mathcal{R} v$. There exist $p, q \in \mathrm{Tr}(\mathcal{A})$ such that $u = vp$ and $v = uq$, so we can write $u = uqp$. Now, for any $a \in A$,

$$au = auqp \ \text{ implies } \ au = auq = av,$$

by the previous lemma, and hence $u = v$, i.e. $\mathrm{Tr}(\mathcal{A})$ is $\mathcal{R}$-trivial.

Let $\mathrm{Tr}(\mathcal{A})$ be $\mathcal{R}$-trivial and suppose that $\mathcal{A}$ is not monotone. Then, for some $a \in A$, $p, q \in \mathrm{Tr}(\mathcal{A})$, $a = aqp$, but $aq \neq a$.

Since $\mathrm{Tr}(\mathcal{A})$ is finite, there exist $k, l \in \mathbb{N}$, $l > k$ such that

$$(qp)^l = (qp)^k.$$

Then, $(qp)^l = (qp)^k q(pq)^{l-k-1}p$ and $(qp)^k q = (qp)^l q$. Thus, $(qp)^l \mathcal{R} (qp)^k q$ and by our assumption $(qp)^l = (qp)^k q$.

Now, especially

$$a = a(qp)^l = a(qp)^k q = aq,$$

which gives a contradiction. Thus, $\mathcal{A}$ is monotone. □

It is easy to see that the previous theorem does not hold for all infinite algebras.

4 Applications to tree language theory

In the following, we consider some applications of Theorem 3 to tree languages. We begin by introducing informally some basic notions of the theory of tree language varieties.

A *generalized variety of finite algebras* (GVFA) is a class **K** of finite algebras which is closed under forming generalized subalgebras, generalized morphic images and finite generalized products of members in **K**. A class of tree languages is a *general variety of tree languages* (GVTL) if it consists of the tree languages recognized by members of some given GVFA. For the original definition of a GVTL and the correspondence theorem, see [7].

It is well-known that the class **R** of finite $\mathcal{R}$-trivial monoids is a variety of finite monoids (cf. [4] for example). Thus, by [7] and Theorem 3, we obtain the following theorem.

Theorem 4. *The class of finite monotone algebras is a GVFA and the corresponding class of monotone tree languages is a GVTL, both definable by the variety of finite $\mathcal{R}$-trivial monoids.*

This is actually the main result of this paper, because apart from the GVTL of aperiodic tree languages [8], there seems to be a lack of natural examples of GVFAs or GVTLs definable by monoids.

As a negative example we show in the following that the general variety of finite and co-finite tree languages is not definable by syntactic monoids.

Example 1. Let $X = \{x, x'\}$ and $\Sigma = \Sigma_1 = \{f\}$. Then the syntactic monoids of both $T = \{f^n(x) \mid n \geq 0\}$ and $T_\Sigma(X)$ are trivial, but $T_\Sigma(X)$ is co-finite, while T is neither finite nor co-finite (as a ΣX-tree language). Thus, the GVTL Nil of finite and co-finite tree languages is not definable by monoids.

The problem of deciding whether a given family of tree languages is definable by a variety of monoids seems to be quite difficult. Already in [7] it is proved that a family definable by a variety of finite monoids has to be a GVTL, and in [5] the families of tree languages definable by VFMs are characterized by certain closure properties. Some of these properties are however quite tricky to check for a given family of tree languages.

Nevertheless, if we can prove that a family of tree languages is definable by monoids, we get directly many good closure properties such as closure under boolean operations, inverse translations and inverse non-deleting and linear tree homomorphisms. The corresponding family of algebras is also closed under some changes of signature such as copying and omitting symbols. These properties translate also into tree languages very naturally, but are a bit lengthy to describe.

We note here also a characterization for $\mathcal{R}$-trivial monoids which we can derive directly from [3] and the general theory. First we need some definitions.

Let S be a semigroup and $s \in S$. An element $r \in S$ is a *divisor* of s, if $s = rt$ or $s = tr$, for some $t \in S$. A subsemigroup $S' \subseteq S$ is *closed under divisors*, if for all $s \in S'$, $r, t \in S$, $s = rt$ implies $r, t \in S'$. A subsemigroup $S' \subseteq S$ is a *right-unit subsemigroup* of S if there exists an $s \in S$ such that $S' = \{r \in S \mid s = sr\}$.

Corollary 5. *Let* $\mathbf{M}$ *denote the class of finite monoids such that all right-unit submonoids of any* $M \in \mathbf{M}$ *are closed under divisors. Then,* $\mathbf{R} = \mathbf{M}$.

Proof. Follows from Theorem 3, the characterization presented in [3] and the fact that every finite monoid is a syntactic monoid for some regular tree language, i.e. the translation monoid of some finite algebra (cf. [6]). □

Most likely, there is also an easy direct proof for the previous corollary.

5 Further remarks

In [1] and [3] monotone string languages are characterized by regular expressions, and a similar characterization would also seem possible for monotone tree languages. However, even more interesting would be to find a good characterization for the variety of finite g-congruences [7] corresponding to the monotone tree languages. For monotone string languages, a congruence characterization is proved in [1].

One promising example of other interesting GVTLs definable by a variety of monoids could be the GVTL of piecewise testable tree languages (ongoing work by the author). Piecewise testability can be generalized quite naturally for trees, and it is already clear that the GVTL of piecewise testable tree languages is included in the GVTL defined by $\mathcal{J}$-trivial monoids, but at the moment the equality of these varieties is still unsolved.

Acknowledgement The author is thankful to Magnus Steinby for his guidance and remarks and to Saeed Salehi for many useful discussions and comments.

References

[1] J. Brzozowski, F. Fich, Languages of R-Trivial Monoids, *Journal of Computer and System Sciences*, **20** (1980), 32-49.

[2] S. Eilenberg, *Automata, Languages and Machines*, Vol. **B**, Academic Press, New York, 1976.

[3] F. Gécseg, B. Imreh, On monotone automata and monotone languages, *Journal of Automata, Languages and Combinatorics*, **7** (2002), 71-82.

[4] J.-E. Pin, *Varieties of formal languages*, Plenum Publishing Corp., New York, 1986.

[5] S. Salehi, Varieties of tree languages definable by syntactic monoids, submitted.

[6] K. Salomaa, *Syntactic monoids of regular forests* (Finnish), M.Sc. Thesis, Department of Mathematics, University of Turku, 1983.

[7] M. Steinby, General varieties of tree languages, *Theoretical Computer Science*, **205** (1998), 1-43.

[8] W. Thomas, Logical Aspects in the Study of Tree Languages, in: *Proceedings of the 9th International Colloquium on Trees in Algebra and Programming, CAAP '84*, Cambridge University Press, 1984, 31-50.

Inexpensive Linear-Optical Implementations of Deutsch's Algorithm

Michael Stay

Deutsch's algorithm was the first algorithm proposed for which quantum computers could outperform classical computers. It requires only a single qubit; since photons make very stable qubits, and expensive materials are only needed for multi-qubit gates, one can implement Deutsch's algorithm using inexpensive, readily available parts. Here we describe two implementations. Such a computer can be useful in demonstrating simple quantum effects.

1 Brief History

Quantum computation is a relatively new field; Richard Feynman [3] proposed the idea that one well-controlled quantum system could simulate another, more inaccessible system. Since then, several algorithms have been published specifically for quantum computers. In 1985, David Deutsch published a design for a universal quantum computer [1], and later published the first algorithm for which a quantum computer could outperform a classical computer [2], of which we present an implementation below. Two years later, Peter Shor invented one of the two algorithms that justify all the money being spent in researching quantum computation: an algorithm to factor large integers in polynomial time [6]; if quantum computers ever reach the point where they can control thousands of quantum bits for millions of steps, then the asymmetric cryptography that protects nearly all of the world's financial transactions will be broken. One year later, Shor gave the first example of a quantum error-correcting code, a necessary tool for achieving that goal. Lov Grover published the next important algorithm [4], an unordered database search. This algorithm has the potential to affect the design of symmetric cryptosystems, since it effectively halves the length of a symmetric key.

2 Implementations

In the latter part of the 1990s and early part of this decade, considerable progress has been made towards the implementation of a quantum computer. In most im-

plementations, like nuclear magnetic resonance (NMR) and ion traps, the quantum sytems are extremely sensitive to the environment and need to be cooled and shielded. Optical quantum computers are one strong possibility for scalable quantum computation, because photons are extremely stable quantum systems—so stable, in fact, that the major hurdle is getting the photons to interact with each other at all! One proposal [7] for making an optical quantum computer involves the use of the optical Kerr effect. Kerr media can be used to form the so-called "controlled-phase rotation" gate.

Since Kerr media is bulky, difficult to work with, and expensive, other ways of doing optical quantum computation are being researched. In [5], Knill, Laflamme, and Milburn describe quantum gates made entirely out of linear optics: cheap, reliable, and easy-to-use. The downside is that the gates are only probabalistic; the output channel decreases in brightness for each gate used. Deutsch's algorithm, however, requires only a single qubit, so in implementing it, we can avoid all the problems with interaction between qubits.

3 Linear-optical quantum gates

The simplest quantum system is the quantum-bit, or *qubit*. We can describe a qubit mathematically by a normalized two-dimensional vector with complex elements:

$$\Psi = \begin{pmatrix} a \\ b \end{pmatrix}$$

$$aa^* + bb^* = 1,$$

where * indicates complex conjugation. Also, there is a global phase factor that is unobservable: we can only measure the phase difference between a and b.

One-bit gates are represented mathematically as 2×2 matrices. The first gate we consider is the polarizer. It implements a projection operator that transmits an impinging photon in the state $\Psi = \binom{a}{b}$ with probability aa^*, or absorbs the photon with probability bb^*. Written as a matrix, the polarizer is

$$P = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Polarized film was invented in the mid 1930's by a scientist named Edwin Land. Before Land's invention, polarizers were made from two clear crystals of calcite, carefully cut and arranged such that, due to the birefringence, oppositely polarized beams of light exited at right angles to each other. They were large and expensive, and people were trying to figure out another way to achieve the same effect. Land had a tradition of reading old science journals with his wife.

They came across the account of a physician, Dr. Willam Herapath, who was researching the effect of the anti-malaria drug quinine on dogs. Herapath noted that microscopic crystals had formed in the dog's urine, and that when parallel to one another, the crystals were transparent. When they crossed at right angles, however, the crystals were dark. Herapath recognized the phenomenon as polarization and was able to grow some slightly larger crystals, but the process was inefficient and unwieldy. Land came up with the idea of embedding the crystals in a plastic sheet and aligning them, originally with a large electromagnet, and later by stretching the film. Land went on to form the very successful Polaroid company, providing these new, inexpensive polarizers.

The long molecules in the crystals act like antennae for optical wavelengths; an electron moving in response to the electric field of a photon will absorb the photon, but will only feel a restoring force perpendicular to the molecule; thus any photon reemitted by the electron will be polarized perpendicularly to the molecule.

Next, mirrors implement phase inversion. A beam of light reflecting off a mirror as aligned in Fig. 1 will have left-right polarization switched, while up-down polarization remains unaffected:

$$F = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Phase inversion is a special case of phase shift. More general phase shift is implemented with cellophane: light polarized along the same axis as the stretch of the cellophane is unaffected, whereas light polarized perpendicularly to the stretch is shifted slightly. The amount of shifting depends on the color of the light, similar to the creation of a rainbow by a prism. The difference in phase is visible as a strong blue color when viewed through a polarizer. The matrix representation of a phase shift θ is

$$S_\theta = \begin{pmatrix} e^{i\theta} & 0 \\ 0 & 1 \end{pmatrix}.$$

Rotation of the cellophane, mirror, or polarizer is represented with the usual rotation matrix:

$$R_\theta = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}.$$

4 Deutsch's Problem

Deutsch's problem was the following: given one of the functions $f(x)$ from the sets $\{0, 1\}$ ("constant" functions) or $\{x, \overline{x}\}$ ("balanced" functions) as a black box,

determine from which set it came by computing f on one input. One bit of information is enough to distinguish the two sets, but classically there's no way of calculating the bit we need, since it's the xor of two function outputs. With a quantum computer, we get qubits for input and output, and we can do better.

5 Implementation

In the mirror-based quantum version, we have four boxes that do the following:

$$0 \to \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad : \text{No reflections.}$$

$$x \to \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \quad : \text{Horizontal reflection.}$$

$$\bar{x} \to \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad : \text{Vertical reflection.}$$

$$1 \to \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \quad : \text{Both horizontal and vertical reflections.}$$

In other words,

$$f(x) \to \begin{pmatrix} (-1)^{f(0)} & 0 \\ 0 & (-1)^{f(1)} \end{pmatrix}.$$

If we choose a horizontally or vertically polarized photon, we won't be able to tell if it was reflected: they're both symmetric horizontally and vertically. These qubits correspond to the classical inputs zero and one, and as before, we gain no information about which function it is. However, if we choose a diagonally polarized photon, then each reflection makes the resulting state perpendicular to the state immediately before:

$$F\Psi = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 \\ 1 \end{pmatrix};$$

$$\Psi \cdot F\Psi = \frac{1}{2}(-1 \cdot 1 + 1 \cdot 1) = 0.$$

Deutsch's algorithm, then, proceeds like this:

1. Turn the input polarizer to 45 degrees.

2. Turn the output polarizer to 45 degrees.

3. Shine light into the box. If we don't see any light coming out, it's a constant function; otherwise it's a balanced function.

Mathematically, the algorithm can be represented as

$$D = (R_{\pi/4}P)(R_{\pi/2}F)^{f(0)}F^{f(1)}(R_{\pi/4}P)$$

See figures 1, 2, and 3.

The cellophane-based implementation is similar:

$$0 \to \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad : \text{No layers of cellophane.}$$

$$x \to \begin{pmatrix} e^{i\theta} & 0 \\ 0 & 1 \end{pmatrix} \quad : \text{One layer, aligned horizontally.}$$

$$\bar{x} \to \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix} \quad : \text{One layer, aligned vertically.}$$

$$1 \to \begin{pmatrix} e^{i\theta} & 0 \\ 0 & e^{i\theta} \end{pmatrix} \quad : \text{Two layers, aligned horizontally and vertically.}$$

Or,

$$f(x) \to \begin{pmatrix} e^{i\theta f(0)} & 0 \\ 0 & e^{i\theta f(1)} \end{pmatrix}.$$

As before, horizontally and vertically polarized photons pass through visibly unaffected; while the phase does get shifted, the shift is global, and therefore unmeasurable. Diagonally polarized photons acquire a phase difference between the components when there's a single layer of cellophane and only blue photons pass through unabsorbed.

In this case, instead of blocking the light completely for balanced functions, it blocks all colors but blue. See figures 4, 5.

Mathematically, the algorithm can be represented as

$$D = (R_{\pi/4}P)(R_{\pi/2}S_\theta)^{f(0)}S_\theta^{f(1)}(R_{\pi/4}P)$$

6 Conclusion

Since photons are such stable qubits, and since Deutsch's algorithm only requires one, a linear-optical implementation of his algorithm is well within even the most restricted budgets and can help to illustrate some of the basic concepts of quantum computation.

References

[1] Deutsch, D., Quantum Theory, the Church-Turing Principle, and the Universal Quantum Computer. *Proc. Roy. Soc. Lond.* **A400**. (1985) pp. 97-117.

[2] Deutsch, David and Richard Jozsa, Rapid solutions of problems by quantum computation. *Proceedings of the Royal Society of London, Series A* **439**. (1992) pp. 553.

[3] Feynman, R. P., Simulating Physics with Computers. *International Journal of Theoretical Physics* **21**. (1982) pp. 467-488.

[4] Grover, L. K., A Fast Quantum Mechanical Algorithm for Database Search. *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing*. (1996) pp. 212-219.

[5] Knill E., Laflamme R. and Milburn G., A scheme for efficient quantum computation with linear optics. *Nature* **409**. (2001) pp. 46-52

[6] Shor, P., Algorithms for quantum computation: discrete logarithms and factoring. *Proceedings 35th Annual Symposium on Foundations of Computer Science*. IEEE Comput. Soc. Press. (1994) pp. 124-134.

[7] Tanas, R., Nonclassical states of light propagating in Kerr media. *Theory of Non-Classical States of Light*, eds. V. Dodonov and V. I. Man'ko. Taylor & Francis, London. (2003) p. 267.

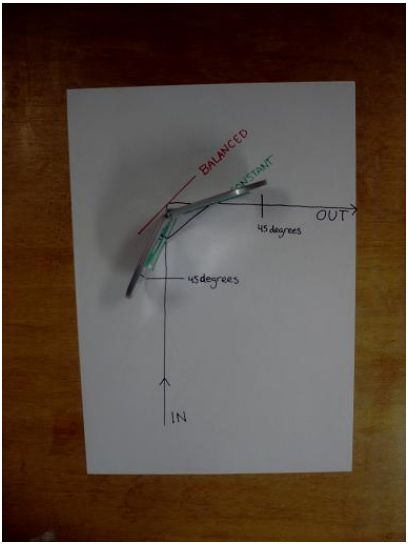


Figure 1: Layout of mirror-based implementation

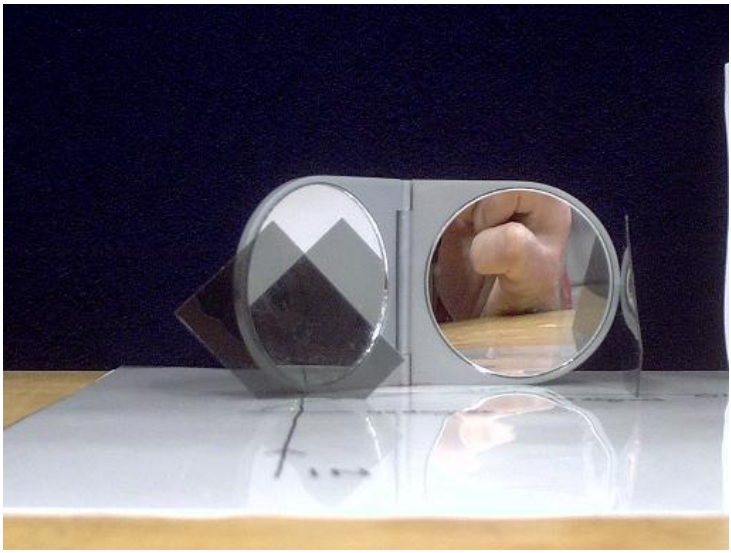


Figure 2: Balanced function

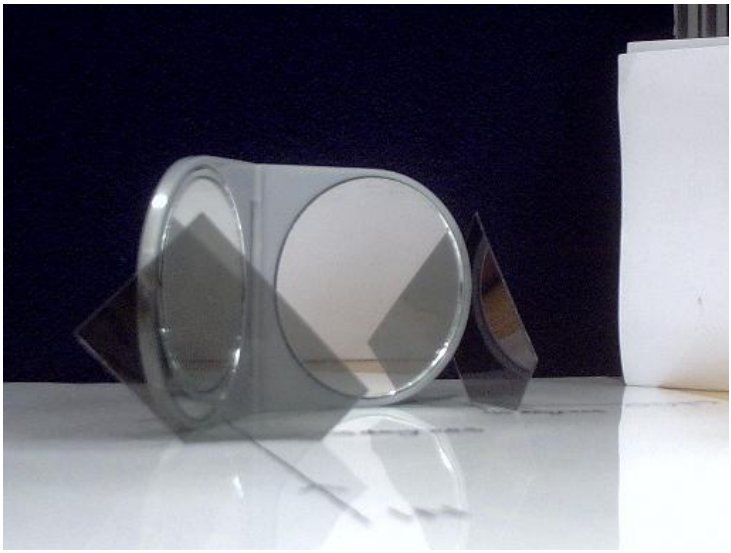


Figure 3: Constant function

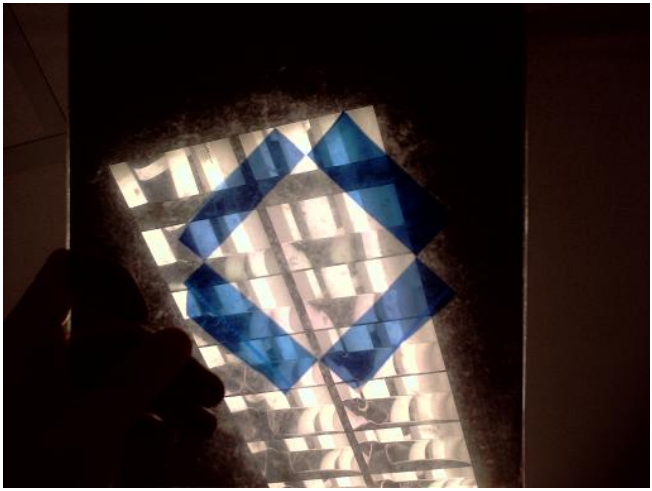


Figure 4: Cellophane-based implementation. There are two layers of cellophane aligned perpendicularly to each other, and at 45 degrees to the polarizers. Where they overlap, both componenents of the phase are shifted by the same amount, giving rise to an unobservable global phase shift

Figure 5: Cellophane aligned with the polarizers, simulating classical inputs; we gain no information

The Puzzle Corner

by

Laurent Rosaz

LRI, Orsay CNRS-Université de Paris Sud
Bât 490, 91405 Orsay France
Laurent.Rosaz@lri.fr

Readers are invited to send comments, and to send exercises, even if they don't know the answer. Write to Laurent.Rosaz@lri.fr.

66 Another chessboard tiling

I tiled an $8 * 8$ chessboard with 32 dominoes. Each domino is either horizontal or vertical. Show that there is an even number of horizontal dominoes.

Does the result generalize to every chessboard with an even size ? If not, to which of them ?

67 To get the multiplication

My logic is the first-order logic on natural integers with the added operators "+" (with an integer result: the usual addition) and "divides" (with a boolean result: a divides b iff $b \mod a = 0$).

Define the multiplication with that logic.

(You are likely to get a "big" formula, so let you know that I have a solution with only two quantifiers, one of them is only to get $\leq$)

Solutions to Previous Puzzles

64 Exponential determinization

Can you find for every integer n a (non-deterministic) automaton A_n on the alphabet $\{a, b\}$ such that the determinization of A_n leads to a minimal automaton B_n with 2^n states ?

Solution:

Let A_n be the following automaton:

The states are $1, ..., n = [\![1..n]\!]$

1 is the initial state

n is the only final state

for every $i \in [\![1..n-1]\!]$, there is an a-arrow from i to $i+1$

for every $i \in [\![1..n]\!]$, there is a b-arrow from i to itself

for every $i \in [\![1..n]\!]$, there is a b-arrow from i to 1

There is no arrows but those previously described.

Let B_n be the automaton obtained by determinizing A_n (and where, of course, only the usefull states are kept).

Note that if Y is a subset of $[\![1..n]\!]$, then, in B_n, Y leads by a b-arrow to $Y \cup \{1\}$, and by an a-arrow to $\{j+1 | j \in Y \wedge j < n\}$. In a nutshell, b adds $\{1\}$ to the state, while a adds 1 to the elements (but to n which disappears).

We first prove that every subset X of $[\![1..n]\!]$ is a subset of B_n. We prove so by induction on the cardinality of X.

$\{1\}$ is the initial state, and every singleton is obtained by inductive use of a-arrows. Moreover, the empty set is obtained by use of the a-arrow on $\{n\}$. Thus the empty set and the singletons are obtained.

Assume every set of cardinality k is obtained, then, by use of the b-arrows, you get every state of cardinality $k+1$ containing 1, and next, by repetitive use of a-arrows, you get all the states of cardinality $k+1$.

Thus every subset appears in B_n which thus gets 2^n states.

We now prove that B_n is minimal

To prove so, consider two different states X and Y of B_n. They are subsets of $[\![1..n]\!]$ and there is an integer i which is contained in one and only one of them, say that it is contained in X but not in Y. To prove that X and Y will not be gathered by minimization, just notice that in B_n, the word a^{n-i} leads X, but not Y, into a final state.

65 Linear algorithms for finding the diameter of a tree graph

A tree is a connected undirected graph without loops. The diameter of a tree is the maximum distance between vertices. One wants a linear algorithm to find the

diameter of a tree.

Find two solutions (one is very classical and you are likely to know it, the other one is much less known (hint: it works with two depth first searches))

Solution:

Here is the classical solution:

On a rooted tree, the point is that the depth plus the underdepth (the depth obtained when you remove the deepest subtree with the corresponding root-son link) gives you the length of the longest path which goes through the root, while the max of the diameter of the sons gives the length of the longest path which does NOT go through the root. You get a linear algorithm, by computing depth, underdepth and diameter in parallel. If we assume the tree to be complete binary for simplicity, this leads to the following code:

```
function diameter(in A: tree): integer
   slave_diameter (A,diam,depth)
   return diam

procedure slave_diameter(in A: tree; out diam,depth: integer)
if A is a leaf
then
  diam <- 0
  depth <- 0
else
  slave_diameter(left son of A, diam_left, depth_left)
  slave_diameter(right son of A, diam_right, depth_right)
  depth = 1 + max (depth_left, depth_right)
  diam = max(diam_left, diam_right, 2+depth_left+depth_right)
```

On a non-binary, unrooted tree (ie a “graph tree”), the principle is the same, but one must deal with the lack of top-down orientation and with the possible various degree of the nodes.

Here is a code (before changing the code, watch out the nodes with 0 or 1 “blue neighbor”).

```
for every vertex v, color[v] <- blue
pick up a node v0 and do DFS(v0,depth,diam)
return diam

DFS(in v, out depth, out diam) :
      color[v] <- red
      depth <- 0
```

```
underdepth <- 0
diam <- 0
for each blue neighbor t of v do
        DFS(t,depth_t,diam_t)
        if   diam <  diam_t
        then diam <- diam_t

        if        depth <  1+ depth_t
        then underdepth <-    depth
                  depth <- 1+ depth_t
        else
          if   underdepth <  1+ depth_t
          then underdepth <- 1+ depth_t
diam <- max(diam, depth+underdepth)
```

Here is the other solution, which is much simpler to implement on "graph-trees" than on rooted trees.

Lemma : Let v_0, v_1, v_2 be three nodes such that v_1 is at maximal distance from v_0 and v_2 is at maximal distance from v_1. Then the distance from v_1 to v_2 is the diameter.

Proof : Let D be the v_1-v_2 distance.

Put a top-down orientation on the tree by calling v_0 the root. Note that v_1 is at maximal depth.

Let x and y be two nodes. We must prove that the x-y distance is less than or equal to D.

If x of y is v_1, then the distance from x to y is less than or equal to D since v_2 is at maximal distance from v_1.

If not, let z be the deepest common ascendant of x and y.

If z is not an ascendant of v_1, or if z is an ascendant of v_1, but not in the x-branch, nor in the y-branch, then the x-y distance is less than or equal to both the x-v_1 and the y-v_1 distances, which are less than or equal to D.

If z is an ascendant of v_1 in the x branch (ie the common ascendant of x and v_1 is a strict descendant of z), then the x-y distance is less than or equal to the v_1-y which is less than or equal to D.

Similar or course if z is an ascendant of v_1 in the y branch.

Due to that lemma, the following algorithm works (and it can obviously be implemented linearly):

```
Pick up a node v0 (whichever one)
Find v1 at maximal distance from v0
Find v2 at maximal distance from v1
return distance(v1,v2)
```

Reports from Conferences

Report on ICALP'2004/LICS'2004

International Colloquium on Automata, Languages, Programming
IEEE Symposium on Logics in Computer Science
July 12 – 17, Turku, Finland

Manfred Kudlek

ICALP'04, the 31^{st} in this series of conferences on Theoretical Computer Science, took place at Turku. It was the first time that ICALP returned to a former place, in this case after 27 years. Seven non-Finnish participants of the conference in 1977 were present. ICALP'04 was held jointly with LICS'04, Annual IEEE Symposium on Logic in Computer Science, the 19^{th} in its series. ICALP'04 broke a number of records : 379 submitted papers from 43 countries, and 352 participants from 38 countries, the highest number so far.

ICALP'04 and LICS'04 were organized respectively by EATCS and IEEE, and by the University of Turku, Faculty of Mathematics and Natural Sciences, Department of Mathematics. The organizing Committee consisted of Vesa Halava, Tero Harju, Lauri Hella (co-chair LICS), Mika Hirvensalo, Timo Järvi (co-chair ICALP), Juhani Karhumäki (chair), Timo Knuutila, Jarkko Kari, Elisa Mikkola, Kalle Saari, Elena Petre, Ion Petre, Petri Salmela, Patrick Sibelius, and Magnus Steinby. In particular, Mika Hirvensalo has to be mentioned when he took over organization from Juhani Karhumäki who had become seriously ill, in the most important and busy time of organizing ICALP and LICS.

ICALP'04, together with workshops, was attended by 352 participants from 38 countries, and LICS'04, together with workshops, by 225 participants from 25 countries, both conferences together with the workshops by 432 participants from 39 countries. Details are given in the tables below, where I, IL, L, WI, WIL, WL, ΣI, ΣIL denote only ICALP, ICALP and LICS, only LICS, with corresponding workshops and the corresponding totals.

The scientific program of ICALP'04 consisted of 7 invited lectures (3 joint with LICS'04) and 97 contributions selected from 379 submitted papers, by far the record so far. The invited lectures were presented in plenary sessions, All contributions were presented, except for that of Raja Jothi, Balaji Raghavachari. Because of visa difficulties the contribution of Key Martin was presented by Prakash Panangaden. The scientific program of LICS'04 consisted of 6 invited lectures (3 joint with ICALP'04), 40 contributions, and 8 short presentations, selected from 168 submissions. There were 11 workshops, 3 associated with ICALP'04 and LICS'04 jointly, 5 only with ICALP'04, and 3 only with LICS'04. They were

C	I	WI	IL	WIL	L	WL	ΣI	ΣIL
AT			2				2	2
AU			1				1	1
BE	1		1	1		1	3	4
BR		1				2	1	3
CA	11	1	8	1	4		21	25
CH	3	3	1				7	7
CL		1					1	1
CN	2						2	2
CY	1						1	1
CZ	6		1				7	7
DE	15	4	15	1	6		36	42
DK	2		1	1			4	4
EG		1					1	1
ES	2		2				4	4
FI	29	4	23	2	1		58	59
FR	18	10	17	2	7	4	47	58
GR	6		1				7	7
HK	1						1	1
HU	1						1	1
IL	7	3					10	10
IR		1					1	1
IS	1						1	1
IT	12		5	3	7	1	20	28
JP	2	1	4	1	2		8	10
LV	1						1	1
MX		1					1	1
NL	1		2		3		3	6
NO	1			1			2	2
NZ					1			1
PL	5	1	1		1		7	8
PT	1		1				2	2
RO			1				1	1
RU	3	5	1	1			10	10
SE	2		4		2	1	6	9

C	I	WI	IL	WIL	L	WL	ΣI	ΣIL
SK	1						1	1
TH				1			1	1
UA		1					1	1
UK	6		15	6	15	9	27	51
US	24	4	13	4	10	3	46	58
Σ	165	42	120	25	59	21	352	432

ALGOSENSORS'04: First International Workshop on Algorithmic Aspects of Wireless Sensor Networks;

DMCS: Discrete Models for Complex Systems;

FCS'04 AND WOLFASI: Foundations of Computer Security and Workshop on Logical Foundations of Adaptive Security;

FL: Formal Languages: Colloquium in Honour of Arto Salomaa;

ITRS'04: Intersection Types and Related Systems;

LCC'2004: International Workshop on Logic and Computational Complexity;

LSB: Logic and Systems Biology;

LRPP: Logics for Resources, Processes, and Programs;

QPL: International Workshop on Quantum Programming Languages;

WACAM: Word Avoidability, Complexity and Morphisms;

WSA: Workshop on Synchronizing Automata.

The following table presents details with I, L standing for ICALP, LICS, respectively, dates in July (D), number of invited lectures and presentations (I+T), and number of participants (P).

W		D	I+T	P
DMCS	I	10	3+10	29
FL	I	11	6	60
LCC	IL	12-13	9	29
FCS, WOLFASI	IL	12-13	1+18	47
QPL	L	12-13	1+10	36
ITRS	L	13	1+9	16
LRPP	IL	13	1+14	29
ALGOSENSORS	I	16	2+15	29
WSA	I	16	10	38
WACAM	I	17	2+8	33
LSB	L	18	6	29

The programs of ICALP'04, LICS'04, and of the workshops can be found at `http://www.math.utu.fi/icalp04`. From the 379 submitted papers for ICALP'04, three were withdrawn (1 after acceptance), and 5 eliminated because of double submission. In the list of accepted papers there are 97, and in the preliminary program of ICALP'2004 web site there were two more in track A.

The distribution by Topics in track A was

Topic	S	A
Security, Data Compression	7	2
Distributed Protocols	21	6
Data Structures	14	1
Approximation	21	4
On-line	4	2
Learning, Streaming, Property Testing	20	6
Parametrized	4	2
Other Algorithms and Data Structures	44	16
Quantum, Cellular	16	4
Natural Computing, Neural, Heuristics	10	1
Automata, Formal Languages	30	3
Game Theory	9	4
Models of Large Networks, Power Laws	8	1
Combinatorics	13	2
Other Complexity	14	8

and that for track B (with overlaps) is

Topic	S	A
Algebraic and Categorial Models	14	5
Applications of Automata in Logic	13	4
Concurrency, Mobility and Distributed Systems	26	4
Databases, Semi-structured Data and Finite Model Theory	10	4
Program Logigs, Formal Methods and Model Checking	30	7
Logigs and their Applications	21	8
Principles of Programming Languages	11	3
Security Analysis and Verification	6	1
Semantics of Programming Languages	18	10
Specification, Refinement and Verification	18	4
Type Systems and Typed Calculi	13	4

The following table shows the distribution by number of authors

A	AS	AA	BS	BA	TS	TA
1	81	13	33	12	114	25
2	105	26	45	9	150	35
3	46	13	21	6	67	19
4	30	11	6	1	36	12
5	7	4	2		9	4
6	2	1			2	1
7			1		1	
	271	68	108	28	379	96

The distribution by countries is given in the following (long) table

C	I	AS		AA		BS		BA		TS		TA	
AT			$\frac{1}{2}$				$\frac{1}{2}$		$\frac{1}{2}$	1			$\frac{1}{2}$
AU		1	$\frac{1}{4}$			1				2	$\frac{1}{4}$		
BE						2	$\frac{11}{30}$			2	$\frac{11}{30}$		
BR		2				1				3			
CA		21	$\frac{47}{60}$	5	$\frac{17}{60}$	3	$\frac{1}{2}$		$\frac{1}{2}$	25	$\frac{17}{60}$	5	$\frac{47}{60}$
CL			$\frac{1}{2}$								$\frac{1}{2}$		
CH	1	2		1						2		1	
CN		1	$\frac{3}{4}$	1		4				6	$\frac{3}{4}$	1	
CY					$\frac{1}{5}$						$\frac{1}{5}$		$\frac{1}{5}$
CZ		4	$\frac{2}{3}$	2	$\frac{1}{2}$	1	$\frac{1}{2}$			6	$\frac{1}{6}$	2	$\frac{1}{2}$
DE	1	32	$\frac{47}{60}$	7	$\frac{11}{20}$	8	$\frac{9}{14}$	5	$\frac{1}{6}$	41	$\frac{179}{420}$	12	$\frac{43}{60}$
DK		2		1		1	$\frac{1}{2}$	1		3	$\frac{1}{2}$	2	
DZ						1				1			
EE		2				1				3			
EG		1								1			
ES		4	$\frac{1}{3}$		$\frac{1}{2}$					4	$\frac{1}{3}$		$\frac{1}{2}$
FI		7								7			
FR	1	29	$\frac{1}{12}$	5	$\frac{5}{12}$	19	$\frac{1}{4}$	4		48	$\frac{1}{3}$	9	$\frac{5}{12}$
GR		2		2	$\frac{2}{3}$					4	$\frac{2}{3}$	2	$\frac{2}{3}$
HK		1	$\frac{1}{2}$		$\frac{3}{4}$		$\frac{1}{4}$			2	$\frac{1}{4}$		$\frac{3}{4}$
HU		2	$\frac{5}{6}$							2	$\frac{5}{6}$		
IE		1								1			
IL		18	$\frac{1}{10}$	4	$\frac{7}{10}$	4	$\frac{1}{105}$			22	$\frac{23}{210}$	4	$\frac{7}{10}$
IN		7	$\frac{5}{12}$			2				9	$\frac{5}{12}$		
IR		5								5			
IS			$\frac{1}{2}$		$\frac{1}{4}$						$\frac{1}{2}$		$\frac{1}{4}$
IT		6	$\frac{1}{6}$	3		11	$\frac{2}{3}$	2	$\frac{1}{2}$	17	$\frac{5}{6}$	5	$\frac{1}{2}$

C	I	AS		AA		BS		BA		TS		TA	
JP		4	$\frac{1}{12}$							4	$\frac{1}{12}$		
KR		1								1			
LB		1								1			
LV		1		1						1		1	
NL			$\frac{1}{2}$			2				2	$\frac{1}{2}$		
NO			$\frac{1}{3}$		$\frac{5}{6}$					1	$\frac{1}{6}$		$\frac{5}{6}$
PL	1	4	$\frac{1}{4}$	2	$\frac{1}{4}$	2		1		6	$\frac{1}{4}$	3	$\frac{1}{4}$
PT		3								3			
RO		3					$\frac{1}{2}$			3	$\frac{1}{2}$		
RU	1	3	$\frac{11}{12}$	1	$\frac{5}{12}$	1	$\frac{1}{2}$			5	$\frac{5}{12}$	1	$\frac{5}{12}$
SE		4	$\frac{1}{2}$		$\frac{2}{3}$	1	$\frac{1}{2}$			5	$\frac{1}{6}$		$\frac{2}{3}$
SG		1								1			
TH						1				1			
TR		6								6			
TW		3								3			
UK		6	$\frac{1}{3}$	2		16	$\frac{1}{6}$	7	$\frac{1}{3}$	21	$\frac{1}{3}$	9	$\frac{1}{3}$
US	2	66	$\frac{43}{60}$	25	$\frac{1}{60}$	19	$\frac{57}{70}$	6		86	$\frac{307}{420}$	31	$\frac{1}{60}$
YU		1					$\frac{1}{3}$			1	$\frac{1}{3}$		
Σ		271		68		108		28		379		96	

ICALP'04 was opened on Monday morning by Juhani Karhumäki, talking on ICALP'77 and the present one, LICS'04, the workshops, the sauna excursion, dinner, and the sponsors. It was followed by Harri Lönnberg, the vice rector of Turku university. He talked on the 775th anniversary of Turku, and the academic tradition (a monastary school in 1249, the first university in 1649 after Tartu and Uppsala, the fire in 1827 destroying a part of the academy, the transfer of the university to Helsinki and its new foundation in 1917, the first university with Finnish language in 1920). Finally he welcomed all of us wishing success and better weather.

The first invited lecture *'Theory and Practice of Some Probabilistic Counting'* by Philippe Flajolet was very good and interesting. He talked on estimation characteristics of large data streams, analysis of algorithms for it, the angel-daemon model, birthday paradoxon, urn weighting by shaking, sampling, approximate counting, cardinality estimators, and *loglog* counting, mentioning also Mahābhārata as an example (8 MB, 10^6 words, 177601 different ones). With *'Grammar Compression, LZ-encodings and String Algorithms with Implicit Input'*, Wojciech Rytter gave a nice second invited lecture, talking on CFG grammars generating single strings, the Lempel-Ziv encoding, and word equations.

Monika Henzinger, research director of GOOGLE, presented a very nice,

clear and interesting third one with *'The Past, Present, and Future of Web Search Engines'*, talking quite fast on strategies of such engines, text only and text link techniques, ranking, and hardware The fourth (joint) invited lecture *'Testing, Optimization, and Games'* by Mihalis Yannakakis was a nice and very interesting survey on the software reliability problem, testing as an optimization problem, as a game, as learning, deterministic and non-deterministic finite state machines, connections to specification, games, learning and combinatorics. Finally *Pythia* as a toolset was mentioned. Alexander Razborov, in the fifth (joint) invited talk *'Feasible Proofs and Computations : Partnership and Fusion'*, presented an excellent overview on multi-interactive proofs (MIP), bounded arithmetic, propositional proof complexity, feasible (un)provability of P≠NP, non-determinism, interactive proofs, probabilistic checkable proofs (PCP), hardness of approximation, automatizabi- lity, and natural proofs. He started with *'extremely introductory. everybody in the audience will know almost everything'*, showing pictures of famous computer scientists, and finished with the statement that mathematics has to meet metamathe- matics and the question *'how feasible is feasible ?'*.

Martin Hofmann, with the sixth one *'What Do Program Logics and Type Systems Have in Common ?'*, gave a good presentation on the theses *'PL on the level of bytecode is the lingua franca for formal correctness, types for high level languages should be translated into PL, types help to find proofs in PL, PL enhances trust in type systems, and PL make type systems undertand each other'*. The seventh (joint) invited lecture *'Self-adjusting Computation'* by Robert Harper (with Umut A. Acar, Guy E. Blelloch) was a good and interesting survey on combination of algorithic and program language techniques, adaptivity, and selective and adaptive memoization. He finished with a movie demonstrating a hull algorithm for 5 points, and with *'question ?'*.

LICS'04 was opened on Tuesday afternoon by Jarmo Hietaranta, the dean of the Faculty of Mathematics and Natural Sciences The fourth invited lecture of LICS'04, *'Bisimulation and Coinduction : From the Origins to Today'*, given by Davide Sangiorgi, was a very good and interesting historical overview (*'many questions left'*) on locality, theoretical computer science, (algebraic foundations, finite automata, ω-regular languages), philosophical logic (modal logic, Kripke structures), set theory, and coinduction, fixed point theorems. He mentioned most of the pioneers in these areas. Igor Walukiewicz, with *'A Landscape with Games in the Background'*, presented a very good and interesting fifth one on two-player games and winning strategies for them, like regular and non-regular conditions, as well as probalistic and multi-player games, and their application to verification and synthesis. The sixth one, *'Information is Physical, but Physics is Logical!'* by Samson Abramsky was an excellent overview on the history of quantum information and computation (in the spirit of LICS), qubit, entanglement, measurement,

teleportation, quantum information flow, mathematical foundations of quantum mechanics, its computer science perspective, and the new picture of quantum mechanics, forulated with help of strongly compact closed category with biproducts, bipartite projectors, and compositionality. He finished with *'It's logic! Thus Physics is Logical!'*. Samson Abramsky had arrived just a short time before his talk, due to his election as a *Fellow of Royal Society*.

Only a few personal remarks on other contributions can be given here. A very good and interesting presentation was given by Ryan Williams on optimal constraint satisfaction algorithm (the best student paper in track A). Good and nice presentations were given by Michal Kunc on regular solutions of language inequalities, by Markus Lohrey on compressed word problem, and by Robert Dąmbrowski on word equations with two variables. S. Cenk Sahinalp had problems to start his laptop (*Aaah! Kryptonite, malicious script detected)*. Good and interesting talks were presented by Wolfgang Merkle on properties and existence of universal self-delimiting Turing machines, by Véronique Cortier, with nice finger pointer technique, on security protocol under equational theories, by Nicole Schweikardt on monadic least fixed point logic, and by Anca Muscholl, starting *'an e-donkey user'*, on counting in trees.

An excellent presentation on a language for separating deterministic and nondeterministic tree-walking automata, the best paper in track B, was given by Thomas Colcombet and Mikołaj Bojanczyk. Nice and interesting talks were also given by Thorsten Altenkirch, changing the title to *'Inductive Types for Free'* and having conversation with the audience, and by Esfandiar Haghverdi on a categorial model for interaction geometry, saying among others *'It is not important to understand anything of this slide', 'thus I'll tell you the theorem in one word only'*, and *'no difficult questions, easy ones !'*. A very good and interesting presentation gave Mehdi Mhalla, with the best paper in track A, on quantum query complexity. Gatis Midrijānis on quantum queries, Shengyu Zhang on Ambainis' lower bounds, and Emmanuel Jeandel on universality in quantum computing, also gave nice and interesting talks.

Good presentations were given by Marko Samer on LTL queries, by Mariëlle Stoelinga, showing nice pictures, on linear and branching metrics for quantitative transition systems, by Roberto Grossi on a general technique for managing strings, starting with *'At the end of the talk you will have a theorem'*, and by J. Ian Munro, having problems to start his laptop and a dynamic audience, on representation of functions (*'it is 4h: lets go on into the talk', 'a function is just a hairy permutation'*). Good talks also gave Dirk Pattison on initial problems in domain theory, Stephan Kreutzer on inflationary fixed points, and Ran Raz on a lower bound for satisfiability. A very good and interesting presentation was given by Olivier Serre, with the best student paper in track B, on games with winning

conditions with high Borel complexity.

A very good paper, the best student paper in LICS'04, was presented by Felix Klaedtke on a triple exponential lower bound for the size of deterministic automata for Presburger arithmetic.

Wednesday afternoon was reserved for a number of special events. The first one was introduced by Juhani Karhumäki explaining the decision for the 2004 Gödel Prize. From 11 nominations 2 papers have been selected, *'The Topological Structure of Asynchronous Computability'* (JACM 46(6), pp 838-923, 1999) by Maurice Herlihy, Nir Shavit, and *'Wait-free k-set Agreement is Impossible: the Topology of Public Knowledge'* (SIAM JC 29(5), pp 1449-1483, 2000) by Michael Saks, Fotios Zaharoglu. Except the third one all were present at Turku. Giorgio Ausiello informed us on the scientific life of the authors, and presented the award to the authors. After that Maurice Herlihy gave an excellent and interesting presentation with *'Topology and Distributed Computing'*, starting with an article in NYT, May 8, 2001, on the future of computing (*'what to do with theory?'*), the agreement problem with an example from air traffic, citing Alan Turing *'time is nature's way of making sure that anything doesn't happen all at once'*, and finishing with a crash course on combinatorial topology.

The next event was the presentation of the EATCS Distinguished Achievement Award to Arto Salomaa for his outstanding scientific life for 40 years. The introduction was given by Mogens Nielsen and Jan van Leeuwen on Arto Salomaa's scientific life which started in 1964 on Moore automata, and his broad research in automata theory, formal languages, biology, cryptography, his publishing activity (more than 25 books), his 25 PhD students, his activity in academy, and activity for conferences (12 times in PC for ICALP). After that Arto Salomaa talked about his view of his scientific life, as reported elsewhere in this volume.

The third event was the presentation of the best student paper for LICS'04 by Andrei Voronkov to Felix Klaedtke for his paper *'On the Automata Size for Presburger Arithmetic'*. Then Josep Díaz presented the awards of ICALP'04. The best paper award for track A was given to Christoph Dürr, Mark Helligman, Peter Høyer, Mehdi Mhalla for *'Quantum Query Complexity of some Graph Problems'*. that for track B to Mikolaj Bojanczyk, Thomas Colcombet for *'Tree-walking Automata Cannot be Determinized'*. The best student paper award for track A received Ryan Williams for *'A New Algorithm for Optimal Constraint Satisfaction and Its Implications'*, and for track B Olivier Serre for *'Games with Winning Conditions of High Borel Complexity'*.

Finally, Andrei Voronov gave a short speech in honour of Harald Ganzinger who had passed away on June 3, 2004 at the age of only 54 years. He talked on the scientific work of Harald Ganzinger in modern theorem proving systems, depth of his results,, implementation techniques, and promotion of automated reasoning, and the donation of the Herbrand award for 2004 to him, finishing with the words

of Immanuel Kant *'Je mehr Du gedacht, Du getan hast, desto länger hast Du gelebt'*. After this we stood up for one minute for commemoration.

The ICALP proceedings have been edited by Josep Díaz, Juhani Karhumäki, Arto Lepistö, Donald Sannella, and been published as Springer LNCS 3142. With 1255+XIX pages it is the biggest ICALP volume so far. The LICS'04 proceedings, dedicated to the memory of Harald Ganzinger have been edited by Harald Ganzinger and published by IEEE.

The social program started on Sunday evening with a welcome reception. On Monday evening, starting at 19.30, we had a City Reception at Turun Vapaaehtoinen Palokunta (VPK) Sosiaalikeskus (Social Centre of Turku Fire Brigade). Olli Manni, the Deputy of the Lord Mayor of Turku welcomed us, talking about Turku (built also on 7 hills) and its history, and the university with 30000 students. During the reception the Fire Brigade Orchestra played music. It was well after 21 when this event finished. On Tuesday evening the next reception was given by Springer Verlag in Mauno Koivisto centre for the 20^{th} anniversary of Springer EATS Monographs and Texts Series. Following that was the EATCS General Assembly. A report on it is given in this issue. Juhani Karhumäki gave presents of Finnish glass to all participants of ICALP'77 not from Finland and present at ICALP'04 (Giorgio Ausiello, Burkhard Monien, Maurice Nivat, Azaria Paz, Grzegorz Rozenberg, Jan van Leeuwen, and Manfred Kudlek). The author of this report distributed a golden EATCS button to Kurt Mehlhorn (who could not come to Turku) for more than 10 full contributions to ICALP, and silver buttons to the editors of the ICALP proceedings. The present state of contributors is given in the table next page.

The next event was an excursion on Wednesday late afternoon (17.30) to Harjattula, about 15 km southwest from Turku. There we visited and enjoyed a sauna, some of us also swimming in the nearby inlet of Baltic Sea, with temperature about 18°. A barbecue was offered too, with sausages, steaks, chicken, rosted maiz, salad, and bread. On Thursday evening, starting at 19.30, we had the joint Conference Dinner in Turun Linna (Castle of Turku). After entering the castle Juhani Karhumäki gave a welcome speech. Entering the festivity hall we had a temporal transfer into the year 1563, and we were part of a party given by His Highness Duke John (Trygve Forssell) and Her Highness Katarina Jagellonica (Tuufi Stuns). The group Mats Lillhannus (trumpet, recorders), Anna Edgren (recorders), and Pekka Railamaa (Renaissance guitar) played medieval music during the banquet. The court etiquette obliged us to recite each time, announced by a drum signal by the music group, *'Vivant Johannis et Katarina'*, when His Highness adressed us, his guests. In 1563 there were no forks in Northern Europe, only introduced from Poland. Thus we had to eat without such devices. To wash our fingers bowls with rose water were put on the tables. But later this new

Jean-Eric Pin	$10\frac{1}{2}$	Kurt Mehlhorn	$10\frac{1}{12}$
Juhani Karhumäki	$8\frac{47}{60}$	Matthew Hennessy	$5\frac{1}{2}$
Zvi Galil	8	Juris Hartmanis	$5\frac{1}{3}$
Amir Pnueli	$7\frac{5}{6}$	Ronald Book	$5\frac{1}{4}$
Philippe Flajolet	$7\frac{1}{4}$	Christian Choffrut	5
Paul Vitányi	$6\frac{11}{12}$	Michael Rabin	5
Claus-Peter Schnorr	$6\frac{1}{2}$	Zohar Manna	5
Torben Hagerup	$6\frac{1}{2}$	Arnold Schönhage	5
Karel Čulik II	6	Dominique Perrin	$4\frac{5}{6}$
Géraud Sénizergues	6	Moti Yung	$4\frac{2}{3}$
John Reif	$5\frac{2}{3}$	Burkhard Monien	$4\frac{2}{5}$
Walter Vogler	$5\frac{1}{2}$	Christophe Reutenauer	4
Joost Engelfriet	$5\frac{1}{2}$	Marcel Paul Schützenberger	4
Grzegorz Rozenberg	$5\frac{1}{2}$	Volker Diekert	4

devices were also introduced to Finland, and we changed from fingers to forks. His Highness invited us to his banquet mentioning that at his time 34l of beer were consumed every day. He also explained us the situation in Europe in 1563, mentioning the countries (some not known at that time) of the participants, in particular Denmark, the enemy of Sweden in his time. His and Her Highness also enjoyed us with medieval songs, accompanied by the music group. We had to sing *'Gaudeamus igitur, vivat academia, vivant profesores, ergo bibamus'*. After His Highness had left at 23h, Juhani Karhumäki thanked all participants, speakers, contributors, and the program and organizing committees, in particular Tina Ahonen and Elisa Mikkola, as well as Arto Lepistö and Mika Hirvensalo for organization, and the sponsors.

ICALP'04 and LICS'04 wwere very successful symposia again, of high scientific level, very well organized, and in a nice relaxed Finnish atmosphere. Thanks to the organizers, in particular to Juhani Karhumäki and Mika Hirvensalo. Next ICALP will be held at **Lisboa**, for the first time in Portugal, on July 11–15, 2005.

Näkemiin Turusta and **Bem-vindo a Lisboa**.

Pictures from ICALP 2004 (by M. Kudlek)

Arto Salomaa

Maurice Nivat

Magnus Steinby

Keijo Virtanen

Juha Sarkio

Timo Järvi

Petri Salmela

Philippe Flajolet

Michal Kunc

Markus Lohrey

Robert Dambrowski

Wojciech Rytter

Emmanuel Hainry

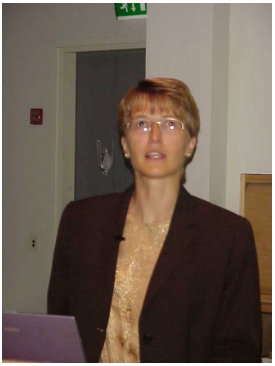
Klaus Meer

Rafail Ostrovsky

S. Cenk Sahinalp

Wolfgang Merkle

Jim Laird



Monika Henzinger

Véronique Cortier

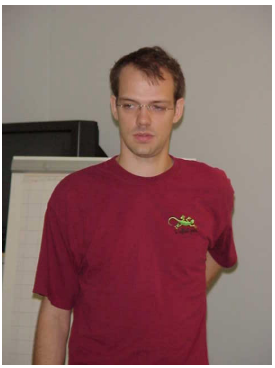
Anca Muschol

Nicole Schweikardt

Claudia Faggian

Mariëlle Stoelinga

Anna Friday



Thomas Colcombet

Thorsten Altenkirch

Grigore Rosu

Pierre Boudes

Esfandiar Haghverdi

Maxim Ushakov

Wilfried Brauer

Luis Monteiro

Michele Bugliesi

Mihalis Yannakakis

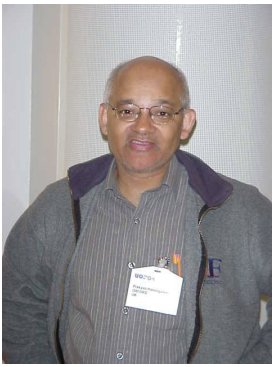
Gatis Midrijanis

Emmanuel Jeandel

Maurice Herlihy

Andrei Voronkov

Alexander Razborov



Prakash Panangaden

Marko Samer

Patrice Godefroid

Shin-ya Katsumata

Martin Hofmann

Roberto Grossi

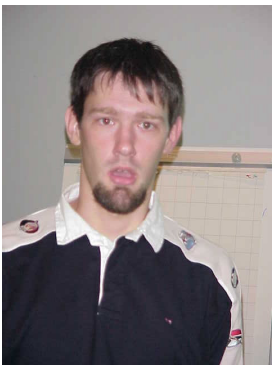
J. Ian Munro

Bin Fu

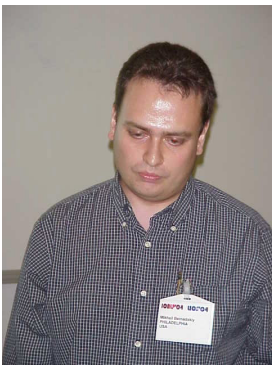
Robert Harper

Dirk Pattison

Stephan Kreutzer

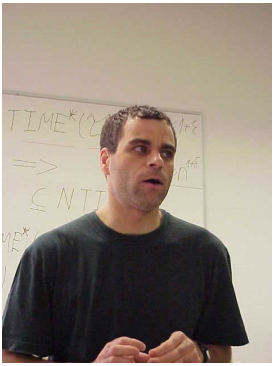


Olivier Serre



Mikhail Bernadskiy

Davide Sangiorgi



Ran Raz

Eran Ofek

Spyros Kontogiannis

George Cristodolou

Martin Gairing

Igor Walukiewicz

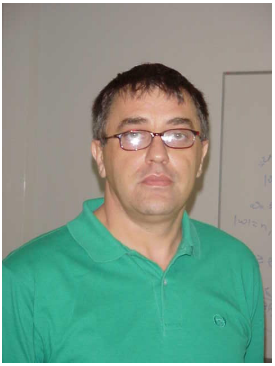
Gwénaël Richomme

Jeffrey Shallit

Peter Leupold

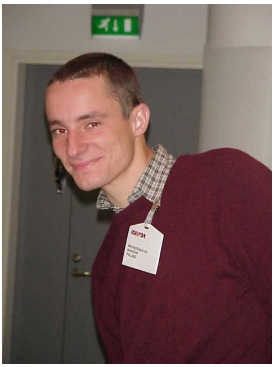
Narad Rampersad

Ali Aberkane



Stefano Varrichio

Samson Abramsky



Mikolaj Bojanczyk

Tomi Kärki

Emmanuel Hyon

Vesa Halava

Jamey Gabbay

Azaria Paz

Juhani Karhumäki

Shengyu Zhang

Awards and Prizes

M. Nielsen, A. Salomaa,
J. van Leeuwen

Jan van Leeuwen

M. Herlihy,
F. Zaharoglu, N. Shavit

Mehdi Mhella

Felix Klaedtke

Don Sannella

Olivier Serre

M. Mhalla,
C. Dürr, P. Hoyer

T. Colcombet,
M. Bojanczyk

REPORT ON ACSD 2004

Intl Conference on Application of Concurrency to System Design 16–18 June, Hamilton, Canada

Laure Petrucci

The annual international conference on application of concurrency to system design took place in Hamilton, Canada. It was organised by the SQRL (Software Quality Research Laboratory) of the McMaster University. The organising group was chaired by Alan Wassyng with efficient help from Doris Burns.

On Tuesday, a satellite workshop (MOdel-based Methodologies for Pervasive and Embedded Software) took place. It was organized by João M. Fernandes, Johan lilius, Ricardo J. Machado and Ivan Porres. Nine selected papers were presented.

On Wednesday, the three-day conference was officially opened. The conference was attended by 60 participants from 13 different countries:

Canada	24	Denmark	1	Finland	5	France	8
Germany	2	Italy	1	Japan	1	The Netherlands	2
Portugal	1	Spain	3	Sweden	1	UK	8
USA	3						

The scientific programme consisted of 3 invited lectures given by Ed Brinksma, Gregor von Bochmann and John Thistle and 21 selected papers. The programme committee, chaired by Philippe Darondeau and Mike Kishinevsky, selected these 21 papers out of the 61 submitted. The distribution by country of authors for accepted papers is as follows:

Country	Acc.	Country	Acc.	Country	Acc.
Australia	0.5	Belgium	0.33	Canada	2
Finland	4	France	4.5	Iran	2
Spain	1.33	Sweden	0.33	UK	3.66
US	2.33				

The social events included a wine and cheese party on Wednesday, an excursion and the conference dinner on Thursday. The excursion to the impressive Canadian Niagara falls took place on thursday afternoon, followed by a visit of the winery hosting the banquet.

Finally, the conference was concluded on friday afternoon with the closing session. The next edition of ACSD will take place in France in June 2005.

Report on CCA'04

International Workshop on Computability and Complexity in Analysis

Martin Ziegler

The last ten years of regular scientific meetings on Computability and Complexity in Analysis (CCA, see `http://cca-net.de/events.html`) have revealed an increasing interest in this synthesis of classical (that is discrete) recursion and complexity theory with continuous mathematics. On a sound logical basis, it reflects and takes into consideration that practical computational problems from physics or engineering commonly include not only booleans or integers but also real numbers, handled by means of rational approximations. The investigation of such problems with respect to computability and complexity in general involves theoretical computer science as well as domain theory, logic, constructive mathematics, computer arithmetic, numerics, analysis, etc.

After last year's CCA conference in Cincinnati (USA), this year's has taken place from August 16 to 20 in Lutherstadt Wittenberg (Germany), UNESCO world heritage and the very town where Martin Luther in 1517 declared his famous 95 theses. Particularly well organized by computer scientists from the University of Halle-Wittenberg, the conference schedule included arrangements for a guided tour through the historic city center with closing barbecue as well as for a cinema presentation of the recent movie "*Luther*".

Scientifically speaking, CCA'04 attracted 40 participants from five different continents; e.g., from South Africa, Siberia, New Zealand, Japan, USA, Canada. Out of their submissions the program committee has accepted 19 for presentation and, after further refereeing, considered for publication in Elsevier's ENTCS series. These contributions include for instance works on effectivized classical mathematical theorems by strengthening existence claims (like of fundamental solutions for linear partial differential equations with constant coefficients) to computability. We also heard interesting complexity results (such as on motion planning or Julia Sets) which, in contrast to algebraic models of computation, realistically take into account the running time's dependence on the desired output precision.

The workshop succeeded very well in his aim to bring together people interested in computability and complexity aspects of analysis and to explore connections with numerical methods, physics, computer science, and logic. Regarding the strong Japanese attendance this year, it has been suggested to hold CCA'05 in Kyoto.

REPORT ON CIAA 2004

9th International Conference on Implementation and Application of Automata Kingston, Ontario, Canada, July 22–24, 2004

M. Domaratzki, A. Okhotin, K. Salomaa

The past academic year brought unexpected increase in the volume of new work in theoretical computer science. A mighty flow of papers swept through ICALP and MFCS, breaking all the records on the number of submissions. We at CIAA, having expected to receive about 40 submissions, got 62 in total, and the Program Committee was faced with an uneasy task of selecting papers to fit in the short timeframe of the conference.

Due to the fierce competition many good papers could be accepted only as posters. The scientific program consisted of 2 invited talks by Oscar Ibarra and by Jeffrey Shallit, 25 regular papers and 14 poster papers. The program of the conference is posted at `http://www.cs.queensu.ca/ciaa2004`. The statistics on the distribution of submissions and authors by country is given in the following table, where the columns contain the number of (I)nvited papers, (S)ubmissions, (A)ccepted full papers and (P)osters accepted.

Country	I	S	A	P	Country	I	S	A	P
Austria		1	1		Japan		2		
Brazil		$1\frac{2}{5}$			Jordan		1		
Canada	1	$9\frac{2}{3}$	$6\frac{2}{3}$	$1\frac{1}{2}$	Korea		1		
China		1			Netherlands		$4\frac{1}{4}$	$1\frac{1}{4}$	1
Czech Republic		$4\frac{1}{3}$	$\frac{1}{3}$	2	Portugal		1		1
Finland		3	1	1	Russia		1	1	
France		$10\frac{3}{5}$	4	2	Slovakia		1	1	
Germany		$5\frac{1}{2}$	$4\frac{1}{2}$	1	South Africa		1		1
Hong Kong		2	2		Spain		2	1	1
Hungary		1			UK		2		1
India		2			USA	1	$3\frac{1}{4}$	$1\frac{1}{4}$	$1\frac{1}{2}$
Israel		1			*total*	2	62	25	14

Every full paper and most of the posters were presented at the conference by one of the authors. The papers and the abstracts of the posters were included in the pre-proceedings that was distributed during the conference. The revised versions of the papers will be published in an LNCS proceedings volume. CIAA'04 was attended by 65 participants from 14 countries and regions; their distribution by country is given in the next table.

Austria	1	France	10	Netherlands	3	UK	1
Canada	25	Germany	6	Russia	1	USA	7
Czech Republic	2	Hong Kong	3	Slovakia	1		
Finland	2	Italy	1	Spain	2		


Oscar Ibarra

The first day of the conference, Thursday, July 22, started with a materialization of every organizer's fears: the conference room's data projector, which we have thoroughly tested a couple of days before, refused to operate in front of the audience. Our impatient attempts to resuscitate it were in vain, and a couple of minutes before the conference was supposed to be opened, the CIAA registration desk staff were amazed to see two thirds of the organizing committee frantically running out of the conference room! Having grabbed a backup projector from our department's main office upstairs, we managed to install it almost in time.

In the first session, Oscar Ibarra gave an excellent invited lecture titled "Automata-Theoretic Techniques for Analyzing Infinite-State Systems" that provided fresh insights into work on counter automata as well as into applications of automaton models ranging from verification to molecular computing. After a coffee break, the Thursday morning's scientific programme was continued with the talks by Bryan Krawetz on the state complexity of DFAs, by Fabien Coulon on NFA to DFA conversion, and by Olivier Carton on Hopcroft's DFA minimization algorithm.

After a buffet lunch in Leonard Hall cafeteria there was an afternoon session, which included talks by Petr Sosík on a new language operation – substitution on trajectories, by Yonghua Han on a sequence matching algorithm, by Rudolf Freund on catalytic P systems, by Mathieu Giraud on weighted finite automata, by German Tischler on figure drawing using automata and by Wojceich Fraczak on a new class of automata called concatenation state machines.

The first event of the social program of the conference was held in the evening. A cruise over Lake Ontario on a triple deck boat "Island Queen", took conference participants and guests along the Kingston waterfront and through western parts of the world-renowned "Thousand Islands" in the mouth of St. Lawrence river.

The weather forecast promised a rain storm during the cruise, which fortunately did not occur. The weather was hazy, and the more adventurous participants spent their time on the upper deck.

The second day of the conference, Friday, started with an invited talk given by Jeffrey Shallit, titled "Regular Expressions: Enumeration and State Complexity". There a novel use of the Chomsky Schutzenberger theorem for context-free languages was presented in order to obtain enumerations of regular languages recognized by regular expressions of a given length.


Jeffrey Shallit

The morning session was continued with the talks by Ivan Zakharyaschev on automaton-based equivalence testing for a model of programs, by Cyril Allauzen on a library for handling weighted automata, by Bruce Watson on a graphical software for manipulating automata, and by Alfons Geser on string rewriting systems. After the lunch we had talks by Anssi Yli-Jyrä on dependency parsing in natural language processing, by Manuel Vilares on error repair for spelling correction, and by Björn Borchardt on tree automata for programming language processing.

The CIAA general meeting chaired by Sheng Yu took place on Friday afternoon. The location of CIAA 2005 was announced to be Sophia Antipolis (Nice, France). Tentative plans for locations of CIAA conferences in the following years were discussed. During the general meeting the CIAA 2004 best paper award was presented to Lila Kari, Stavros Konstantinidis and Petr Sosík for their paper "Substitutions, Trajectories and Noisy Channels". The award is sponsored by the University of California at Santa Barbara. The poster session was held right after the general meeting. The conference banquet took place in Ban Righ Dining hall on Queen's campus on Friday evening.


Sheng Yu

The last day of the conference, Saturday, started with a discussion session on XML representation for finite automata. It was followed by the morning session that included talks by Yo-Sub Han on expression automata – a generalization of NFAs, by Galina Jirásková on the state

complexity of NFAs, by Florent Nicart on improved conversion of regular expressions to automata, by Sebastian John on minimization of ε-NFAs, and by Martin Kutrib on undecidable problems for restricted classes of context-free grammars.

Traditionally the CIAA conferences have lasted for two days and a half. Due to the larger number of submissions and accepted papers this time the program continued into Saturday afternoon. After the lunch, there were talks by Baozhen Shan on the use of graph grammars in biochemistry, by Zeshan Peng on sequence alignment algorithms, by Mark Daley on natural data compression found in the genome of the viruses, and by Harumichi Nishimura on quantum finite automata. The conference was closed at 3pm, and many of the CIAA participants, including the whole of the organizing committee, proceeded to London, Ontario (about 500 km west of Kingston) to attend to the 6th Workshop on Descriptional Complexity of Formal Systems (DCFS 2004) organized by Lucian Ilie and Detlef Wotschke.

CIAA 2004 was the ninth in the line of the Conferences on Implementation and Application of Automata, held annually since 1996. Started as a small workshop, the conference series has become well-established, with a growing number of submissions and participants, in accord with the *renaissance of automata theory* optimistically predicted by Sheng Yu in 2000. It has been a great pleasure for us to host CIAA this year and to welcome this renaissance within the walls of our university. We had three good days of automata theory in Kingston – see you at CIAA 2005 in Nice!

Report on FL'04

Formal Languages – Colloquium in Honour of Arto Salomaa
11 July 2004, Turku, Finland

Manfred Kudlek

The workshop Formal Languages, Colloquium in Honour of Arto Salomaa, FL'04, was held at Turku on July 11 2004, to celebrate his 70^{th} birthday (on June 6, 2004). It took place in Mauno Koivisto-Keskus (Mauno Koivisto Centre). It was organized by Tero Harju, Juhani Karhumäki, Elisa Mikkola, and Magnus Steinby, and attended by 60 participants from 16 countries, shown in the following table

AT	1	CN	1	FR	6	RO	1
BE	1	CZ	2	IL	1	SK	1
CA	4	DE	9	IT	3	UK	1
CL	1	FI	26	NL	1	US	1

FL'04 was opened by Magnus Steinby, who talked on the activities of Arto Salomaa in Turku, in science and for EATCS, Keijo Virtanen, the Rector of the University of Turku, who talked on the research areas of Arto Salomaa in mathematics and computer science, also wishing better weather and the best for the conference and Arto Salomaa, and Juha Sarkio, Director of administration of the Academy of Finland, who talked on the former Finnish president Mauno Koivisto, and the scientific contribution of Arto Salomaa as teacher, writer, and administrator, as well as on his family (*mathematical theory is beautiful and harmonious as a Beethoven symphony*).

The scientific program consisted of 6 presentations. Juhani Karhumäki on behalf of the Department of Mathematics, University of Turku, mentioned Arto Salomaa's influence on the department, university, Academy, as well as his internationality, always looking for new things, and that his retirement is not visible in science. He also mentioned that Arto Salomaa was the first to visit him when he himself was ill. Timo Järvi, for MATINE and TUCS talked on TUCS and Arto Salomaa's work for it as well as on MATINE (Maanpuolustuksen Tieteellinen Neuvottelukunta, Scientific Advisory Board for Defence) and relations between university, industry and defence administration. Finally he presented a panel of MATINE to Arto Salomaa.

Grzegorz Rozenberg, broke privacy, and together with Ingeborg Mayer, Hans Wössner as representatives of Springer-Verlag, presented LNCS volume 3113 '**Theory is Forever**', edited by Juhani Karhumäki, Hermann Maurer, Gheorghe

PĂUN, and himself. It contains 24 essays dedicated to the 70^{th} birthday of ARTO SALOMAA. All congratulated ARTO SALOMAA.

Following these, ARTO SALOMAA thanked all speakers, the participants, and spoke on his life at Turku University, from student to rector, Academy, TUCS, defence (cryptography), and cooperation with Springer, as well as on ARTHUR SCHOPENHAUER's (his name patron) words on age. Finally he thanked especially JUHANI KARHUMÄKI.

After a coffee break the scientific program started. In the first talk GRZEGORZ ROZENBERG talked about *TBA*, a topic he never had worked on (*TBA = Tarzan's Big Achievements* or *Tarzan's and Bolganis Adventures*). In the first part he presented a lot of historical pictures showing first meeting of himself and ARTO SALOMAA in 1971, ICALP's, family, scientific family, and scientific sauna effect (SSE). In the second part (*Theoretical Basis of Alchemy* he talked about forbidding and enforcing, selective competition, the *molecular landscapes*, illustrated with nice pictures, on *mennology* (mental experiments in biology), and on basic units for interactions. He had two endings : first the picture of DADARA for the cover of LNCS 3113, and the second *TBA = Thanks for the Benevolent Attention*.

MAURICE NIVAT, remembering the meeting with ARTO SALOMAA in 1971 in Ontario, using a white board, presented with *'Matrices with Integer Coefficients'* interesting problems (questions to ARTO SALOMAA) on $\{0, 1\}$-matrices with given number of 1's in each row and column, and the complexity of the solution, problems with convex polynomies, their counting, and Sturmian sequences, their factors, and their generalization to two dimensions.

WERNER KUICH, also congratulating, gave with *'An Algebraic Generalization of ω-Regular and ω-Context-free Languages'* a nice and interesting talk on ω-languages WOLFGANG THOMAS, also appreciating ARTO SALOMAA, gave with *'Finite Automata and Algorithmics on Infinite Graphs'* a nice historical survey on relations between structures and logic, single relational structures and classical model theory, decidability and axiomatizability, and (M)SO fragments and FOL. GHEORGHE PĂUN presented with *'Membrane Computing. Back to Turku after Six Years'* a nice introduction and historical survey on membrane computing, and the reasons for research in that area (*Biology needs models!*). He also mentioned the responsibility of ARTO SALOMAA and SOLOMON MARCUS for his own book on matrix grammars. With *'State Complexity: Recent Results and Open Problems'* SHENG YU presented an interesting overview on the state complexity for various operations on regular languages.

In the breaks coffee, tea, and cakes were offered. Lunch was in the restaurant of MAUNO KOIVISTO centre. After closing the colloquium we had a reception (combined with registration and welcome for ICALP'04). Thus this colloquium was a very nice and successful event to celebrate ARTO SALOMAA's birthday, held in a familiar atmosphere. We are looking for the next one in 2009.

Report on Grammar Systems Week 2004

July 5 – 9, 2004, Budapest, Hungary

Rudolf Freund

With the Grammar Systems Week 2004, after 8 years the workshop on grammar systems returned to its roots: in 1996, the workshop Grammar Systems had already been organized by Erzsébet Csuhaj-Varjú in Budapest, Hungary; in 1998, the MFCS'98 Satellite Workshop on Grammar Systems was realized by Alica Kelemenová in Brno, Czech Republic; the International Workshop Grammar Systems 2000 was carried through by Rudolf Freund in Bad Ischl, Austria.

The Grammar Systems Week 2004 in Budapest, Hungary, now was organized by the Theoretical Computer Science Research Group at the Computer and Automation Research Institute of the Hungarian Academy of Sciences (SZTAKI) in the frame of the project EU Centre of Excellence in Information Technology, Computer Science, and Control, contract no. ICA1-CT-2000-70025, HUN-TING, Workpackage 5 as well as under the auspices of the European Molecular Computing Consortium (EMCC) and the IFIP Working Group 1.2 on Descriptional Complexity.

The steering committee of the grammar systems workshops consists of the authors of the monograph 'Grammar Systems: A Grammatical Approach to Distribution and Cooperation' (Gordon and Breach, London, 1994) Erzsébet Csuhaj-Varjú, Jürgen Dassow, Jozef Kelemen, and Gheorghe Păun as well as of the organizers of the workshops in the years 1998 and 2000, i.e., Alica Kelemenová and Rudolf Freund. The organizing committee of the Grammar Systems Week 2004 was formed by Erzsébet Csuhaj-VarjÚ, György Vaszil, and Mariann Kindl.

More than twenty registered participants from Austria, from the Czech Republic, Germany, Hungary, Italy, Romania, Spain, and even from India attended the Grammar Systems Week 2004; it was organized in such a way that after the scientific talks in the morning, the afternoons were reserved for free discussions, especially devoted to specific topics in the grammar systems area.

The scientific program started on Monday morning with Erzsébet Csuhaj-Varjú opening the workshop and then giving a very nice and informative presentation on 'Grammar systems: Past, present, and future'. Then Gheorghe Păun initiated a very interesting discussion on 'Grammar systems vs. membrane computing: A preliminary approach,' which was continued on Thursday, when Rudolf Freund and Marion Oswald presented their ideas for 'Modelling grammar systems by tissue P systems'.

Jozef Kelemen spoke about 'Embodiment - A computational point of view' and Alica Kelemenová considered 'Monocultures and homogeneous environment in

ECO-GRAMMAR SYSTEMS'. Jürgen Dassow in his invited talk 'ON COOPERATING DISTRIBUTED GRAMMAR SYSTEMS WITH COMPETENCE BASED START AND STOP CONDITIONS' introduced very interesting (new) aspects for choosing a component in CD grammar systems.

Kamala Krithivasan as an invited speaker presented several new and interesting results of the Indian group on 'DISTRIBUTED PROBABILISTIC FINITE AUTOMATA', 'DISTRIBUTED 2-WAY FINITE STATE QUANTUM AUTOMATA', and 'SIMPLE SPLICING GRAMMAR SYSTEMS'. Petr Sosík, on leave from London, Ontario, presented 'DNA INVOLUTIONS AND HAIRPIN STRUCTURES' and, together with Peter Sebestyén, 'MULTIPLE ROBOTS IN SPACE: AN ADAPTIVE ECO-GRAMMAR MODEL.'

Interesting joint work with Henning Bordihn was presented by Markus Holzer dealing with 'CD GRAMMAR SYSTEMS AS MODELS OF DISTRIBUTED PROBLEM SOLVING, REVISITED', by György Vaszil considering 'CD GRAMMAR SYSTEMS WITH LL(K) CONDITIONS', and by Suna Bensch talking about 'ACTIVE SYMBOLS IN PURE SYSTEMS.'

Bettina Sunckel presented nice results 'ON METALINEAR CD GRAMMAR SYSTEMS.' Liliana Cojocaru presented new results 'ON THE TIME, SPACE, AND COMMUNICATION COMPLEXITY OF COOPERATING DISTRIBUTED GRAMMAR SYSTEMS' and on 'PARALLEL COMMUNICATING PUSHDOWN TRANSDUCER SYSTEMS'.

In the second session on eco-grammar systems (on Wednesday late morning), Francesco Bernardini presented new interesting results obtained in Sheffield, United Kingdom, together with Marian Gheorghe on 'POPULATION P SYSTEMS AND GRAMMAR SYSTEMS,' and Katalin Lázár then spoke about 'ECO-GRAMMAR SYSTEMS: AN APPROACH TO THE CRAWLERS' PROBLEM'.

Gemma Bel Enguix and Maria Dolores Jiménez-López were successful in 'EXPLAINING LANGUAGE CHANGE WITH MEMBRANES', and Maria Adela Grando tried to answer the challenging question (posed together with Victor Mitrana) 'CAN PC GRAMMAR SYSTEMS BENEFIT FROM CONCURRENT PROGRAMMING?'

In the afternoon sessions, the participants of the GRAMMAR SYSTEMS WEEK 2004 were invited to meet for discussions on specific topics in the grammar systems area as well as to establish new co-operations: On Monday afternoon, general issues and descriptional complexity issues were discussed. The afternoon session on Tuesday started with Maria Dolores Jiménez-López asking 'WHAT CAN GRAMMAR SYSTEMS DO FOR LINGUISTICS?' and Suna Bensch (together with Helmut Jürgensen) 'MODELLING DIALOGUES BY GRAMMAR SYSTEMS' and was devoted to linguistic applications. Evolutionary models and eco-grammar systems were considered on Wednesday, and, finally, Thursday afternoon was devoted to bio-computing and unconventional models of computing.

The workshop ended on Friday with a special session on results obtained during the workshop, where Gheorghe Păun and Rudolf Freund presented some new ideas concerning the relationship between grammar systems and P systems.

Detailed informations about the scientific program of the GRAMMAR SYSTEMS

Week 2004 as well as a lot of very nice pictures made by György Vaszil can be found at http://www.sztaki.hu/tcs/gsweekarchiv.

The whole workshop was held in a very kind and familiar atmosphere, also allowing for intensive discussions of new ideas and results. During the coffee breaks, coffee and soft drinks as well as very good sweet cookies were available directly in the lecture hall, and most of the participants met for lunch in the mensa of SZTAKI before meeting again for the afternoon sessions.

The main (social) event of the Grammar Systems Week 2004 was the Workshop Dinner at the famous Hotel Gellért. The participants enjoyed excellent Hungarian food and wine. Celebrating the tenth anniversary of the publication of the monograph 'Grammar Systems: A Grammatical Approach to Distribution and Cooperation' and the seventeenth anniversary of the *birth* of grammars systems, an enormous birthday cake with 17 candles was presented by Erzsébet Csuhaj-Varjú and then distributed by the four authors of the monograph (Erzsébet Csuhaj-Varjú, Jürgen Dassow, Jozef Kelemen, Gheorghe Păun). This surprising excellent dessert was *the* highlight of this very successful international workshop, promising a successful continuation of interesting research in this area of grammar systems for (at least) another 10 or 17 years...

Pictures from CCA 2004 (by M. Ziegler)

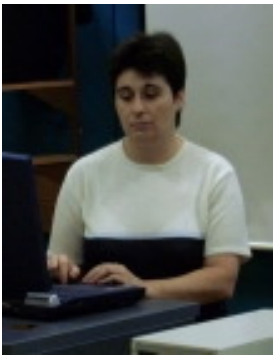
Pictures from GS 2004 (by Rudolf Freund)

Jürgen Dassow

Alica Kelemenová,
Petr Sosík

Liliana Cojocaru

Erzsébet Csuhaj-Varjú



Gemma Bel Enguix

Maria Dolores
Jiménez López

Rudolf Freund,
Bettina Sunckel

Suna Bensch,
Marion Oswald

István Katsányi,
Petr Sebestyén

Report on SOS 2004

Structural Operational Semantics
August 30, 2004, London, United Kingdom

Peter D. Mosses

This was one of the pre-conference workshops affiliated with CONCUR 2004. It was held concurrently with five other events in the beautiful setting of the Royal Society. The concurrency was a mixed blessing for the many participants who were interested in the topics of more than one of the events: some stuck with their main preference all day, while others commuted between the competing events. Around 25 participants attended essentially the whole SOS workshop.

The scientific programme for the workshop was kicked off by Andrew Pitts, who delivered a beautifully paced talk, in which he surveyed his work on a lightweight theory of structural recursion and alpha-congruence for languages that involve binders. The programme included two very clear tutorials by David Sands and Rob van Glabbeek. David showed how the use of a higher-order syntax representation of contexts combines smoothly with higher-order syntax for evaluation rules, so that definitions can be extended to work over contexts, and argued that this leads to a useful technique for directly reasoning about operational equivalence. Rob gave an entertaining and interactive presentation of his work on how to assign meaning to transition systems specifications with negative premises.

The three interesting contributed talks were delivered by Matthias Mann, Olivier Tardieu and Marija Kulas. Matthias presented his difficult result on the congruence of bisimulation in a non-deterministic call-by-need lambda calculus. Olivier gave a talk on his new deterministic logical semantics for Esterel. Marija's talk on the other hand described a structural operational semantics for Prolog that can handle backtracking. The papers appear in the preliminary proceedings of the workshop, available at `http://www.brics.dk/NS/04/1/`.

The workshop also marked the publication of a large special issue of the Journal of Logic and Algebraic Programming (Vols. 60–61) devoted to SOS. The special issue features a reprint, with corrections, of Gordon Plotkin's seminal Aarhus lecture notes from 1981 on *A Structural Approach to Operational Semantics*. Three of the authors of papers appearing in the special issue gave talks at the workshop: Bartek Klin gave a very lucid account of his work on the derivation of congruences from bialgebraic semantics; my own talk was on a modular variant of SOS; and Ralf Lämmel closed the workshop with a well attended talk on evolution scenarios for rule-based systems.

The workshop was efficiently co-organized by Luca Aceto, Wan Fokkink, and Irek Ulidowski. It was felt to have been a success, both scientifically and socially, and a second edition should take place some time next year.

Report on VODCA 2004

First Intl Workshop on Views On Designing Complex Architectures September 11–12, Bertinoro, Italy

Maurice ter Beek

The *First International Workshop on Views On Designing Complex Architectures* (VODCA 2004) was held in Bertinoro, Italy on 11–12 September 2004, as a satellite event to the *Fourth International School on Foundations of Security Analysis and Design* (FOSAD 2004). VODCA 2004 aimed at providing a platform for young scientist to present their research views on all areas related to the design of complex architectures, with a special focus on the security and management of information. The Programme Committee consisted of 11 members from 9 different countries and it was chaired by Fabio Gadducci from the University of Pisa, Italy. The organising committee consisted of 5 members from 3 different countries and it was chaired by Maurice ter Beek from the CNR Institute of Information Science and Technologies, Pisa, Italy. The publicity chair, finally, was Rebeca Díaz Redondo from the University of Vigo, Spain.

The workshop took place at the well-known University Residential Centre of Bertinoro—situated in Bertinoro, Italy—a small village on a scenic hill with a wonderful panorama. The village is dominated by *la rocca* (The Rock), a commanding fortress built on top of the village's hill, where the centre is situated. More information about this centre can be found on their web site `www.centrocongressibertinoro.it`. More information about VODCA 2004 can be found on their web site `www-gris.det.uvigo.es/vodca`.

A total of 17 papers were submitted to the workshop by researchers from 11 different countries. After a regular refereeing process 9 of them were selected for a long presentation, while 3 of them were selected for a short presentation. In addition to the presentations of original research results, the programme also included 5 invited lectures. The proceedings of the workshop will be published as a separate volume in the Electronic Notes in Theoretical Computer Science (ENTCS) series.

The workshop had a total of 28 participants from 10 different countries. The *conference dinner* took place on Saturday evening in the Belvedere restaurant and it was closed by a toast that remained true to the name of the workshop. The participants all shared the opinion that VODCA 2004 was a well-organized and successful workshop and they declared to look forward to a follow-up.

Pictures from VODCA 2004 (by M.H. ter Beek)

The chairs of VODCA: Fabio Gadducci and Maurice ter Beek

Raymond McGowan

Costantino Pistagna

Stefano Bistarelli

Sanjay Rawat

Sebastian Nanz

REPORT ON WACAM 04

Workshop on Word Avoidability, Complexity and Morphisms 17 July 2004, Turku, Finland

Manfred Kudlek

WACAM (Workshop on Word Avoidability, Complexity and Morphisms) was held at Turku on July 17, 2004, at PharmaCity. It was attended by 33 participants from 10 countries, in details

CA	5	DE	1	ES	1	FR	4	JP	1
CZ	1	EG	1	FI	13	IT	3	RU	3

WACAM was organized by GWÉNAËL RICHOMME. The scientific program consisted of 2 invited lectures and 8 contributions. JEFFREY SHALLIT, in the first invited lecture *'Avoidability in Words: Recent Results and Open Problems'* gave a nice and interesting historical survey on problems on words like square and cube freeness and avoidability, new results and open problems like the existence of an infinite sequence on $\{1,2,3\}$ avoiding ww'.

The second one by ANNA FRID on *'Possible Growth of Arithmetical Complexity'* was a good presentation on the arithmetical complexity of infinite words, giving a new class with complexities from $O(n \log n)$ to $O(n^2)$.

To mention are also the other good presentations by PETER LEUPOLD on uniformly n-bounded duplication codes, NARAD RAMPERSAD on squares and overlaps in the Thue-Morse sequence, ALI ABERKANE on the number of ternary words avoiding Abelian cubes, STEFANO VARRICCHIO on avoidable sets and well-quasi orders, TIMO KÄRKI on transcendence of k-ary numbers of subword complexity $2n + 1$, EMMANUEL HYON on relations between factorization of mechanical words and continued fractions, and by VESA HALAVA on infinite solutions of the marked PCP. Unfortunately, PASCAL OCHEM could not come, due to a strike in Paris.

The proceedings, containing all invited lectures and contributions, edited by GWÉNAËL RICHOMME, have been published as **LoRIA** Technical Report 2004-07. In the breaks coffee, tea, was offered. Lunch was in the restaurant of PharmaCity.

Report on WMC5

Fifth Workshop on Membrane Computing
June 14 – 16, 2004, Milano, Italy

Rudolf Freund

Immediately after DNA10, from June 14 to 16 the already fifth workshop on Membrane Computing (WMC5) took place in Milano, Italy, also organized by the group of Giancarlo Mauri (Daniela Besozzi, Claudio Ferretti, Alberto Leporati, Giancarlo Mauri, Claudio Zandron) from the Dipartimento di Informatica, Sistemistica e Comunicazione, Universitá degli Studi di Milano Bicocca.

WMC5 was realized under the auspices of the European Molecular Computing Consortium (EMCC) and supported by the IST-2001-32008 project MolCoNet. The workshop attracted more than fifty registered participants from all over Europe, most of them from Italy, Spain, and Romania, but also from China, Japan, and India. Detailed informations about WMC5 can be found at the P Systems Web Page (`http://psystems.disco.unimib.it`); the Pre-Proceedings contain the material of 7 invited talks and 34 regular papers. The program committee consisted of Giancarlo Mauri, Gheorghe Păun (chair), Grzegorz Rozenberg, and Arto Salomaa. Before the scientific program started, the participants got the chance to join an excursion to the idyllic town of Bergamo on Sunday, June 13. As throughout the whole period of the workshop, the weather was sunny and warm.

On Monday morning, Grzegorz Rozenberg perfectly lead over from DNA10 to WMC5 with his exciting invited talk 'Selectivity in molecular computing.' On Monday afternoon, Maurice Margenstern in his invited talk presented an interesting and challenging joint work with Loïc Colson, Nataša Jonoska, and Gheorghe Păun 'About P systems and lambda-calculus.'

On Tuesday morning, the first session consisted of two invited talks: Vincenzo Manca outlined an interesting framework for investigating the behaviour of dynamic processes based 'On the Dynamics of P systems.' Rudolf Freund gave an introduction to the definitions and to first results for 'Asynchronous P systems', i.e., P systems working in the sequential derivation mode. Kazunori Ueda (with Norio Kato) developed a special programming language 'LMNtal: a language model with links and membranes', thus showing a nice application of computing with membranes.

On Wednesday morning, Erzsébet Csuhaj-Varjú in her invited talk presented a fascinating and comprehensive introduction to 'P automata: Models, Results, and Research Topics.' Mario J. Pérez-Jiménez, in his invited talk on Wednesday afternoon, presented mathematically deep results on determining 'Complexity

CLASSES IN MEMBRANE COMPUTING.' As the topics of the invited talks, also the contents of most of the talks based on regular contributions ranged from pure theoretical results with new models, improvements of the descriptional complexity of various classes of P systems to biologically motivated applications of P systems.

During the coffe breaks, coffee and soft drinks as well as very good sweet cookies were available in a room nearby the lecture hall, and also the lunches with typical Italian food were served there, which also gave the possibility to continue scientific discussions on molecular computing, especially on P systems. The participants of the workshop really enjoyed the hospitality of GIANCARLO MAURI and his team as well as the familiar atmosphere of the meeting.

As it has already become a nice tradition in this series of workshops organized by him or having him as program chair, all participants of WMC5 got a diploma signed by GHEORGHE PĂUN. Although he did not give a talk himself, many results presented at the workshop had been influenced by GHEORGHE PĂUN directly or at least by pointing out new ideas or open problems to his colleagues. The vivid interest of the participants of WMC5 in old and new variants of P systems and their applications was a promise for the future of the area of P systems.

Forthcoming important events (conferences, workshops, etc.) in the area of Membrane Computing already are or soon will be announced at the P Systems Web Page (`http://psystems.disco.unimib.it`); for example, the 1ST BRAINSTORMING WORKSHOP ON UNCERTAINTY IN MEMBRANE COMPUTING will be arranged in Palma de Mallorca, from November 8 to 10, 2004, and at the beginning of February 2005, the THIRD BRAINSTORMING WEEK ON MEMBRANE COMPUTING will take place in Sevilla, Spain.

The sixth workshop on Membrane Computing (WMC6) will be held in Vienna, Austria, in the third week of July, 2005; details will be available at the URL `http://www.emcc.at/WMC06`.

Pictures from WMC5 (by Rudolf Freund)

Gheorghe Păun

Giancarlo Mauri, Grzegorz Rozenberg

Claudio Zandron

Alberto Leporati, Giuditta Franco, Vincenzo Manca

Rudolf Freund, Gheorghe Păun

Kazunori Ueda

Abstracts of PhD Theses

Abstract of PhD Thesis

Author:	Michael Domaratzki
Title:	Trajectory-Based Operations
Language:	English
Supervisor:	Dr. Kai T. Salomaa
Institute:	School of Computing, Queen's University, Canada
Date:	19 August 2004

Abstract

Shuffle on trajectories was introduced by Mateescu *et al.* as a method of generalizing several studied operations on words, such as the shuffle, concatenation and insertion operations. This natural construction has received significant and varied attention in the literature. Shuffle on trajectories is a parameterized class of language operations, with each language T over a two-letter alphabet defining a unique language operation, which is denoted $\sqcup\!\sqcup_T$. In this thesis, we consider several unexamined areas related to shuffle on trajectories.

We first examine the state complexity of shuffle on trajectories. The state complexity of an operation is an expression of the effect of applying that operation to regular languages, in terms of the size of the minimal finite automata recognizing these languages. We find that the density of the set of trajectories is an appropriate measure of the complexity of the associated operation, since low density sets of trajectories yield less complex operations. This work constitutes the first attempt to examine the state complexity of a class of operations, rather than a fixed, individual operation.

We introduce the operation of deletion along trajectories, which serves as an inverse to shuffle on trajectories. The operation is also of independent interest, and we examine its closure properties. We find that, unlike the case of shuffle on trajectories, there exist non-regular sets of trajectories T such that the associated deletion along trajectories operation, $\rightsquigarrow_T$, does not preserve regularity.

The study of deletion along trajectories also leads to the study of language equations and systems of language equations with shuffle on trajectories. We investigate several language equation forms, including language decompositions, that is, language equations of the form $L = X_1 \sqcup\!\sqcup_T X_2$ where L is fixed and X_1, X_2 are unknown. The decidability of whether a given regular language has a shuffle decomposition (in terms of unrestricted, classical shuffle) is an open problem. We positively solve the decomposition decidability problem for regular languages

and a significant and interesting class of sets of trajectories, namely the letter-bounded sets of trajectories. This class includes concatenation, insertion and other operations as particular cases.

The notion of shuffle on trajectories also has applications to the theory of codes. Each shuffle on trajectories operation $\sqcup\!\sqcup_T$ defines a class of languages, which we call T-codes. Several of these language classes are important in the theory of codes, including the prefix-, suffix-, biprefix-codes and the hypercodes. We investigate these classes of languages, decidability questions, and related binary relations. Of particular interest are characterizations relating to convexity and transitive closure, and maximal T-codes.

We conclude with results relating to iteration of shuffle and deletion on trajectories. We characterize the smallest language closed under shuffle on trajectories or deletion along trajectories, as well as generalize the notion of primitive words and primitive roots. Further examination of language equations are also possible with the iterated counterparts of shuffle and deletion along trajectories.

Author's correspondence address

Michael Domaratzki
Jodrey School of Computer Science
Acadia University
Wolfville, NS
B4P 2R6
CANADA
email: mike.domaratzki@acadiau.ca
URL: euler.acadiau.ca/~mdomarat

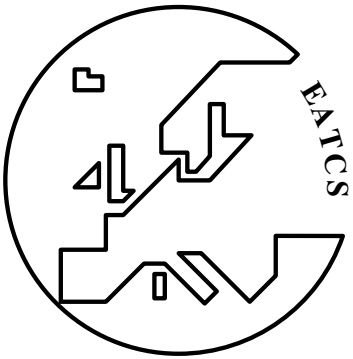
European

Association for

Theoretical

Computer

Science



E A T C S

COUNCIL OF THE EATCS

BOARD

President:	M. Nielsen	Denmark
Vice Presidents:	J. van Leeuwen,	The Netherlands
	P. Spirakis,	Greece
Treasurer:	D. Janssens,	Belgium
Secretary:	B. Rovan,	Slovakia
Bulletin Editor:	V. Sassone,	UK

OTHER COUNCIL MEMBERS

P. Degano	Italy
M. Dezani-Ciancaglini	Italy
J. Díaz	Spain
Z. Ésik	Hungary
J. Esparza	UK
H. Gabow	USA
A. Gibbons	UK
K. Iwama	Japan
J.-P. Jouannaud	France
J. Karhumäki	Finland
D. Peleg	Israel
J. Sgall	Czech Republic
A. Tarlecki	Poland
W. Thomas	Germany
D. Wagner	Germany
E. Welzl	Switzerland
G. Wöeginger	Austria
U. Zwick	Israel

MONOGRAPHS EDITORS AND TCS

Monographs Editors:	W. Brauer	Germany
	G. Rozenberg	The Netherlands
	A. Salomaa	Finland
TCS Editors:	G. Ausiello	Italy
	D. Sannella	UK
	M.W.Mislove	USA

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, a Treasurer and a Secretary. 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;"
- Publication of the "EATCS Monographs" and "EATCS Texts;"
- Publication of the journal "Theoretical Computer Science."

Other activities of EATCS include the sponsorship or the cooperation in the organization of various more specialized meetings in theoretical computer science. Among such meetings are: CAAP (Colloquium on Trees in Algebra and Programming), TAPSOFT (Conference on Theory and Practice of Software Development), STACS (Symposium on Theoretical Aspects of Computer Science), MFCS (Mathematical Foundations of Computer Science), LICS (Logic in Computer Science), ESA (European Symposium on Algorithms), Conference on Structure in Complexity Theory, SPAA (Symposium on Parallel Algorithms and Architectures), Workshop on Graph Theoretic Concepts in Computer Science, International Conference on Application and Theory of Petri Nets, International Conference on Database Theory, Workshop on Graph Grammars and their Applications in Computer Science.

Benefits offered by EATCS include:

- Subscription to the "Bulletin of the EATCS;"
- Reduced registration fees at various conferences;
- Reciprocity agreements with other organizations;
- 25% discount in purchasing ICALP proceedings;
- 25% discount in purchasing books from "EATCS Monographs" and "EATCS Texts;"
- Discount (about 70 %) per individual annual subscription to "Theoretical Computer Science;"
- Discount (about 70 %) per individual annual subscription to "Fundamenta Informaticae."

(1) THE ICALP CONFERENCE

ICALP is an international conference covering all aspects of theoretical computer science and now customarily taking place during the second or third week of July.

Typical topics discussed during recent ICALP conferences are: computability, automata theory, formal language theory, analysis of algorithms, computational complexity, mathematical aspects of programming language definition, logic and semantics of programming languages, foundations of logic programming, theorem proving, software specification, computational geometry, data types and data structures, theory of data bases and knowledge based systems, cryptography, VLSI structures, parallel and distributed computing, models of concurrency and robotics.

SITES OF ICALP MEETINGS:

- Paris, France 1972
- Saarbrücken, Germany 1974
- Edinburgh, Great Britain 1976
- Turku, Finland 1977
- Udine, Italy 1978
- Graz, Austria 1979
- Noordwijkerhout, The Netherlands 1980
- Haifa, Israel 1981
- Aarhus, Denmark 1982
- Barcelona, Spain 1983
- Antwerp, Belgium 1984
- Nafplion, Greece 1985
- Rennes, France 1986
- Karlsruhe, Germany 1987
- Tampere, Finland 1988
- Stresa, Italy 1989
- Warwick, Great Britain 1990
- Madrid, Spain 1991
- Wien, Austria 1992
- Lund, Sweden 1993
- Jerusalem, Israel 1994
- Szeged, Hungary 1995
- Paderborn, Germany 1996
- 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

(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) EATCS MONOGRAPHS AND TEXTS

This is a series of monographs published by Springer-Verlag and launched during ICALP 1984; more than 50 volumes appeares. The series includes monographs in all areas of theoretical computer science, such as the areas considered for ICALPs. Books published in this series present original research or material of interest to the research community and graduate students. Each volume is normally a uniform monograph rather than a compendium of articles. The series also contains high-level presentations of special topics. Nevertheless, as research and teaching usually go hand in hand, these volumes may still be useful as textbooks, too. Texts published in this series are intended mostly for the graduate level. Typically, an undergraduate background in computer science is assumed. However, the background required may vary from topic to topic, and some books may be self-contained. The texts cover both modern and classical areas with an innovative approach that may give them additional value as monographs. Most books in this series will have examples and exercises. Updated information about the series can be obtained from the publisher.

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

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

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.

(4) THEORETICAL COMPUTER SCIENCE

The journal *Theoretical Computer Science*, founded in 1975, is published by Elsevier Science Publishers, Amsterdam, currently in 20 volumes (40 issues) a year. Its contents are mathematical and abstract in spirit, but it derives its motivation from practical and everyday computation. Its aim is to understand the nature of computation and, as a consequence of this understanding, provide more efficient methodologies.

All kinds of papers, introducing or studying mathematical, logical and formal concepts and methods are welcome, provided that their motivation is clearly drawn from the field of computing.

Papers published in *TCS* are grouped in three sections according to their nature. One section, "Algorithms, automata, complexity and games," is devoted to the study of algorithms and their complexity using analytical, combinatorial or probabilistic methods. It includes the fields of abstract complexity (i.e., all the results about the hierarchies that can be defined using Turing machines), of automata and language theory (including automata on infinite words and infinitary languages), of geometrical (graphic) applications and of system performance using statistical models. A subsection is the Mathematical Games Section, which is devoted to the mathematical and computational analysis of games.

The second section, "Logic, semantics and theory of programming," is devoted to formal methods to check properties of programs or implement formally described languages; it contains all papers dealing with semantics of sequential and parallel programming languages. All formal methods treating these problems are published in this section, including rewriting techniques, abstract data types, automatic theorem proving, calculi such as SCP or CCS, Petri nets, new logic calculi and developments in categorical methods.

The newly introduced third section is devoted to theoretical aspects of "Natural Computing."

The Editors-in-Chief of "Theoretical Computer Science" are:

G. Ausiello, Università di Roma 'La Sapienza', Dip. Inform. e Sistemistica,
via Salaria 113, 00198 Roma, Italy;
D. Sannella, University of Edinburgh, Lab. for Foundations of Computer Science,
Division of Informatics, King's Building, Mayfield Road, Edinburgh, EH9 3JZ, UK
M.W. Mislove, Tulane University, Dept. of Mathematics, New Orleans, LA 70118, USA.

ADDITIONAL INFORMATION

For further information please visit `http://www.eatcs.org`, or contact the Secretary of EATCS:

Prof. Dr. Branislav Rovan, Department of Computer Science, Comenius University,
SK-84248 Bratislava, Slovakia, Email: `secretary@eatcs.org`

DUES

The dues are €30 for a period of one year. If the initial membership payment is received in the period Dec 21st–Apr 20th, Apr 21st–Aug 20th, or Aug 21st–Dec 20th, then the first membership year will start on Jun 1st, Oct 1st or Feb 1st, respectively. Every contribution payment continues the membership for the same time period. In order to encourage double registration, we are offering a discount: SIGACT members can join EATCS for €25 per year. Additional €25 fee is required for ensuring the *air mail* delivery of the EATCS Bulletin outside Europe.

HOW TO JOIN EATCS

The easiest way to join EATCS is from our website `www.eatcs.org`. Alternatively, send the annual dues, or a multiple thereof (to cover a number of years), 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. When submitting your payment, please make sure to indicate your complete name and address. For this purpose you may use the form below.

Transfers of larger amounts may be made via the following bank account. 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

I would like to join EATCS / renew my EATCS membership and pay

€ ..... for my **membership fee** for .... year(s) (and € ..... for air mail delivery)

□ Please charge my credit card Total amount:€ .....

Card Number (**16 digits**): .....

□ EUROCARD/MASTERCARD □ VISA Exp. Date: .....

....
Date/Signature Address(if different from mailing address below)

□ Check Enclosed □ Transfered to EATCS account □ Cash Enclosed

□ ACM SIGART Member no. .....

Name: ..... First Name: ..... email: .....

Address: .....