ISSN 0252–9742

Bulletin

of the

European Association for Theoretical Computer Science

EATCS

Number 120 October 2016

EATCS Council Members

email addresses

Lars Arge	large@madalgo.au.dk
Jos Baeten	jos.baeten@cwi.nl
Lars Birkedal	large@madalgo.au.dk
Mikolaj Bojanczyk	bojan@mimuw.edu.pl
Fedor Fomin	fomin@ii.uib.no
Pierre Fraigniaud	pierre.fraigniaud@liafa.univ-paris-diderot.fr
Leslie Ann Goldberg	leslie.goldberg@cs.ox.ac.uk
Magnus Halldorsson	magnusmh@gmail.com
Monika Henzinger	monika.henzinger@univie.ac.at
Kazuo Iwama	iwama@kuis.kyoto-u.ac.jp
Christos Kaklamanis	kakl@ceid.upatras.gr
Antonin Kucera	tony@fi.muni.cz
Elvira Mayordomo	elvira@unizar.es
Michael Mitzenmacher	michaelm@eecs.harvard.edu
Anca Muscholl	anca@labri.fr
Luke Ong	luke.Ong@cs.ox.a.uk
Catuscia Palamidessi	catuscia@lix.polytechnique.fr
Giuseppe Persiano	giuper@dia.unisa.it
Alberto Policriti	alberto.policriti@uniud.it
Jean-Francois Raskin	jraskin@ulb.ac.be
Alberto Marchetti Spaccamela	alberto@dis.uniroma1.it
Vladimiro Sassone	vs@ecs.soton.ac.uk
Thomas Schwentick	thomas.schwentick@udo.edu
Paul Spirakis	p.spirakis@liverpool.ac.uk
Jukka Suomela	jukka.suomela@aalto.fi
Thomas Wilke	thomas.wilke@email.uni-kiel.de
Peter Widmayer	widmayer@inf.ethz.ch
Gerhard Wöeginger	g.j.woeginger@math.utwente.nl

Bulletin Editor: Kazuo Iwama, Kyoto, Japan
Cartoons: DADARA, Amsterdam, The Netherlands

The bulletin is entirely typeset by pdfTEX and ConTEXt in TXfonts.

All contributions are to be sent electronically to

`bulletin@eatcs.org`

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

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

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

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

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

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

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

Table of Contents

EATCS Matters

Dear EATCS members,

As you might know already, the Council of the EATCS elected me as the new president of the association at its meeting at ICALP 2016 in Rome. This is a great honour for me. It is also a great responsibility. It will be a hard job to follow on the footsteps of Luca Aceto, who led the association so expertly since the end of ICALP 2012 in Warwick. I promise to do my best.

I feel that, in the near future , the EATCS can (and should) play an even more important role in the development of theoretical computer science than ever. Some of my thoughts and feelings can be found in this Bulletin, in my interview with Luca Aceto ! I have a number of things that I would like to achieve during my term as president, but I welcome proposals from all of you. Feel free to contact me via email at president@eatcs.org, if you have anything that you would like to see the EATCS do for the benefit of the theoretical computer science community.

An important thing that we can all do in order to increase the impact of the EATCS is to recruit new members, especially amongst the young researchers who are the future of our field. I hope that you will contribute to recruiting new members at your institution and amongst your closest collaborators.

ICALP 2016, which took place in Rome, organised by Sapienza University, was a very successful conference with a truly

high-quality scientific programme. It had the largest number of submissions in history (515 papers). We thank the organizers Tiziana Calamoneri, Irene Finocchi, Nicola Galesi and Daniele Gorla from the Universita di Roma "La Sapienza", and the programme chairs Yuval Rabani (track A), Davide Sangiorgi (track B) and Michael Mitzenmacher (track C) for the perfect organization of the conference and the excellent choice of the scientific program.

The conference attendees were treated to invited talks by Devavrat Shah, Xavier Leroy, Seffi Naor (who replaced Subhash Khot at the last minute), and Marta Z. Kwiatkowska. The program of ICALP 2016 also included presentation of the EATCS Award 2016 to Dexter Kozen, the Gödel Prize 2016 to Steve Brookes and Peter O'Hearn, and the Presburger Award 2016 to Mark Braverman. As usual, a special award session took place to honour this yearś prize winners. Dexter Kozen received the EATCS Award 2016 and Stephen Brookes and Peter W. O'Hear shared the Gödel Prize 2016. The Presburger Award 2016 for young scientists was given to Mark Braverman and for the EATCS Distinguished Dissertation Award, three dissertations were selected by the committee for year 2015: "The Simple, Little and Slow Things Count: On Parameterized Counting Complexity", by Radu Curticapean. "Complexity Classification of Exact and Approximate Counting Problems", by Heng Guo. "Monoids as storage mechanisms", by Georg Zetzsche.

The organization of ICALP 2017 in Warsaw is proceeding well. It will again be organized in three tracks and the topics of *the tracks remain the same as this year. The Conference Chairs are Mikolaj Bojanczyk and Piotr Sankowski. The PC Chairs of the three tracks are Piotr Indyk (Track A) , Anca Muscholl (Track B) and Fabian Kuhn (Track C). The invited talks will be given by Mikolaj Bojanczyk, Monika Henzinger and Mikkel Thorup. I hope that you will submit your best paper to ICALP 2017!*

In this issue of the Bulletin, you will find the calls for nominations for the EATCS Award, the EATCS Fellows, the Presburger Award, the Gödel Prize and the EATCS Distinguished Dissertation Award. As usual, we are lucky to have very strong committees for each of the awards.

I encourage you to send nominations for these awards. I realize that this takes a little work, and that we are all very busy. However, individuals and papers can only receive awards if they are nominated. Moreover, awards put areas of, as well as inspirational figures in, theoretical computer science in the spotlight and can serve to inspire young researchers. I look forward to seeing who the award winners will be and to working with all of you to make the EATCS even more influential than it already is.

Paul Spirakis, Liverpool, UK and Patras, Greece

October 2016

Dear Reader,
Three years have passed since I started writing this letter, during which our president was Luca. You probably know an exercise game (mostly for kids) called a three-legged race. We often say here like "we did a good job with three-legs." I am not sure if this phrase is also common in Europe, but it is particularly common in my country to express a close cooperation between two people. All supports given to me by Luca in the past three years made possible our collaboration exactly suited for this phrase. Thank you very much, Luca! He worried about possible too much interference in my editorialship and often asked me if he was going too far. No, no, I greatly appreciate his "interference," without which I am not sure if I could have done this job without big problems.(By the way, the team game itself, three-legged race and its extensions, is very popular here. There is even a national championship game for 29-people-30-legs race for high school teams. The idea is of course good teamwork which is an important base of this country. I was in Barcelona this summer and went to la Merce. Among others, Castellers (human towers) was especially impressive to me in the same idea.) I am not sending any specific message to our new president, but Paul, your interaction, interference, or whatever, if you want, would be always welcome and equally appreciated.

This is an October issue with a full of pieces including several reports of ICALP and EATCS matters. Regular stuff (technical columns, reports of several

kinds, PhD thesis abstracts, book introduction, etc.) is also rich as usual, my sincere thanks to column editors and all the contributors.

One thing I have been thinking is a possible enhancement of technical contributions in general. As you know, we have plenty of journals for publishing original results and the role of our Bulletin is probably different. Note that we have technical columns for surveys of wide range of different topics that are invitation-base. So, if a submission is a (good) survey, I can forward it to a most suited column editor. In fact I have done this several times, which were mostly successful. A possible challenge is to invite more original contributions that are of high quality but may not be very much suited to existing journals. One example is short but cute papers often appearing in, e.g., IPL (I was in its editorial board and came across quite a few such papers), giving simpler and easier-to-read proofs of existing theorems. Or I am wondering if we can even accept write-ups of already published results from a bit different angle emphasizing their basic ideas by the same authors. I would like to discuss with Paul about this matter as soon as possible.

We are heading for a cold and dark season (well, sorry, I am a bit biased, I mean for many of us...). I love this season for many reasons; one of them is of course, outdoors in snow. I hope you will also have a good holiday season and a very nice new year.

Kazuo Iwama, Kyoto
October 2016

Interview with Paul Spirakis
President of the EATCS

Luca Aceto
ICE-TCS, School of Computer Science,
Reykjavik University
luca@ru.is

During its annual meeting at ICALP 2016 in Rome, the Council of the EATCS elected Paul Spirakis (University of Liverpool, UK, and University of Patras, Greece) as its new president. Paul is a well-known figure in the theoretical-computer-science community and truly needs no introduction. However, I felt that it might be a good idea to interview him briefly in order to give him the opportunity to present himself to the community and to discuss briefly some of his plans for his mandate as president of the EATCS.

I interviewed Paul Spirakis (abbreviated to PS in what follows) via email and present his answers to my questions in what follows. In order to preserve the style of Paul's answers, I did not edit them. I hope that the readers of the Bulletin of the EATCS will enjoy reading the text of the interview and will find it as interesting as I did.

The interview

LA: Paul, first of all, congratulations for your election to the presidentship of the EATCS. Could you introduce yourself and your work in a few sentences to the EATCS Community?

PS: My name is Paul G. Spirakis. I was born on August 29, 1955. I got my first degree (Electrical Eng.) from NTUA Athens, Greece, in 1978. I got my MSc and PhD from Harvard in Applied Math/Computer Science (1979 and 1982 respectively). I am a Theorist working in Algorithms and Complexity. I have contributed to the fields of Distributed Computing and Parallel Computing foundations, to the field of Random Graphs and to the subject of average case analysis and probabilistic analysis of algorithms. Also to the field of Algorithmic Game Theory.

LA: Which of your many results are you most fond of?

PS: A hard question to answer because I love many results of mine and co-authors. If I have to point out one, I would point out my result (obtained together with Charalampos Tsaknakis) on polynomial time approximations to Nash Equilibria. The approximation obtained is still unbeatable (till now) but most important is the method (classic optimization but a bit unusual to CS people) and the fact that my co-author (and fellow student in Lyceum) has as his official work the directing of the contracts office of CTI (but he is also a bright Mathematician in his free time).

LA: What changes have you seen in the TCS community since you finished your PhD?

PS: Well, many changes! First of all, the TCS community is now much broader and more dynamic. There are many more Conferences around, due to explosion of some sub-fields! One can see the dramatic effect coming from the intersection of Algorithms, Complexity and Game Theory! This new field (which takes selfishness of players into account) touches important questions in Economics. The Web, and efforts to understand it, is also an evolving field! Many other things have happened (among them the increasing interest of statistical Physics scientists to TCS and in particular in the subjects of Complex Nets but also SAT, the rise and fall (and maybe rise again) of Shared Memory Parallel Algorithms, the appearance and promising future of Quantum Computing, the strong position of Distributed Computing Foundations in many Research Fora etc). Also the new rise (again) of AI and algorithmic Learning! Indeed, while in the past a Theorist believed that he/she knows all CS Foundations, this is not the case now. TCS is expanding rapidly and is affecting many other Sciences. Needless to say that the P versus NP question is now recognized as one of the big unsolved questions in Mathematics, Engineering and many other Sciences.

LA: What do you think are the main challenges for the EATCS right now? What do you plan to focus on during your term as president of EATCS?

PS: A big challenge for EATCS is to become more attractive for all Theorists in Europe and the world. Another big challenge for EATCS is to be able to use its excellent human potential towards offering strong quality scientific advice to the decision makers in Europe. I shall focus on increasing the strength and open character of EATCS. A hard task, given the high level reached by previous leaderships! But I feel that we have a great Council and great Vice Presidents and all together we can advance EATCS. Also a great and trusted Treasurer and also a great and very efficient Secretarial Office Also our Bulletin Editor and column editors! Many of our excellent Members serve in very important Committees! It is not a one man's job.

LA: In your opinion, what role can the EATCS play for TCS in the coming years? In general, do you think that professional societies are still important in our age?

PS: I shall answer via a paradigm from game theory: Consider a set of selfish individuals (selfish here means that each individual just tries to maximize his/her expected utility) in interaction. Game Theory shows that such situations result in some equilibrium (out of many possible in general), not always good for all participants (and also not always good for the whole set — the Price of Anarchy notion). In order for a desirable equilibrium to be obtained, the Theory of Mechanisms Design indicates that some rules have to be set for all (without restricting each one's freedom of choice of course). EATCS, and any good professional society, is somehow a Mechanism for the collective action of many Scientists, trying to improve the framework under which all Theorists (in the case of EATCS) can benefit, without restricting in any way the advance and creativity of each one. Thus, professional societies are very important in our age. Especially Societies that unite Scientists from many countries! After all, Science is universal and maintaining communication among all of them is very crucial. Especially in our days, but also in the not so far past, where Europe and the world was (and might again become) divided or fragmented. Let me also state the obvious facts: that such Societies try to protect and advance our Profession, help to recognize young talents, raise examples of good scientists that can inspire all of us...

LA: What is your view on the funding scenario for TCS research in Europe right now, both at national level and at European level? What can the EATCS do to help ensure that TCS continue to receive appropriate funding?

PS: Different European countries have different levels of funding of Theory. In many examples, austerity measures (perhaps rational for general reasons) result in severely cutting national funds for Research. It is easy to see that public opinion and politicians would prefer to fund "practical" research with short term goals. Of course EU has some nice mechanisms for funding long term research (like the FET scheme and also the most recent and very successful ERC scheme). What EATCS can do (and should do) is to explain in simple language, to all, the importance of Foundations and Theory towards radical (and some times unexpected) future technologies and applications! EATCS can select and speak about important very successful examples (e.g. how some basic algorithms have improved the search for information, or the effectiveness of many industrial processes, how verification of correctness is very important for any new hardware and software design, especially for critical applications — just to mention some big examples). I must also mention the many important achievements of AI and the big fight to cope with Complexity! In this respect, cooperation among several TCS and CS Societies (all over the world) is very important because of the positive synergy towards such a goal.

LA: Increasingly many funding agencies across the globe mandate that the results and data of the research they support be available in open access form. What is your opinion on open access to research results and data?

PS: Let me first note some remarkable exceptions: For example military funding of research (in the most advanced countries) and also competitive industrial research are examples that are done in complete secrecy. In the past, cryptography has suffered from such examples. Also in many situations in industry, crucial data are not so open to access. This is of course due to competition in markets and profit. This being said, it is very positive that funding agencies insist on the availability of results and data in open access. I am definitely in favour, as most of us. The more open Theory is, the best is achieved for our research and education.

But the issue has some sides which complicate it. Research results and data, in some cases, may have a big financial value. Then the rules of Economy apply and the issue is who gets most of this value! Also, the maintenance of open publishing and access has costs and this implies that somebody has to pay for the costs. Many creators of original thought are in the same or similar position with us (for example journalists, authors of books on any subject, film makers etc). Intellectual property rights are important to be fair to the creators. Whole industries exist whose business is to advance (and profit from) intellectual products. Many people work in such industries and such industries are some times a big part of national economies. Tax paying citizens are affected if the society decides to somehow replace all this by a new model. I believe that we are in a transient period. Fortunately things are simpler for Theory! TCS is in the most innocent side of all this and thus it is easier for Theory to achieve Open Access. In any case we need to closely follow and understand the dynamic sides of the issue and its economics.

LA: Do you have any specific message you'd like to send to the TCS community?

PS: Yes, join the EATCS and be active in it! Its long history itself votes for the value of its existence! And there are so many benefits! Awards, discounts in Conferences, the Bulletin, our Web site, and most importantly a polite and high level collective forum that aims to help Theory to advance further.

Paul G. Spirakis
Professor, University of Liverpool (UK) and Patras University (Greece)
Director Research and Academic Computer Technology Institute (CTI)
President of the European Association for Theoretical Computer Science
Member of Academia Europaea
Fellow of EATCS
google: paul spirakis
email: P.Spirakis@liverpool.ac.uk
tel (work).: 0151-795-4255

The EATCS Award 2017

Call for Nominations

Deadline: December 31, 2016

The European Association for Theoretical Computer Science (EATCS) annually honours a respected scientist from our community with the prestigious EATCS Distinguished Achievement Award. The award is given to acknowledge extensive and widely recognized contributions to theoretical computer science over a life long scientific career. For the EATCS Award 2017, candidates may be nominated to the Award Committee consisting of

- Fedor Fomin (chair),
- Christos Papadimitriou and
- Jean-Éric Pin.

Nominations will be kept strictly confidential. They should include supporting justification and be sent by e-mail to the chair of the EATCS Award Committee:

Fedor Fomin
`eatcs-award@eatcs.org`

Previous recipients of the EATCS Award are:

R.M. Karp (2000)	C. Böhm (2001)	M. Nivat (2002)
G. Rozenberg (2003)	A. Salomaa (2004)	R. Milner (2005)
M. Paterson (2006)	D.S. Scott (2007)	L.G. Valiant (2008)
G. Huet (2009)	K. Mehlhorn (2010)	B. Trakhtenbrot (2011)
M.Y. Vardi (2012)	M.E. Dyer (2013)	G.D. Plotkin (2014)
C. Papadimitriou (2015)	Dexter Kozen(2016)	

The next award will be presented during ICALP 2017 in Warsaw, Poland.

The Presburger Award for Young Scientists 2017

Call for Nominations

Deadline: 31 December 2016

Starting in 2010, the European Association for Theoretical Computer Science (EATCS) established the Presburger Award. The Award is conferred annually at the International Colloquium on Automata, Languages and Programming (ICALP) to a young scientist (in exceptional cases to several young scientists) for outstanding contributions in theoretical computer science, documented by a published paper or a series of published papers.

The Award is named after Mojżesz Presburger who accomplished his path-breaking work on decidability of the theory of addition (which today is called Presburger arithmetic) as a student in 1929.

Nominations for the Presburger Award can be submitted by any member or group of members of the theoretical computer science community except the nominee and his/her advisors for the master thesis and the doctoral dissertation. Nominated scientists have to be at most 35 years at the time of the deadline of nomination (i.e., for the Presburger Award of 2017 the date of birth should be in 1981 or later). The Presburger Award Committee of 2017 consists of Stephan Kreutzer (TU Berlin), Marta Kwiatkowska (Oxford, chair) and Jukka Suomela (Aalto). Nominations, consisting of a two page justification and (links to) the respective papers, as well as additional supporting letters, should be sent by e-mail to:

Marta Kwiatkowska
`presburger-award@eatcs.org`

The subject line of every nomination should start with *Presburger Award 2017*, and the message must be received before **December 31st, 2016**.

The award includes an amount of 1000 Euro and an invitation to ICALP 2017 for a lecture.

Previous Winners:

Mikołaj Bojańczyk, 2010	Patricia Bouyer-Decitre, 2011
Venkatesan Guruswami, 2012	Mihai Pătraşcu, 2012
Erik Demaine, 2013	David Woodruff, 2014
Xi Chen, 2015	Mark Braverman, 2016

Official website: `http://www.eatcs.org/index.php/presburger`

EATCS Distinguished Dissertation Award 2016

Call for Nominations

Deadline: 31 December 2016

The EATCS establishes the Distinguished Dissertation Award to promote and recognize outstanding dissertations in the field of Theoretical Computer Science.

Any PhD dissertation in the field of Theoretical Computer Science that has been successfully defended in 2016 is eligible.

Three dissertations will be selected by the committee for year 2016. The dissertations will be evaluated on the basis of originality and potential impact on their respective fields and on Theoretical Computer Science.

Each of the selected dissertations will receive a prize of 1000 Euro. The award receiving dissertations will be published on the EATCS web site, where all the EATCS Distinguished Dissertations will be collected.

The dissertation must be submitted by the author as an attachment to an email message sent to the address `giuper@gmail.com` with subject `EATCS Distinguished Dissertation Award 2016` by `31 December 2016`. The body of the message must specify:

- Name and email address of the candidate;
- Title of the dissertation;
- Department that has awarded the PhD and denomination of the PhD program;
- Name and email address of the thesis supervisor;
- Date of the successful defense of the thesis.

A five page abstract of the dissertation and a letter by the thesis supervisor certifying that the thesis has been successfully defended must also be included. In addition, the author must include an endorsement letter from the thesis supervisor and can include one more endorsement letter.

The dissertations will be selected by the following committee:

- Giuseppe Persiano (chair)
- Elvira Mayordomo
- Dale Miller
- Jaroslav Nesetril
- Damian Niwinski
- David Peleg
- Vladimiro Sassone

The award committee will solicit the opinion of members of the research community as appropriate.
Theses supervised by members of the selection committee are not eligible.
The EATCS is committed to equal opportunities, and welcomes submissions of outstanding theses from all authors.

EATCS-Fellows 2017

Call for Nominations

Deadline: December 31, 2016

- VERY IMPORTANT: all nominees and nominators must be EATCS Members
- Proposals for Fellow consideration in 2017 should be submitted by DECEMBER 31st, 2016 by email to the EATCS Secretary (secretary@eatcs.org). The subject line of the email should read "EATCS Fellow Nomination - <surname of candidate>".

The EATCS Fellows Program is established by the Association to recognize outstanding EATCS Members for their scientific achievements in the field of Theoretical Computer Science. The Fellow status is conferred by the EATCS Fellows-Selection Committee upon a person having a track record of intellectual and organizational leadership within the EATCS community. Fellows are expected to be "model citizens" of the TCS community, helping to develop the standing of TCS beyond the frontiers of the community.

In order to be considered by the EATCS Fellows-Selection Committee, candidates must be nominated by at least four EATCS Members. Please verify your membership at www.eatcs.org.

The EATCS Fellows-Selection Committee consists of

- Rocco De Nicola (IMT Lucca, Italy)
- Paul Goldberg (Oxford, UK, chair)
- Anca Muscholl (Bordeaux, France)
- Dorothea Wagner (Karlsruhe, Germany)

- Roger Wattenhofer (ETH Zurich, Switzerland)

INSTRUCTIONS:

A nomination should consist of details on the items below. It can be co-signed by several EATCS members. At least two nomination letters per candidate are recommended. If you are supporting the nomination from within the candidateâĂŹs field of expertise, it is expected that you will be specific about the individualâĂŹs technical contributions.
To be considered, nominations for 2016 must be received by **December 31, 2016.**

1. Name of candidate Candidate's current affiliation and position Candidate's email address, postal address and phone number Nominator(s) relationship to the candidate
2. Short summary of candidate's accomplishments (citation – 25 words or less)
3. Candidate's accomplishments: Identify the most important contributions that qualify the candidate for the rank of EATCS Fellow according to the following two categories:

A) Technical achievements

B) Outstanding service to the TCS community

Please limit your comments to at most three pages.

4. Nominator(s):

Name(s)
Affiliation(s), email and postal address(es), phone number(s)
Please note: all nominees and nominators must be EATCS Members

Zoltán Ésik (1951–2016)
In Memoriam

Luca Aceto and Anna Ingólfsdóttir
ICE-TCS, School of Computer Science
Reykjavik University

Our friend and colleague Zoltán Ésik passed away in Reykjavik, Iceland, on Wednesday, 25 May 2016. He was visiting us as he did with some regularity, compatibly with his many engagements throughout the world.

The day before his untimely death, Zoltán had delivered an ICE-TCS seminar entitled *Equational Logic of Fixed Point Operations* at Reykjavik University. At the start of his talk, he looked somewhat tired and out of breath. However, the more he was presenting a research topic that he loved and that has kept him busy for most of his research career, the more he seemed to be feeling at ease. After the talk, we spent some time making plans for mutual visits in the autumn of 2016 and we discussed some EATCS-related matters. His wife Zsuzsa and he were due to spend a few days travelling in the north of Iceland before their return to Szeged, but life had other ideas.

Zoltán was a scientist of the highest calibre and has left behind a large body of deep and seminal work that will keep researchers in theoretical computer science busy for a long time to come. The list of refereed publications available from his web site at `http://www.inf.u-szeged.hu/~ze/classified.pdf` includes two books, 32 edited volumes, 135 journal papers, four book chapters, 86 conference papers and seven papers in other edited volumes. However, impressive as they undoubtedly are, these numbers give only a very partial picture of Zoltán's scientific stature. Together with the late Stephen Bloom, Zoltán was the prime mover in the monumental development of Iteration Theories. As Stephen and Zoltán wrote in the preface of their massive book on the topic, which was published in 1993 by Springer:

> Iteration plays a fundamental role in the theory of computation: for example, in the theory of automata, in formal language theory, in the study of formal power series, in the semantics of flowchart algorithms and programming languages, and in circular data type definitions. It is shown that in all structures that have been used as semantical models, the equational properties of the fixed point operation are captured

> by the axioms describing iteration theories. These structures include ordered algebras, partial functions, relations, finitary and infinitary regular languages, trees, synchronization trees, 2-categories, and others.

It is truly remarkable that the equational laws satisfied by fixed point operations are essentially the same in a large number of structures used in computer science. Isolating those laws, and showing their applicability, has been one of the goals of Zoltán's scientific life and we trust that the members of our community will keep reading his work on iteration theories, which continued and went from strength to strength after Stephen and he published their 600-page research monograph in 1993. During his last talk in Reykjavik, we asked Zoltán whether he was planning to write a new edition of that book, and half-jokingly told him that it would probably be about 1,200 pages.

Zoltán's research output includes contributions to automata theory, category theory, concurrency theory, formal languages, fuzzy sets and fuzzy logic, graph theory, logic in computer science, logic programming, order theory, semiring theory and universal algebra, amongst others. The breadth of research areas to which he has contributed bears witness to his amazing mathematical powers and to his curiosity. Wherever he went and no matter how long he had travelled to get there, Zoltán's brain was always open.

Zoltán also contributed to the research community with his service work and received several awards. Here we will limit ourselves to mentioning that he was elected member of the Academy of Europe in 2010, was named Fellow of the EATCS in 2016, was a member of the council of the EATCS from 2003 to 2015, and of the Presburger Award Committee in 2015–2016. He represented the Hungarian theoretical computer science community in the International Federation for Information Processing (IFIP) as member of TC1 since 2000 and was one of the prime mover in the establishment of the IFIP WG 1.8, Working Group on Concurrency. He also received the Gy. Farkas Research Award and the K. Rényi Research Award of the János Bolyai Mathematical Society.

Zoltán's appetite for work was phenomenal, but he also liked to have fun, to spend time with friends eating good food and drinking excellent wine, and to travel. Indeed, Zoltán's lust for travel was amazing. We lost track of his visits to myriads of research institutions and universities all over the world. He attended conferences in the most remote locations and always made sure that he would reserve some time for enjoying the most beautiful and known sites. At times, we had the feeling that he had been everywhere in the world.

Despite being often on the move, Zoltán was very much a family man. He was very proud of his wife Zsuzsanna, their daughter Eszter and their son Robert. He always told us about the latest developments in their lives and was happy about his four grandchildren. We had the pleasure of enjoying Zsuzsanna and Zoltán's exquisite hospitality both in Szeged and in their summer home on Lake Balaton.

Zoltán was very loyal to his friends and would make trips to see them wherever they were living. We were lucky to be amongst them and had the pleasure of hosting him in Aalborg, Florence and Reykjavik, where he visited us a few times and where the thread of his life was cut. We will miss the time we spent doing research or relaxing together, his sense of humour, his conviviality and his hospitality.

Obituary

Solomon Marcus, 1925–2016

The renowned Romanian mathematician and theoretical computer scientist, one of the initiators of mathematical linguistics and mathematical poetics, passed away in Bucharest on March 17, 2016. "Solomon Marcus' Heritage" Memorial Symposium was organised as part of the DCFS2016 Conference on 5 July 2016.

Solomon Marcus was born on March 1, 1925 in the city of Bacău, Romania, the youngest of the eight children of Sima and Alter Marcus.

In 1944 he graduated from Ferdinand I High School in Bacău. Marcus high school results were rather fluctuant: he had to repeat one year, ironically, because he failed to fully recite a poem, but was classified first at "Bacalaureat" (the Romanian school-leaving exam) in Moldavia county. "Ironically", because literature, especially poetry and theatre, have been his lasting passions; till the very end of his life he was able to recite many poems, in Romanian, French and German.

His exposer to non-Euclidean geometries in 1944 stimulated his life-long interest for mathematics. He completed his studies in mathematics at the University of Bucharest: BSc (with Merit) in 1949, PhD in 1956 (with the Thesis

Monotonous Functions of Two Variables under the supervision of M. Nicolescu) and State Doctor in Sciences in 1968.

Marcus started his academic career at the University of Bucharest – an institution he was affiliated with for more than 70 years – as a lecturer in 1955; he continued as professor in 1966 and emeritus professor from 1991.

Marcus has important contributions to pure mathematics. In the first ten years of his research career he published almost 100 papers in mathematical analysis, set theory, measure and integration theory and topology, including the joint paper with P. Erdös, Sur la décomposition de l'espace Euclidien en ensembles homogènes, *Acta Mathematica Academiae Scientiarum Hungaricae* vol. 8, nr. 3–4 (1957), 443–452.

From 1964 he published many papers and books in theoretical computer science, linguistics, poetics and theory of literature, semiotics, cultural anthropology, biology, history and philosophy of science, and education. In these fields he published more than 45 books in Romanian, English, French, German, Italian, Spanish, Russian, Greek, Hungarian, Czech, Serbo-Croatian, and more than 300 research articles. His book *Grammars and Finite Automata*, Ed. Academiei, Bucharest, 1964 (in Romanian) is arguably the first monograph devoted to regular languages. He authored three pioneering books in mathematical linguistics and poetics: *Introduction mathématique à la linguistique structurelle*, Dunod, Paris, 1967, *Algebraic Linguistics; Analytical Models*, Academic Press, New York, 1967, and *Mathematische Poetik*, Ed. Academiei & Athenaum Verlagare, 1973. The book S. Marcus, *Words and Languages Everywhere*, Polimetrica, Milano, 2007, includes a collection of his papers in language theories.

Writing was a passion for Marcus. We have to add to the above impressive list more than a hundred hand-written (mostly mathematical) notebooks and hundreds of articles and interviews, scattered in various magazines and newspapers, on a variety of topics, from science, art and education to teaching, philosophy and sport.

He is cited by more than a thousand authors, including mathematicians, computer scientists, linguists, literary researchers, semioticians, anthropologists and philosophers. Complementary, but still partial, information can be found on his websites at the Institute of Mathematics of the Romanian Academy and the University of Bucharest.

Marcus is featured in the books *People and Ideas in Theoretical Computer Science*, Springer, Singapore, 1998 and *The Human Face of Computing*, Imperial College Press, London, 2015 (edited by C. Calude). The 1500-page book dedicated to his 85th birthday, *Meetings with Solomon Marcus*, Spandugino House, Bucharest, 2010 (edited by L. Spandonide and G. Păun) includes his autobiography and articles by several hundred people. His 90th birthday celebrations at the University of Bucharest and Romanian Academy have been national academic events in 2015.

Marcus was elected a member of the Romanian Academy in 2001; he received many prizes including the *Royal Decoration of Nihil Sine Deo* (2011) and the *Star of Romania* in the rank of officer, awarded by the Romanian President (2015).

He was never bored, he never felt the need of a proper vacation: in our last telephonic conversation, when he was in hospital, his strongest desire was to continue to work. Unfortunately, he left behind many unfinished projects.

In his long career he inspired, stimulated, encouraged and advised many students, undergraduate and graduate, in Romania and abroad, to do research; his 16 PhD students form just a part of this group. I was extremely privileged to be his undergraduate student (real analysis in 1972–3 and mathematical linguistics in 1974–5) and graduate student (PhD, 1976), collaborator and friend. He was for me an inspiration and a role model. Nobody else had more influence on my scientific career than Professor Marcus. In the last 24 years we have been separated by more than 18,000 Km, but this distance didn't alter our close relation and collaboration.

He will be sadly missed and never forgotten.

Cristian S. Calude
Auckland, New Zealand

Boris (Boaz) Trakhtenbrot
1921–2016

Boris (Boaz) Abramovich Trakhtenbrot (Борис Абрамович Трахтенброт; בועזטרכטנברוט), a founding father of computer science, passed away September 19, 2016 at age 95, in Rehovot, Israel. His beloved wife, Berta (née Rabinovich), died three years prior. He is survived by two sons, Mark Trakhtenbrot and Yosef Halakhmi, five grandchildren, and two great-grandchildren.

Trakhtenbrot was born in Brichevo, a shtetl in Northern Bessarabia (now Moldova), about which he always spoke fondly. He studied at the Moldavian Pedagogical Institute in Kishinev, Chernivtsi National University (Ukraine), Kiev Mathematical Institute (Ukraine), and (unoÿcially) at Moscow Univer-sity. After completing his doctorate in 1950, under Petr S. Novikov, he took a position at the Belinsky Pedagogical Institute in Penza (Western Russia), and later—in 1960—joined the just-established Mathematical Institute at Novosi-birsk Akademgorodok, where he established and headed the Theory of Automata and Mathematical Linguistics Department. He received a Doctor of Sciences degree in 1962.

During the Stalin era, Boaz had troubles as many other scientists in the USSR. He was barred from attending international congresses in the west, to which he had been invited.

In 1980, Boaz immigrated to Israel and joined Tel Aviv University's School of Mathematical Sciences. There he was instrumental in the major growth phase of its computer science department. He remained vitally active for many years after his oÿcial retirement in 1991.

Today, Boaz is universally admired as a founding father and long-standing pillar of the discipline of computer science. He was the field's pre-eminent distinguished researcher, and a most illustrious trailblazer and disseminator. He made very many deeply significant contributions to theoretical computer science, on decidability problems in logic, finite automata theory, the connection between automata and monadic second-order logic, complexity of algorithms, abstract complexity, algorithmic logic, probabilistic computation, program verification, the lambda calculus and foundations of programming languages, programming semantics, semantics and methodology for concurrency, networks, hybrid systems, and more. He was unmatched in combining farsighted vision, unfaltering commitment, masterful command of the field, technical virtuoso, æsthetic expression, eloquent clarity, and creative vigor with humility and devotion to students and colleagues.

No fewer than three famous theorems in theoretical computer science bear Trakhtenbrot's name:

- **Trakhtenbrot's Theorem** (1950): *The validity of (first-order) statements that hold true for all finite universes is undecidable.*
- **The Büchi-Elgot-Trakhtenbrot Theorem** (1962): *Finite automata and weak monadic (second-order) logic have the same expressive power.*
- **The Borodin-Trakhtenbrot Gap Theorem** (1964): *There are arbitrarily large (computable) gaps in the hierarchy of complexity classes.*

Trakhtenbrot's doctoral dissertation inaugurated finite model theory. His subsequent Novosibirsk period was very productive; his regular seminar there was legendary. He introduced the use of monadic second-order logic as a spec-ification formalism for the infinite behavior of finite automata. This logic has turned out to be very fundamental; various temporal logics are just "sugared" fragments of monadic logic. And he was among the very first to consider time and space eÿciency of algorithms (using what he called "signalizing functions") and to speak about abstract complexity measures (independently and in parallel with similar developments by Western theoreticians).

Trakhtenbrot initiated the study of topological aspects of ω-languages and operators and provided a characterization of operators computable by finite automata. Furthermore, he supplied solutions to special cases of the Church synthesis problem, later solved by Büchi and Landweber. The equivalence with automata and the solvability of Church's problem laid the necessary under-pinnings for the development of formalisms for describing interactive systems and their properties. These have led to tools for algorithmic verification and automatic synthesis of correct implementations and for the advanced algorith-mic techniques that are now embodied in industrial tools for verification and validation.

His justly famous and truly elegant Gap Theorem (proved independently by Allan Borodin in the West) and his development of the "crossing sequence" method were groundbreaking. His paper on "auto-reducibility" provided a turn-ing point in abstract complexity. In the USSR, these works quickly became very influential, and, in the US, complexity took over as the central preoccupation of theoretical computer science.

Trakhtenbrot was at the same time a master pedagogue and expositor. His book, *Algorithms and Automatic Computing Machines*, first written in Russian in 1957, was translated into English and a dozen other languages, and is recog-nized worldwide as the first important text in the field. He played the key rôle in the dissemination of Soviet computer science research in the West, writing surveys on such topics as Soviet approaches to brute force search (*perebor*).

His later works dealt with various aspects of concurrency, including data flow networks, Petri nets, partial-order versus branching-time equivalence, bisimulation, real-time automata, and hybrid systems. All told, he published some one hundred articles, books, and monographs.

A roll call of Trakhtenbrot's students reads like the "Who's Who" of theoret-ical computer science in the USSR. His sixteen doctoral students are: Miroslav Kratko (1964), Nikolai Beljakin (1964), Janis Barzdins (1965), Valery Nepom-nyaschy (1967), Alexei Korshunov (1967), Mars K. Valiev (1969), Valery Aga-fonov (1969), Djavkathodja Hodjaev (1970), Zoya Litvintseva (1970), Rūsiņš Freivalds (1972), Anatoli Vaisser (1976), Vladimir Sazonov (1976), Michael Dekhtyar (1977), Irina Lomazova (1981), Alex Barel (1984), and Alexander Rabinovich (1989). Moreover, a whole generation of computer scientists was shaped by his textbooks on automata theory. Besides building the computer science department in Novosibirsk, he collaborated with computer designers in the Soviet Union and helped in the establishment of a department of theoretical informatics in Jena (East Germany). The Latvian school of computer science flourished under the tutelage of his students, Barzdins and Freivalds.

Trakhtenbrot received numerous prizes and recognitions for his contributions, including the following:

- In June 1991, Tel Aviv University's Department of Computer Science organized "An International Symposium on Theoretical Computer Science in honor of Boris A. Trakhtenbrot on the occasion of his Retirement and Seventieth Birthday". The event took place in Tel Aviv, and many of the world's foremost scientists gathered. (See the report by Val Breazu-Tannen in *SIGACT News*, vol. 22, no. 4, Fall 1991, pp. 27–32, http://portalparts.acm.org/130000/126546/fm/frontmatter.pdf.)

- In the same year, Trakhtenbrot's colleagues and former students from Latvia published a volume, "Dedicated to Professor B. A. Trakhtenbrot, father of Baltic Computer Science, on the occasion of his 70th birthday" (*Baltic Computer Science*, *Lecture Notes in Computer Science*, vol. 502, Springer-Verlag, May 1991).

- The Friedrich Schiller University in Jena bestowed on him the degree of doctor *honoris causa* in October 1997.

- At the *Computer Science Logic (CSL)* conference in Brno in 1998, a special session was organized to celebrate "50 years of Trakhtenbrot's Theorem", in which Boaz took part.

- In July 2001, in honor of his eightieth birthday and his "very important contribution to Formal Languages and Automata", Trakhtenbrot gave the keynote address at the joint session of the *International EATCS Colloquium on Automata, Languages and Programming (ICALP)* and of the *ACM Symposium on Theory of Computing (SIGACT)*, held in Crete (http://acm-stoc.org/stoc2001).

- In 2006, the School of Computer Science at Tel Aviv University held a "Computation Day Celebrating Boaz (Boris) Trakhtenbrot's Eighty-Fifth Birthday" (http://www.cs.tau.ac.il/~nachumd/Boaz.html).

- In 2008, the volume, *Pillars of Computer Science: Essays Dedicated to Boris (Boaz) Trakhtenbrot on the Occasion of His 85th Birthday* appeared in Springer's Festschrift series (*Lecture Notes in Computer Science*, vol. 4800, Springer-Verlag, 2008), with 34 scientific contributions by his friends and colleagues, themselves leading mathematicians, logicians, and com-puter scientists.

- In 2011, the *European Association for Theoretical Computer Science (EATCS)* honored him with their highest award, the Distinguished Achievements Award. The laudation (https://www.eatcs.org/images/awards/LAUDATIO2011.pdf) reads:

  > For over half a century, Trakhtenbrot has been making seminal contributions to virtually all of the central aspects of theoreti-cal computer science, inaugurating numerous brand-new areas of investigation. . . . The entire body of his work demonstrates the same unique melding of supreme mathematical prowess, combined with profound depth and thoroughness. His operative style has always been patient in-depth survey of existing literature, uncompromising evaluation and critical comparison of existing approaches, followed by extraordinary and prescient contributions.

Trakhtenbrot's contributions are astounding under any measure; how much more so when consideration is given to the fact that he worked under very adverse conditions: persecution, lack of support, almost no access to foreign meetings, and so on. His undaunted spirit should serve as an inspiration to all.

His wisdom, courage, and generosity will be sorely missed.

IN MEMORIAM
BORIS TRAKHTENBROT, 1921–2016

Lawrence M. Fisher

Boris (Boaz) Avraamovich Trakhtenbrot, an Israeli and Russian mathematician who worked in the areas of mathematical logic, algorithms, theory of computation and cybernetics, passed away on September 19 at the age of 95.

Trakhtenbrot was born on February 20, 1921 in Bricheva, a village in Northern Bessarabia (now known as Moldova). He attended high school in the neighboring towns of Belts and Soroka, and in 1940 enrolled in the Faculty of Physics and Mathematics of the Moldovian Pedagogical Institute in Kishinev. After than city was bombed at the outset of World War II in June 1941, he relocated to Chkalov (now Orenburg), and enrolled in the local pedagogical institute. In 1942, he and his family move to Buguruslan, but his studies were often interrupted during the war years.

In 1945, having passed the exams to qualify to teach high school math, Trakhtenbrot enrolled in the University of Chernovsty (Ukraine) to pursue the equivalent of a master's degree, which he received in 1947. He spent the next three years studying mathematical logic and computability at the Kiev Mathematics Institute of the Ukrainian Academy of Sciences, where he received his Ph.D. in 1950. He also received a (Soviet) Doctor of Sciences degree in 1962.

From 1950 to 1960, Trakhtenbrot held positions at the Pedagogical and Polytechnic Institutes of Penza, Russia.

From 1960 to 1980, he conducted research at the Mathematical Institute of the USSR Academy of Sciences' Siberian branch, and lectured at Novosibirsk State University in Akademgorodok, Novosibirsk. He joined Novosibirsk's Department of Theoretical Cybernetics when it was launched in 1961, and worked there with J.M. Barzdin on the basic concepts of computational complexity.

After immigrating to Israel in 1981, he became a professor of computer science at Tel Aviv University. In the years before his retirement in 1991, he also visited and collaborated with many Western universities and research centers.

In 1964, Trakhtenbrot discovered and proved a fundamental result in theoretical computer science called the Gap theorem, regarding the complexity of computable functions. He also discovered and proved what is now called Trakhtenbrot's theorem, which states the problem of validity in first-order logic (FO) on the class of all finite models is undecidable, and that the class of valid sentences over finite models is not recursively enumerable (though it is co-recursively enumerable). The theorem was first published in 1950, in the paper "The Impossibility of an Algorithm for the Decidability Problem on Finite Classes."

In 2011, the European Association for Theoretical Computer Science (EATCS) awarded Trakhtenbrot, then about to turn 90, its annual Distinguished Achievements Award. The organization described him as "unquestionably a principal founding father of the discipline of computer science, a preeminent distinguished researcher, and a most illustrious leader and disseminator," as well as "a grand visionary who pioneered many fascinating directions and concepts, which have had enormous impact."

Further, the organization said, "for over half a century, Trakhtenbrot has been making seminal contributions to virtually all of the central aspects of theoretical computer science, inaugurating numerous brand-new areas of investigation. The list of topics in which Trakhtenbrot has made his lasting mark is breathtaking in its scope: decidability problems in logic and schematology of programs, finite automata theory, the connection between infinite automata and monadic second-order logic, complexity of algorithms, abstract complexity, algorithmic logic, probabilistic computation, program verification, the lambda calculus and foundations of programming languages, programming semantics, and much more. The entire body of his work demonstrates the same unique melding of supreme mathematical prowess, combined with profound depth and thoroughness. His operative style has always been patient in-depth survey of existing literature, uncompromising evaluation and critical comparison of existing approaches, followed by extraordinary and prescient contributions.

...Trakhtenbrot's contributions are astounding under any measure; his undaunted spirit should be heralded as an inspiration to the rest of the world."

Trakhtenbrot published about 100 papers and four books:

- Introduction to the Theory of Finite Automata (co-authored with N.E. Kobrinski);
- Algorithms and Automatic Computing Machines;
- Complexity of Algorithms and Computations, and
- Finite Automata: Behavior and Synthesis (co-authored with J.M. Barzdin).

He married Berta I. Rabinovich in 1947; she passed away in 2013. They had two sons, Mark (who was part of a team that received the ACM Software System Award for 2007) and Yossef, and five grandchildren.

About the author Lawrence M. Fisher is Senior Editor/News for ACM Magazines.

Acknowledgement This obituary appeared in ACM News on 21 September 2016. It is reprinted here with the kind permission of the ACM.

Institutional Sponsors

CTI, Computer Technology Institute & Press "Diophantus"
Patras, Greece

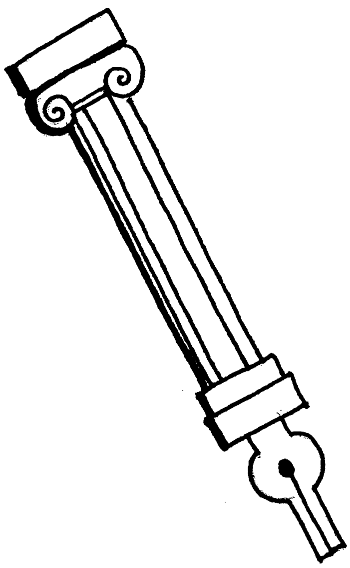
CWI, Centum Wiskunde & Informatica
Amsterdam, The Netherlands

MADALGO, Center for Massive Data Algorithmics
Aarhus, Denmark

Microsoft Research Cambridge
Cambridge, United Kingdom

Springer-Verlag
Heidelberg, Germany

EATCS
Columns



The Algorithmics Column

by

Gerhard J Woeginger

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

Chaining introduction with some computer science applications

Jelani Nelson*

Contents

*Harvard University. `minilek@seas.harvard.edu`. Supported by NSF CAREER award CCF-1350670, NSF grant IIS-1447471, ONR Young Investigator award N00014-15-1-2388, and a Google Faculty Research Award.

1 What is chaining?

Consider the problem of bounding the maximum of a collection of random variables. That is, we have some collection $(X_t)_{t\in T}$ and want to bound $\mathbb{E}\sup_{t\in T} X_t$, or perhaps we want to say this sup is small with high probability (which can be achieved by bounding $\mathbb{E}\sup_{t\in T}|X_t|^p$ for large p and applying Markov's inequality).

Such problems show up all the time in probabilistic analyses, including in computer science, and the most common approach is to combine tail bounds with union bounds. For example, to show that the maximum load when throwing n balls into n bins is $O(\log n/\log\log n)$, one defines X_t as the load in bin t, proves $\mathbb{P}(X_t > C\log n/\log\log n) \ll 1/n$, then performs a union bound to bound $\sup_t X_t$. Or when analyzing the update time of a randomized data structure on some sequence of operations, one argues that no operation takes too much time by understanding the tail behavior of X_t being the time to perform operation t, then again performs a union bound to control $\sup_t X_t$.

Most succinctly, chaining methods leverage statistical dependencies between a (possibly infinite) collection of random variables *to beat this naive union bound*.

The origins of chaining began with Kolmogorov's continuity theorem from the 1930s (see Section 2.2, Theorem 2.8 of [21]). The point of this theorem was to understand conditions under which a stochastic process is continuous. That is, consider a random function $f:\mathbb{R}\to X$ where (X,d) is a metric space. Assume the distribution over f satisfies the property that for some $\alpha,\beta>0$, $\mathbb{E}|f(x)-f(y)|^\alpha = O(|x-y|^{1+\beta})$ for all $x,y\in\mathbb{R}$. Kolmogorov proved that for any such distribution, one can couple with another distribution over functions $\tilde{f}$ such that $\forall x\in\mathbb{R}$, $\mathbb{P}(f(x)=\tilde{f}(x))=1$, and furthermore $\tilde{f}$ is continuous. For the reader interested in seeing proof details, see for example [29, Section A.2].

Since Kolmogorov's work, the scope of applications of the chaining methodology has widened tremendously, due to contributions of many mathematicians, including Dudley, Fernique, and very notably Talagrand. See Talagrand's treatise [29] for a description of many impressive applications of chaining in mathematics. See also Talagrand's STOC 2010 paper [28]. Note that [29] is not exhaustive, and additional applications are posted on the arXiv on a regular basis.

2 Applications in computer science

Several applications are given in [30, Section 1.2.2]. I will repeat some of those here, as well as some other ones.

2.1 Random matrices and compressed sensing

Consider a random matrix $M \in \mathbb{R}^{m\times n}$ from some distribution. A common task is to understand the behavior of the largest singular value of M. Note $\|M\| = \sup_{\|x\|_2=\|y\|_2=1} x^T My$, so the goal is to understand the supremum of the random variables $X_t = t_1^T Mt_2$ for $t \in T = B_{\ell_2^m} \times B_{\ell_2^n}$. Indeed, for many distributions one can obtain asymptotically sharp results via chaining.

Understanding singular values of random matrices has been important in several areas of computer science. Close to my own heart are in compressed sensing and randomized linear algebra algorithms. For the latter, a relevant object is a *subspace embedding*; these are objects used in algorithms for fast regression, low-rank approximation, and a dozen other applications (see [31]). Analyses then boil down to understanding the largest singular value of $M = (\Pi U)^T(\Pi U) - I$. In compressed sensing, where the goal is to approximately recover a nearly sparse signal x from few linear measurements Sx (the measurements are put as rows of the matrix S), analyses again boil down to bounding the operator norm of the same M, but for all U simultaneously that can be formed from choosing k columns from some basis that x is sparse in.

2.2 Empirical risk minimization

This example is taken from [30]. In machine learning one often is given some data, drawn from some unknown distribution, and a loss function $\mathcal{L}$. Given some family of distributions parameterized by some $\theta \in \Theta$, the goal is to find some θ^* which explains the data the best, i.e.

$$\theta^* = \underset{\theta\in\Theta}{argmin}\, \mathbb{E}\, \mathcal{L}(\theta, X). \tag{1}$$

The expectation is taken over the distribution of X. We do not know X, however, and only have i.i.d. samples $X_1, \ldots, X_n$. Thus a common proxy is to calculate

$$\hat{\theta} = \underset{\theta\in\Theta}{argmin}\, \frac{1}{n}\sum_{k=1}^{n} \mathcal{L}(\theta, X_k).$$

We would like to argue that $\hat{\theta}$ is a nearly optimal minimizer for the actual problem (1). For this to be true, it is sufficient that $\sup_\theta X_\theta$ is small, where one ranges over all $\theta \in \Theta$ with

$$X_\theta = \left|\frac{1}{n}\sum_{k=1}^{n} \mathcal{L}(\theta, X_k) - \mathbb{E}\,\mathcal{L}(\theta, X)\right|.$$

2.3 Dimensionality reduction

In Euclidean dimensionality reduction, such as in the Johnson-Lindenstrauss lemma, one is given a set of vectors $P \subset \ell_2^n$, and wants that a (usually random) matrix Π satisfies

$$\forall y, z \in P,\ (1-\varepsilon)\|y-z\|_2^2 \le \|\Pi y - \Pi z\|_2^2 \le (1+\varepsilon)\|y-z\|_2^2. \tag{2}$$

This is satisfied as long as $\sup_{y,z} X_{y,z} \le \varepsilon$, where

$$X_{y,z} = |\frac{1}{\|y-z\|_2^2}\|\Pi y - \Pi z\|_2^2 - 1|,$$

where y, z ranges over all pairs of distinct vectors in P. Gordon's theorem [15] states that a Π with i.i.d. gaussian entries ensures this with good probability as long as it has $\gtrsim (g^2(T)+1)/\varepsilon^2$ rows, where $g(T)$ is the *gaussian mean width* of T and T is the set of normalized differences of vectors in P. Later works gave sharper analysis, and also extended to other types of Π, all using chaining [19, 24, 2, 6, 9, 25].

Another application of chaining in the context dimensionality reduction was in regard to nearest neighbor (NN) preserving embeddings [17]. In this problem, one is given a database $X \subset \ell_2^d$ of n points and must create a data structure such that for any query point $q \in \mathbb{R}^d$, one can quickly find a point $x \in X$ such that $\|q - x\|_2$ is nearly minimized. Of course, if *all* distances are preserved between q and points in X, this suffices to accomplish our goal, but it is more powerful than what is needed. It is only needed that the distance from q to its nearest neighbor does not increase too much, and that the distances from q to much farther points do not shrink too much (to fool us into thinking that they are approximate nearest neighbors). An embedding satisfying such criteria is known as a *NN-preserving embedding*, and [17] used chaining methods to show that certain "nice" sets X have such embeddings into low dimension. Specifically, the target dimension can be $O(\Delta^2\varepsilon^{-2}\frac{\gamma_2(X)}{diam(X)})^2$, where Δ is the aspect ratio of the data and γ_2 is a functional defined by Talagrand (more on that later). All we will say now is that $\gamma_2(X)$ is always $O(\sqrt{\log \lambda_X})$, where λ_X is the doubling constant of X (the maximum number of balls of radius $r/2$ required to cover any radius-r ball, over all r).

2.4 Data structures and streaming algorithms

The potential example to data structures was already mentioned in the previous section. To make it more concrete, consider the following streaming data structural problem in which one sees a sequence $p_1, \ldots, p_m$ with each $p_k \in \{1, \ldots, n\}$. For example, when monitoring a search query stream, p_k may be a word in a dictionary of size n. The goal of the *heavy hitters* problem is to identify words that occur frequently in the stream. Specifically, if we let f_i be the number of occurrences of $i \in [n]$ in the stream, in the ℓ_2 heavy hitters problem the goal is to find all i such that $f_i^2 \geq \varepsilon \sum_i f_i^2$ (think of ε as some given constant). The CountSketch of Charikar, Chen, and Farach-Colton solves this problem using $O(\log n)$ machine words of memory.

A recent work of [5] provides a new algorithm that solves the same problem using only $O(\log \log n)$ words of memory, and even more recently it has been shown how to achieve the optimal $O(1)$ words of memory [4]. These are randomized algorithms that maintain certain random variables in memory that evolve over time, and their analyses require controlling the largest of their deviations. Without getting into technical details here, we describe a related streaming problem: ℓ_2 estimation. The goal here is to use small memory while, after any query, being able to output an estimate Q satisfying $\mathbb{P}(|Q - \|f\|_2| > \varepsilon\|f\|_2) < 1/3$ (the probability is over the randomness used by the algorithm). It turns out this problem can be solved in $O(1/\varepsilon^2)$ words of memory by a randomized data structure known as the "AMS sketch" [3]. The failure probability can be decreased to δ by running $\Theta(\log(1/\delta))$ instantiations of the algorithm in parallel with independent randomness, then returning the median estimate of $\|f\|_2$ during a query. This yields space $O(\varepsilon^{-2} \log(1/\delta))$ words, which is optimal [18].

Recently the following question has been studied: what if we want to track $\|f\|_2$ *at all times*? Recalling the stream contains m updates, one could do as above and set $\delta < 3/m$ and union bound, so with an $O(\varepsilon^{-2} \log m)$-space algorithm, with probability $2/3$ all queries throughout the entire stream are correct. The work [16] showed this bound can be asymptotically improved when the number of distinct indices in the stream and $1/\varepsilon$ are both subpolynomial in m. This restriction was removed in subsequent works [5, 4].

2.5 Random walks on graphs

Ding, Lee, and Peres [11] a few years ago gave the first *deterministic* constant-factor approximation algorithm to the cover time of a random graph. Their work showed that the cover time of any connected graph is, up to a constant, equal to the supremum of a certain collection of random variables depending on that graph: the *gaussian free field*. This is a collection of gaussian random variables whose covariance structure is given by the effective resistances between the graph's vertices. Work of Talagrand (the "majorizing measures theory") and Fernique have provided us with tight, up to a constant factor, upper and lower bounds for the expected supremum of a collection of random variables. Furthermore, these bounds are constructive and efficient. See also the works [23, 8, 32] for more on this topic.

2.6 Dictionary learning

In *dictionary learning* one assumes that some data of p samples, the columns of some matrix $Y \in \mathbb{R}^{n \times p}$, is (approximately) sparse in some unknown "dictionary". That is, $Y = AX + E$ where A is unknown, X is sparse in each column, and E is an error matrix. If $E = 0$, A is square, and X has i.i.d. entries with s expected non-zeroes per column, with the non-zeroes being subgaussian, then Spielman, Wang, and Wright gave the first polynomial-time algorithm which provably recovers A (up to permutation and scaling of its columns) using polynomially many samples. Their proof required $O(n^2 \log^2 n)$ samples, but they conjectured $O(n \log n)$ should suffice.

It was recently shown that their precise algorithm needs roughly n^2 samples, but $O(n \log n)$ does suffice for a slight variant of their algorithm. As per [27], the analysis of the latter result boiled down to bounding the supremum of a collection of random variables. See [22, 1, 7].

2.7 Error-correcting codes

A q-ary linear error-correcting code C is such that the codewords are all vectors of the form xM for some row vector $x \in \mathbb{F}_q^m$ and $M \in \mathbb{F}_q^{m \times n}$. M is called the "generator matrix". Such a code is *list-decodable* up to some radius R, if, informally, if one arbitrarily corrupts any codeword C in at most an R-fraction of coordinates to obtain some C', then the *list* of candidate codewords in C which could have arisen in this way (i.e. are within radius R of C') is small.

Recent work of Rudra and Wootters [26] showed, to quote them, that "any q-ary code with sufficiently good distance can be randomly punctured to obtain, with high probability, a code that is list decodable up to radius $1-1/q-\varepsilon$ with near-optimal rate and list sizes". A "random puncturing" means simply to randomly sample some number of columns of M to form a random matrix M', which is the generator matrix for the new "punctured" code. Their proof relies on chaining.

In the remainder, we show the details of how chaining works, we play with a toy example (bounding the gaussian mean width of the ℓ_1 ball in $\mathbb{R}^n$), then describe an application of chaining to a real computer science problem: Euclidean dimensionality reduction.

3 A case study: (sub)gaussian processes

To give an introduction to chaining, I will focus our attention on a concrete scenario. Suppose we have a bounded (but possibly infinite) collection of vectors

$T \subset \mathbb{R}^n$. Furthermore, let $g \in \mathbb{R}^n$ be a random vector with its entries being independent, mean zero, and unit variance gaussians. We will consider the collection of variables $(X_t)_{t\in T}$ with X_t defined as $\langle g, t\rangle$. In what follows, we will only ever use one property of these X_t:

$$\forall s, t \in T,\ \mathbb{P}(|X_s - X_t| > \lambda) \lesssim e^{-\lambda^2/(2\|s-t\|_2^2)}. \tag{3}$$

This provides us with some understanding of the dependency structure of the X_t. In particular, if s, t are close in ℓ_2, then it's very likely that the random variables X_s and X_t are also close.

Why does this property hold? Well,

$$X_s - X_t = \langle g, s - t\rangle = \sum_{i=1}^{n} g_i \cdot (s - t)_i.$$

We then use the property that adding independent gaussians yields a gaussian in which the variances add. If you haven't seen that fact before, it follows easily from looking at the Fourier transform of the gaussian pdf. Adding independent random variables convolves their pdfs, which pointwise multiplies their Fourier transforms. Since the Fourier transform of a gaussian pdf is a gaussian whose variance is inverted, it then follows that summing independent gaussians gives a gaussian with summed variances. Thus $X_s - X_t$ is a gaussian with variance $\|s-t\|_2^2$, and (3) then follows by tail behavior of gaussians. Note (3) would hold for subgaussian distributions too, such as for example g being a vector of independent uniform ±1 random variables.

Now I will present four approaches to bounding $g(T) := \mathbb{E}_g \sup_{t\in T} X_t$. These approaches will be gradually sharper. For simplicity I will assume $|T| < \infty$, although it is easy to circumvent this assumption for methods 2, 3, and 4.

3.1 Method 1: union bound

Remember that, in general for a scalar random variable Z,

$$\mathbb{E}\,|Z| = \int_0^\infty \mathbb{P}(Z > u)du.$$

Let $\rho_X(T)$ denote the diameter of T under norm X. Then

$$\begin{aligned}
\mathbb{E}\sup_{t\in T} X_t &= \int_0^\infty \mathbb{P}(\sup_{t\in T} X_t > u)du \\
&\leq \int_0^{2\rho_{\ell_2}(T)\sqrt{2\log|T|}} \overbrace{\mathbb{P}(\sup_{t\in T} X_t > u)}^{\leq 1} du + \int_{\rho_{\ell_2}(T)\sqrt{2\log|T|}} \mathbb{P}(\sup_{t\in T} X_t > u)du \\
&\leq \rho_{\ell_2}(T)\sqrt{2\log|T|} + \int_{\rho_{\ell_2}(T)\sqrt{2\log|T|}}^\infty \sum_{t\in T} \mathbb{P}(X_t > u)du \text{ (union bound)} \\
&\leq \rho_{\ell_2}(T)\sqrt{2\log|T|} + |T|\cdot \int_{\rho_{\ell_2}(T)\sqrt{2\log|T|}}^\infty e^{-u^2/(2\rho_{\ell_2}(T)^2)}du \\
&= \rho_{\ell_2}(T)\sqrt{2\log|T|} + \rho_{\ell_2}(T)\cdot|T|\cdot \int_{\sqrt{2\log|T|}}^\infty e^{-v^2/2}dv \text{ (change of variables)} \\
&\lesssim \rho_{\ell_2}(T)\cdot\sqrt{\log|T|} \qquad (4)
\end{aligned}$$

3.2 Method 2: ε-net

Let $T' \subseteq T$ be an ε-net of T under ℓ_2. That is, for all $t \in T$ there exists $t' \in T'$ such that $\|t - t'\|_2 \leq \varepsilon$. Now note $\langle g, t\rangle = \langle g, t' + (t - t')\rangle$ so that

$$X_t = X_{t'} + X_{t-t'}.$$

Therefore

$$g(T) \leq g(T') + \mathbb{E}\sup_{t\in T}\langle g, t - t'\rangle.$$

We already know $g(T') \lesssim \rho_{\ell_2}(T')\cdot\sqrt{\log|T'|} \leq \rho_{\ell_2}(T)\cdot\sqrt{\log|T'|}$ by (4). Also, $\langle g, t - t'\rangle \leq \|g\|_2\cdot\|t - t'\| \leq \varepsilon\|g\|_2$, and

$$\mathbb{E}\|g\|_2 \leq (\mathbb{E}\|g\|_2^2)^{1/2} \leq \sqrt{n}.$$

Therefore

$$\begin{aligned}
g(T) &\lesssim \rho_{\ell_2}(T)\cdot\sqrt{\log|T'|} + \varepsilon\sqrt{n} \\
&= \rho_{\ell_2}(T)\cdot\log^{1/2}\mathcal{N}(T, \ell_2, \varepsilon) + \varepsilon\sqrt{n} \qquad (5)
\end{aligned}$$

where $\mathcal{N}(T, d, u)$ denotes the *entropy number* or *covering number*, defined as the minimum number of radius-u balls under metric d centered at points in T required to cover T (i.e. the size of the smallest u-net). Of course ε can be chosen to minimize (5). Note the case $\varepsilon = 0$ just reduces back to method 1.

3.3 Method 3: Dudley's inequality (chaining)

The idea of Dudley's inequality [13] is to, rather than use one net, use a countably infinite sequence of nets. That is, let $S_r \subset T$ denote an ε_r-net of T under ℓ_2, where $\varepsilon_r = 2^{-r} \cdot \rho_{\ell_2}(T)$. Let t_r denote the closest point in S_r to some $t \in T$. Note $T_0 = \{0\}$ is a valid ε_0-net. Then

$$\langle g, t\rangle = \langle g, t_0\rangle + \sum_{r=1}^{\infty} \langle g, t_r - t_{r-1}\rangle,$$

so then

$$\begin{aligned} g(T) &\le \sum_{r=1}^{\infty} \mathbb{E} \sup_{t\in T} \langle g, t_r - t_{r-1}\rangle \\ &\lesssim \sum_{r=1}^{\infty} \frac{\rho_{\ell_2}(T)}{2^r} \cdot \log^{1/2}(\mathcal{N}(T, \ell_2, \frac{\rho_{\ell_2}(T)}{2^r})^2) \text{ (by (4))} \qquad (6) \\ &\lesssim \sum_{r=1}^{\infty} \frac{\rho_{\ell_2}(T)}{2^r} \cdot \log^{1/2} \mathcal{N}(T, \ell_2, \frac{\rho_{\ell_2}(T)}{2^r}) \qquad (7) \end{aligned}$$

where (6) used the triangle inequality to yield

$$\|t_r - t_{r-1}\|_2 \le \|t - t_r\|_2 + \|t - t_{r-1}\|_2 \le \frac{3}{2^r} \cdot \rho_{\ell_2}(T).$$

The sum (7) is perfectly fine as is, though the typical formulation of Dudley's inequality then bounds the sum by an integral over ε (representing $\rho_{\ell_2}(T)/2^r$) then performs the change of variable $u = \varepsilon/\rho_{\ell_2}(T)$. This yields the usual formulation of Dudley's inequality:

$$g(T) \lesssim \int_0^{\infty} \log^{1/2} \mathcal{N}(T, \ell_2, u) du \qquad (8)$$

It is worth pointing out that Dudley's inequality is equivalent to the following bound. We say $T_0 \subset T_1 \subset \ldots \subset T$ is an *admissible sequence* if $|T_0| = 1$ and $|T_r| \le 2^{2^r}$. Then Dudley's inequality is equivalent to the bound

$$g(T) \lesssim \sum_{r=0}^{\infty} 2^{r/2} \cdot \sup_{t\in T} d_{\ell_2}(t, T_r). \qquad (9)$$

To see this most easily, compare with the bound (7). Note that to minimize $\sup_{t\in T} d_{\ell_2}(t, T_r)$, we should pick the best quality net we can using 2^{2^r} points. From $r = 0$ until some r_1, the quality of the net will be, up to a factor of 2, equal to

$\rho_{\ell_2}(T)$, and for the r in this range the summands of (9) will be a geometric series that sum to $O(2^{r_1/2} \cdot \rho_{\ell_2}(T))$. Then from $r = r_1$ to some r_2, the quality of the best net will be, up to a factor of 2, equal to $\rho_{\ell_2}(T)/2$, and these summands then are a geometric series that sum to $O(2^{r_2/2} \cdot \rho_{\ell_2}(T)/2)$, etc. In this way, the bounds of (7) and (9) are equivalent up to a constant factor.

Note, this is the primary reason we chose the T_r to have doubly exponential size in r: so that the sum of $\log^{1/2} |T_r|$ in any contiguous range of r is a geometric series dominated by the last term.

3.4 Method 4: generic chaining

Here we will show the generic chaining method, which yields the bound of [14], though we will present an equivalent bound that was later given by Talagrand (see his book [29]):

$$g(T) \lesssim \inf_{\{T_r\}_{r=0}^{\infty}} \sup_{t \in T} \sum_{r=0}^{\infty} 2^{r/2} \cdot d_{\ell_2}(t, T_r). \tag{10}$$

where the infimum is taken over admissible sequences.

Note the similarity between (9) and (10): the latter bound moved the supremum *outside* the sum. Thus clearly the bound (10) can only be a tighter bound. For a metric d, Talagrand defined

$$\gamma_p(T, d) := \inf_{\{T_r\}_r} \sup_{t \in T} \sum_{r=0}^{\infty} 2^{r/p} \cdot d(t, T_r),$$

where again the infimum is over admissible sequences. We now we wish to prove

$$g(T) \lesssim \gamma_2(T, \ell_2).$$

You are probably guessing at this point that had we not been working with subgaussians, but rather random variables that have decay bounded by $e^{-|x|^p}$, we would get a bound in terms of the γ_p-functional — your guess is right. I leave it to you as an exercise to modify arguments appropriately!

For nonnegative integer r and for $t \in T$, define $\pi_r t = \operatorname{argmin}_{t' \in T_r} d(t, t')$. For $r \geq 1$ define $\Delta_r t = \pi_r t - \pi_{r-1} t$. Then for any $t \in T$

$$t = \pi_0 t + \sum_{r=1}^{\infty} \Delta_r t$$

so that

$$\mathbb{E} \sup_{t \in T} \langle g, t \rangle \quad = \quad \mathbb{E} \sup_{t \in T} \sum_{r=1}^{\infty} \underbrace{\langle g, \Delta_r t \rangle}_{Y_r(t)} .$$

since $\mathbb{E}\, sup_{t\in T} \langle g, \pi_0 t\rangle = \mathbb{E}\langle g, \pi_0 t\rangle = 0$, with the first equality using that $|T_0| = 1$.

Note for fixed t, by gaussian decay

$$\mathbb{P}(|Y_r(t)| > 2u2^{r/2}\|\Delta_r t\|) < 2e^{-u^2 2^r}.$$

Therefore

$$\begin{aligned}
\mathbb{P}(\exists t \in T, r > 0 \ s.t.\ |Y_r(t)| > 2u2^{r/2}\|\Delta_r t\|) &\lesssim \sum_{r=1}^{\infty} |T_r| \cdot |T_{r-1}| \cdot e^{-u^2 2^r} \\
&\leq \sum_{r=1}^{\infty} 4^{2^r} \cdot e^{-u^2 2^r} \qquad (11)
\end{aligned}$$

since $|T_r|, |T_{r-1}| \leq 2^{2^r}$. The above sum is convergent for $u \geq 2$.

Now, again using that $\mathbb{E}\,|Z| = \int_0^\infty \mathbb{P}(|Z| > w)dw$, we have

$$\begin{aligned}
g(T) &\leq \int_0^\infty \mathbb{P}(\sup_{t\in T} \sum_{r=1}^{\infty} Y_r > w)dw \\
&= \left(2 \sup_{t\in T} \sum_{r=1}^{\infty} 2^{r/2}\|\Delta_r t\|\right) \\
&\qquad \times \int_0^\infty \mathbb{P}(\sup_{t\in T} \sum_{r=1}^{\infty} Y_r > u \cdot 2 \sup_{t\in T} \sum_{r=1}^{\infty} 2^{r/2}\|\Delta_r t\|)du \text{ (change of variables)} \\
&\lesssim \left(\sup_{t\in T} \sum_{r=1}^{\infty} 2^{r/2}\|\Delta_r t\|\right) \\
&\qquad \times [2 + \int_2^\infty \mathbb{P}(\sup_{t\in T} \sum_{r=1}^{\infty} Y_r > u \cdot 2 \sup_{t\in T} \sum_{r=1}^{\infty} 2^{r/2}\|\Delta_r t\|)du] \\
&\lesssim \left(\sup_{t\in T} \sum_{r=1}^{\infty} 2^{r/2}\|\Delta_r t\|\right) \\
&\qquad \times [2 + \int_2^\infty \mathbb{P}(\exists t \in T, r > 0 \ s.t.\ |Y_r(t)| > 2u2^{r/2}\|\Delta_r t\|)du] \\
&\lesssim \sup_{t\in T} \sum_{r=1}^{\infty} 2^{r/2}\|\Delta_r t\|. \qquad (12)
\end{aligned}$$

Now note $\|\Delta_r t\| = \|t_r - t_{r-1}\| \leq 2d_{\ell_2}(t, T_r)$ by the triangle inequality, and thus (12) is at most a constant factor larger than $\gamma_2(T, \ell_2)$, as desired.

Surprisingly, Talagrand showed that not only is $\gamma_2(T, \ell_2)$ an asymptotic upper bound for $g(T)$, but it is also an asymptotic *lower bound* (at least when the entries of g are *actually* gaussians — the lower bound does not hold for subgaussian entries). That is, $g(T) \simeq \gamma_2(T, \ell_2)$ for *any* T. This is known as the "majorizing measures theorem" for reasons we will not get into. In brief: the formulation of [14] did not talk about admissible sequences, or discrete sets at all, but rather worked with measures and provided an upper bound in terms of an infimum over a set of probability measures of a certain integral — this formulation is equivalent to the formulation discussed above in terms of admissible sets, and a proof of the equivalence appears in [29].

4 A concrete example: the ℓ_1 ball

Consider the example $T = B_{\ell_1^n} = \{t \in \mathbb{R}^n : \|t\|_1 = 1\}$, i.e. the unit ℓ_1. I picked this example because it is easy to already know $g(T)$ using other methods. Why? Well, $\sup_{t \in B_{\ell_1^n}} \langle g, t \rangle = \|g\|_\infty$, since the dual norm of ℓ_∞ is ℓ_1! Thus $g(B_{\ell_1^n}) = \mathbb{E}\,\|g\|_\infty$, which one can check is $\Theta(\sqrt{\log n})$. Thus we *know* the answer is $\Theta(\sqrt{\log n})$.

So now the question: what do the four methods above give?

4.1 Method 1: union bound

This method gives nothing, since T is an infinite set.

4.2 Method 2: ε-net

To apply this method, we need to understand the size of an ε-net of the ℓ_1 unit ball under ℓ_2. One bound comes from Maurey's empirical method.

Lemma 1 (Maurey's empirical method)**.** $\mathcal{N}(B_{\ell_1^n}, \ell_2, u) \le (2n)^{4/u^2}$

Proof. Consider any $t \in B_{\ell_1^n}$. It can be written as a convex combination $t = \sum_{i=1}^{2n} \alpha_i x_i$ where $x_1, \ldots, x_n = e_1, \ldots, e_n$ and $x_{n+1}, \ldots, x_{2n} = -e_1, \ldots, -e_n$. Now, consider a distribution over $\mathbb{R}^n$ in which we pick a random vector v which equals t_i with probability α_i. Then $\mathbb{E}\,v = t$. Now pick $Z_1, \ldots, Z_q, Z'_1, \ldots, Z'_q$ i.i.d. from this distribution. Define the vectors $Z = (Z_1, \ldots, Z_q)$ and $Z' = (Z'_1, \ldots, Z'_q)$. Then

$$\begin{aligned}
\mathop{\mathbb{E}}_{Z} \|t - \frac{1}{q}\sum_{i=1}^{q} Z_i\|_2 &= \frac{1}{q}\mathop{\mathbb{E}}_{Z}\|\mathop{\mathbb{E}}_{Z'}\sum_{i=1}^{q}(Z_i - Z'_i)\|_2 \\
&= \frac{1}{q}\mathop{\mathbb{E}}_{Z}\|\mathop{\mathbb{E}}_{\sigma,Z'}\sum_{i=1}^{q}\sigma_i(Z_i - Z'_i)\|_2 \\
&\le \frac{1}{q}\mathop{\mathbb{E}}_{Z,Z',\sigma}\|\sum_{i=1}^{q}\sigma_i(Z_i - Z'_i)\|_2 \text{ (Jensen)} \\
&\le \frac{2}{q}\mathop{\mathbb{E}}_{Z}\mathop{\mathbb{E}}_{\sigma}\|\sum_{i=1}^{q}\sigma_i Z_i\|_2 \\
&\le \frac{2}{q}\mathop{\mathbb{E}}_{Z}(\mathop{\mathbb{E}}_{\sigma}\|\sum_{i=1}^{q}\sigma_i Z_i\|_2^2)^{1/2} \\
&= \frac{2}{\sqrt{q}}.
\end{aligned}$$

where the σ_i are independent uniform ± 1 random variables. Thus, in expectation, t is u-close to an average of q such random Z_i for $q \ge 4/u^2$. Thus in particular, every t in $B_{\ell_1^n}$ is u-close in ℓ_2 to *some* average of $4/u^2$ of the vectors $\pm e_i$, and thus the set of all such averages is a u-net in ℓ_2, of which there are at most $(2n)^q$. □

One can also obtain a bound on the covering number via a simple volumetric argument, which implies $\mathcal{N}(B^n_{\ell_1}, \ell_2, \varepsilon) = O(2 + 1/(u\sqrt{n}))^n$. Without giving the precise calculations, the argument is to first *upper bound* the maximum number of disjoint radius $(u/2)$-ℓ_2 balls one can pack in $B_{\ell_1^n}$. Then if one takes those balls and considers the union of radius-u balls from their centers, these balls must cover of $B_{\ell_1^n}$ by the triangle inequality and maximality of the original packing. Since all the original packed balls are fully contained in the ℓ_1 ball of radius $1 + (u/2)\sqrt{n}$ by Cauchy-Schwarz, the number of balls in the packing could not have been more than the ratio of the volume of the ℓ_1 ball of radius $(1 + (u/2)\sqrt{n})$, and the volume of an ℓ_2 ball of radius $u/2$. Thus, combining Maurey's lemma and this argument,

$$\forall \varepsilon \in (0, \frac{1}{2}),\ \log^{1/2}\mathcal{N}(B^n_{\ell_1}, \ell_2, \varepsilon) \lesssim \min\{\varepsilon^{-1}\sqrt{\log n},\ \sqrt{n}\cdot\log(1/\varepsilon)\}. \qquad (13)$$

By picking $\varepsilon = ((\log n)/n)^{1/4}$, (5) gives us $g(T) \lesssim (n\log n)^{1/4}$. This is exponentially worse than true bound of $g(T) = \Theta(\sqrt{\log n})$.

4.3 Method 3: Dudley's inequality

Combining (13) with (8),

$$g(T) \lesssim \int_0^{1/\sqrt{n}} \sqrt{n} \cdot \log(1/u) du + \int_{1/\sqrt{n}}^{1} u^{-1} \sqrt{\log n} du \lesssim \log^{3/2} n.$$

This is exponentially better than method 2, but still off from the truth. We can though wonder: perhaps the issue is not Dudley's inequality, but perhaps the entropy bounds of (13) are simply loose? Unfortunately this is not the case. To see this, take a set R of vectors in $\mathbb{R}^n$ that are each $1/\varepsilon^2$-sparse, with ε^2 in each non-zero coordinate, and so that all pairwise ℓ_2 distances are 2ε. A random collection R satisfies this distance property with high probability for $|R| = n^{\Theta(1/\varepsilon^2)}$ and $\varepsilon \gg 1/\sqrt{n}$. Then note $R \subset B_{\ell_1^n}$ and furthermore one needs at least $|R|$ radius-ε balls in ℓ_2 just to cover R.

It is also worth pointing out that this is the worst case for Dudley's inequality: it can never be off by more than a factor of $\log n$. I'll leave it to you as an exercise to figure out why (you should assume the majorizing measures theorem, i.e. that (10) is tight)! **Hint:** compare (9) with (10) and show that nothing interesting happens beyond $r > \log n + c \log\log n$.

4.4 Method 4: generic chaining

By the majorizing measures theorem, we *know* there must exist an admissible sequence giving the correct $g(T) \lesssim \sqrt{\log n}$, thus being superior to Dudley's inequality. Once as an exercise, I tried with Eric Price and Mary Wootters to construct an explicit admissible sequence demonstrating that $\gamma_2(B_{\ell_1^n}, \ell_2) = O(\sqrt{\log n})$. Eric and I managed to find a sequence yielding $O(\log n)$, and Mary found a sequence that gives the correct $O(\sqrt{\log n})$ bound. Below I include Mary's construction.

Henceforth, to be concrete log denotes $\log_2$. Let $\mathcal{N}_s$ be a $1/2^s$-net of the 2^s-sparse vectors in $B_{\ell_2^n}$. Thus

$$|\mathcal{N}_s| \le \binom{n}{2^s} (3 \cdot 2^s)^{2^s}.$$

Then defining $s_k = k - \lceil \log\log(3en) \rceil$,

$$|\mathcal{N}_{s_k}| \le 2^{2^k}.$$

Then we define $T_0 = T_1 = \cdots = T_{\lceil \log\log(3en) \rceil - 1} = \{0\}$, and $T_k = \mathcal{N}_{s_k}$ for $\lceil \log\log(3en) \rceil \le k \le \ell_{max}$ for $\ell_{max} = \log n + \lceil \log\log(3en) \rceil$. For $k \ge \ell_{max}$, we set T_k to be an ε_k-net of $B_{\ell_2^n}$ of size 2^{2^k} for the smallest ε_k possible. If $k = \ell_{max} + j$, then

$$\varepsilon \le n^{-2^j}.$$

We now wish to upper bound the supremum over all $x \in B_{\ell_1^n}$ of

$$\sum_{k=0}^{\infty} 2^{k/2} d_{\ell_2}(x, T_k). \tag{14}$$

We henceforth focus on a particular $x \in B_{\ell_1^n}$ and show that (14) is $O(\sqrt{\log n})$. We split the sum into three parts:

(1) $0 \le k < \lceil \log\log(3en) \rceil$

(2) $\lceil \log\log(3en) \rceil \le k < \ell_{max}$

(3) $\ell_{max} \le k < \infty$

For the summands in (1), each $d_{\ell_2}(x, T_k)$ equals $\|x\|_2 \le 1$, and thus these terms in total contribute at most $2 \cdot 2^{\lceil \log\log(3en) \rceil} = O(\sqrt{\log n})$ to (14). The summands in (3) are also easy to handle: writing $k = \ell_{max} + j$, the summand with index k is at most

$$2^{(\ell_{max}+j)/2} \cdot n^{-2^j} \le \sqrt{n \log n} \cdot 2^{j/2} n^{-2^j},$$

and thus the sum over $j \ge 0$ is $o(1)$ for any $n \ge 2$.

We now proceed with the most involved part of the argument: bounding the contribution of summands in the range (2). For this, we will use a technique that is often referred to in the compressed sensing community as *shelling*. Consider sorting the indices $i \in [n]$ by magnitude $|x_i|$, i.e. $|x_{i_1}| \ge |x_{i_2}| \ge \ldots \ge |x_{i_n}|$. Define the vector $|x|$ by $|x|_i = |x_i|$. Let $A_0 \subset [n]$ denote the coordinates of the 2^0 largest entries of $|x|$, then A_1 the next 2^1 largest entries, then A_2 the next 2^2 largest entries, etc. (if less than 2^s entries remain in x, then A_s is simply the set of remaining entries). The A_s partition $[n]$. Let $x_A \in \mathbb{R}^n$ denote the projection of x onto coordinates in A.

$$\begin{aligned}
\sum_{k=\lceil \log\log(3en) \rceil}^{\log n + \lceil \log\log(3en) \rceil} 2^{k/2} \cdot d_{\ell_2}(x, \mathcal{N}_{s_k}) &\lesssim \sqrt{\log n} \cdot \sum_{s=0}^{\log n} 2^{s/2} \cdot d_{\ell_2}(x, \mathcal{N}_s) \\
&\lesssim \sqrt{\log n} \cdot \sum_{s=0}^{\log n} 2^{s/2} \cdot \left(d_{\ell_2^n}(x_{A_s}, \mathcal{N}_s) + \|x - x_{A_s}\|_2 \right) \\
&\lesssim \sqrt{\log n} + \sqrt{\log n} \cdot \underbrace{\sum_{s=0}^{\log n} 2^{s/2} \cdot \|x - x_{A_s}\|_2}_{\alpha}
\end{aligned}$$

We now wish to show $\alpha = O(1)$.

$$\begin{aligned}
\alpha &\leq \sum_{s=0}^{\log n} 2^{s/2} \left(\sum_{j=s+1}^{\log n} \|x_{A_j}\|_2 \right) \\
&\leq \sum_{s=0}^{\log n} 2^{s/2} \cdot \left(\sum_{j=s+1}^{\log n} 2^{j/2} \|x_{A_j}\|_\infty \right) \\
&= \sum_{j=1}^{\log n} 2^{j/2} \|x_{A_j}\|_\infty \cdot \left(\sum_{s=0}^{j-1} 2^{s/2} \right) \\
&\lesssim \sum_{j=1}^{\log n} 2^{j} \cdot \|x_{A_j}\|_\infty \qquad (15)
\end{aligned}$$

The largest entry of $|x|_{A_j}$ is at most the smallest entry of $|x|_{A_{j-1}}$ by construction, and hence is at most the average entry of $|x|_{A_{j-1}}$. Thus

$$\begin{aligned}
(15) &\leq \sum_{j=1}^{\log n} 2^{j} \cdot \frac{\|x_{A_{j-1}}\|_1}{2^{j-1}} \\
&\leq 2 \cdot \sum_{j=0}^{\log n - 1} \|x_{A_j}\|_1 \\
&\leq 2 \cdot \|x\|_1,
\end{aligned}$$

which is at most $2 = O(1)$, as desired.

5 Application details: dimensionality reduction

We again use the definitions of π_r, Δ_r from Section 3.4. Also, throughout this section we let $\|\cdot\|$ denote the $\ell_{2\to 2}$ operator norm in the case of matrix arguments, and the ℓ_2 norm in the case of vector arguments. Recall $\rho_X(T)$ denotes diameter of T under norm $\|\cdot\|_X$. We use $\|\cdot\|_F$ to denote Frobenius norm.

Krahmer, Mendelson, and Rauhut showed the following theorem [20].

Theorem 1. *Let $\mathcal{A} \subset \mathbb{R}^{m\times n}$ be arbitrary. Let $\sigma_1, \ldots, \sigma_n$ be independent subgaussian random variables of mean* 0 *and variance* 1. *Then*

$$\mathop{\mathbb{E}}_{\sigma} \sup_{A\in\mathcal{A}} \left| \|A\sigma\|^2 - \mathbb{E}\,\|A\sigma\|^2 \right| \lesssim \gamma_2^2(\mathcal{A}, \|\cdot\|) + \gamma_2(\mathcal{A}, \|\cdot\|) \cdot \rho_F(\mathcal{A}) + \rho_F(\mathcal{A}) \cdot \rho_{\ell_{2\to 2}}(\mathcal{A}).$$

We now show that Theorem 1, combined with the majorizing measures theorem, can be used to prove the theorem of Gordon [15] as described in Section 2, and in fact a theorem that is slightly stronger. Gordon's original proof did not use chaining at all. Recall from (2) that we have a point set $P \subset \mathbb{R}^d$, and we want to show that a random matrix $\Pi \in \mathbb{R}^{m\times n}$ satisfies

$$\forall x, y \in P,\ (1-\varepsilon)\|x-y\|_2^2 \le \|\Pi x - \Pi y\|_2^2 \le (1+\varepsilon)\|x-y\|_2^2.$$

for m not too large. In other words, for $T = \{(x-y)/\|x-y\| : x \neq y \in P\}$, we want

$$\sup_{x\in T} |\|\|\Pi x\|^2 - 1| < \varepsilon. \tag{16}$$

We below show that Theorem 1 implies that the *expectation* of the left hand side of (16) is less than ε for $m \gtrsim (g^2(T)+1)/\varepsilon^2$, when the entries of Π are i.i.d. subgaussian with mean 0 and variance $1/m$. Gordon showed the same result but only when the $\Pi_{i,j}$ were independent gaussians and not subgaussians. Note bounding the expectation by ε in (16) implies the actual sup is at most 3ε with probability $2/3$, by Markov's inequality. Much stronger concentration analyses have been given by bounding the L^p norm of the left hand side then performing Markov's inequality on a high moment [24, 9, 10]; we do not cover those approaches here.

We only show the theorem when T is finite. In many applications we care about infinite T (e.g. all the unit norm vectors in a d-dimensional subspace, for applications in numerical linear algebra [31]). In fact, for $T \subset \ell_2^n$ bounded it is without loss of generality to consider only finite T. This is because we can take T' a finite α-net of T, i.e. $\forall x \in T\ \exists x' \in T' : \|x - x'\| \le \alpha$. Then

$$g(T) = \mathop{\mathbb{E}}_g \sup_{x\in T} \langle g, x'\rangle + \langle g, x - x'\rangle = g(T') \pm \mathop{\mathbb{E}}_g \sup_{x\in T} \langle g, x' - x\rangle = g(T') \pm \alpha\sqrt{n}$$

since $|\langle g, x'-x\rangle| \le \|g\|\cdot\|x-x'\|$ and $\mathbb{E}_g \|g\| \le (\mathbb{E}_g \|g\|^2)^{1/2} = \sqrt{n}$. Then we can choose α arbitrarily small so that $g(T')$ is as close to $g(T)$ as we want.

Theorem 2. *Let $T \subset \mathbb{R}^n$ be a finite set of vectors each of unit norm, and let $\varepsilon \in (0, 1/2)$ be arbitrary. Let $\Pi \in \mathbb{R}^{m\times n}$ be such that $\Pi_{i,j} = \sigma_{i,j}/\sqrt{m}$ for independent subgaussian variables $\sigma_{i,j}$ of mean* 0 *and variance* 1*, where $m \gtrsim (g^2(T)+1)/\varepsilon^2$. Then*

$$\mathop{\mathbb{E}}_\sigma \sup_{x\in T} \left|\|\Pi x\|^2 - 1\right| < \varepsilon.$$

Proof. For $x \in T$ let A_x denote the $m \times mn$ matrix defined as follows:

$$A_x = \frac{1}{\sqrt{m}} \cdot \begin{bmatrix} x_1 & \cdots & x_n & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \cdots & 0 & x_1 & \cdots & x_n & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & x_1 & \cdots & x_n \end{bmatrix}.$$

Then $\|\Pi x\|^2 = \|A_x\sigma\|^2$, so letting $\mathcal{A} = \{A_x : x \in T\}$,

$$\mathop{\mathbb{E}}_{\sigma} \sup_{x\in T} \left|\|\Pi x\|^2 - 1\right| = \mathop{\mathbb{E}}_{\sigma} \sup_{A\in\mathcal{A}} \left|\|A\sigma\|^2 - \mathop{\mathbb{E}}_{\sigma} \|A\sigma\|^2\right|.$$

We have $\rho_F(\mathcal{A}) = 1$. Also $A_x^*A_x$ is a block-diagonal matrix, with m blocks each equal to xx^*/m, and thus the singular values of A_x are 0 and $\|x\|/\sqrt{m}$, implying $\rho_{\ell_{2\to 2}}(\mathcal{A}) = 1/\sqrt{m}$. Similarly, since $A_x - A_y = A_{x-y}$, for any vectors x, y we have $\|A_x - A_y\| = \|x - y\|$, and thus $\gamma_2(\mathcal{A}, \|\cdot\|) \le \gamma_2(T, \|\cdot\|)/\sqrt{m}$. Thus by Theorem 1,

$$\mathop{\mathbb{E}}_{\sigma} \sup_{x\in T} \left|\|\Pi x\|^2 - 1\right| \lesssim \frac{\gamma_2^2(T, \|\cdot\|)}{m} + \frac{\gamma_2(T, \|\cdot\|)}{\sqrt{m}} + \frac{1}{\sqrt{m}},$$

which is at most ε for $m \gtrsim (\gamma_2^2(T, \|\cdot\|) + 1)/\varepsilon^2$ as in the theorem statement. This inequality holds by setting $m \gtrsim (g^2(T) + 1)/\varepsilon^2$, since $\gamma_2(T, \|\cdot\|) \lesssim g(T)$ by the majorizing measures theorem. □

We now prove Theorem 1. We only prove it in the case that the σ_i are Rademacher, i.e. uniform ± 1, since this setting already contains the main ideas of the proof. Before we can continue with the proof though, we need a few standard lemmas. The proofs given below are also standard. Recall that for a scalar random variable Z, $\|Z\|_p$ denotes $(\mathbb{E}|Z|^p)^{1/p}$. It is known that $\|\cdot\|_p$ is a norm for $p \ge 1$.

Lemma 2 (Khintchine's inequality)**.** *Let $x \in \mathbb{R}^n$ be arbitrary and $\sigma_1, \ldots, \sigma_n$ be independent Rademachers. Then*

$$\forall p \ge 1,\ \|\langle\sigma, x\rangle\|_p \le \sqrt{p} \cdot \|x\|.$$

This is equivalent, up to constant factors in the exponent, to the following:

$$\forall \lambda > 0,\ \mathop{\mathbb{P}}_{\sigma}(|\langle\sigma, x\rangle| > \lambda) \le 2e^{-\lambda^2/(2\|x\|^2)}.$$

Proof. For the first inequality, consider $\langle g, x\rangle$ for g a vector of independent standard normal random variables. The random variable $\langle g, x\rangle$ is distributed as a gaussian with variance $\|x\|^2$, and thus $\|\langle g, x\rangle\|_p < \sqrt{p} \cdot \|x\|$ by known moment bounds on gaussians. Meanwhile, for positive even integer p, one can expand $\mathbb{E}|\langle g, x\rangle^p| = \mathbb{E}\langle g, x\rangle^p$ as a sum of expectations of monomials. If one similarly expands $\langle\sigma, x\rangle^p$, then we find that these monomials' expectations are term-by-term dominated in the gaussian case, since any even Rademacher moment is 1 whereas all even gaussian moments are at least 1. □

Lemma 3 (Decoupling [12]). *Let $x_1, \ldots, x_n$ be independent and mean zero, and $x'_1, \ldots, x'_n$ identically distributed as the x_i and independent of them. Then for any $(a_{i,j})$ and for all $p \geq 1$*

$$\|\sum_{i \neq j} a_{i,j} x_i x_j\|_p \leq 4 \|\sum_{i,j} a_{i,j} x_i x'_j\|_p$$

Proof. Let $\eta_1, \ldots, \eta_n$ be independent Bernoulli random variables each of expectation $1/2$. Then

$$\begin{aligned}\|\sum_{i \neq j} a_{i,j} x_i x_j\|_p &= 4 \cdot \|\mathbb{E}_{\eta} \sum_{i \neq j} a_{i,j} x_i x_j |\eta_i| |1 - \eta_j| \|_p \\ &\leq 4 \cdot \|\sum_{i \neq j} a_{i,j} x_i x_j \eta_i (1 - \eta_j)\|_p \text{ (Jensen)}\end{aligned} \tag{17}$$

Hence there must be some fixed vector $\eta' \in \{0,1\}^n$ which achieves

$$\|\sum_{i \neq j} a_{i,j} x_i x_j \eta_i (1 - \eta_j)\|_p \leq \|\sum_{i \in S} \sum_{j \notin S} a_{i,j} x_i x_j\|_p$$

where $S = \{i : \eta'_i = 1\}$. Let x_S denote the $|S|$-dimensional vector corresponding to the x_i for $i \in S$. Then

$$\begin{aligned}\|\sum_{i \in S} \sum_{j \notin S} a_{i,j} x_i x_j\|_p &= \|\sum_{i \in S} \sum_{j \notin S} a_{i,j} x_i x'_j\|_p \\ &= \|\mathbb{E}_{x_S} \mathbb{E}_{x'_{\bar{S}}} \sum_{i,j} a_{i,j} x_i x'_j\|_p \ (\mathbb{E}\, x_i = \mathbb{E}\, x'_j = 0) \\ &\leq \|\sum_{i,j} a_{i,j} x_i x'_j\|_p \text{ (Jensen)}\end{aligned}$$

□

5.1 Proof of Theorem 1

We now prove Theorem 1 in the case the σ_i are independent Rademachers. Without loss of generality we can assume $\mathcal{A}$ is finite (else apply the theorem to a sufficiently fine net, i.e. fine in $\ell_2 \to \ell_2$ operator norm). Define

$$E = \mathbb{E}_{\sigma} \sup_{A \in \mathcal{A}} \left| \|A\sigma\|^2 - \mathbb{E}\, \|A\sigma\|^2 \right|$$

and let A^i denote the ith column of A. Then by decoupling

$$E = \mathbb{E}_{\sigma} \sup_{A \in \mathcal{A}} \left| \sum_{i \neq j} \sigma_i \sigma_j \left\langle A^i, A^j \right\rangle \right|$$

$$\leq 4 \cdot \mathop{\mathbb{E}}_{\sigma,\sigma'} \sup_{A\in\mathcal{A}} \left| \sum_{i,j} \sigma_i \sigma'_j \left\langle A^i, A^j \right\rangle \right|$$
$$= 4 \cdot \mathop{\mathbb{E}}_{\sigma,\sigma'} \sup_{A\in\mathcal{A}} |\langle A\sigma, A\sigma'\rangle| .$$

Let $\{T_r\}_{r=0}^{\infty}$ be admissible for $\mathcal{A}$. Direct computation shows

$$\langle A\sigma, A\sigma'\rangle = \langle(\pi_0 A)\sigma, (\pi_0 A)\sigma'\rangle + \sum_{r=1}^{\infty} \underbrace{\langle(\Delta_r A)\sigma, (\pi_{r-1}A)\sigma'\rangle}_{X_r(A)} + \sum_{r=1}^{\infty} \underbrace{\langle(\pi_r A)\sigma, (\Delta_r A)\sigma'\rangle}_{Y_r(A)} .$$

We have $T_0 = \{A_0\}$ for some $A_0 \in \mathcal{A}$. Thus $\mathbb{E}_{\sigma,\sigma'} |\langle(\pi_0 A)\sigma, (\pi_0 A)\sigma'\rangle|$ equals

$$\mathop{\mathbb{E}}_{\sigma,\sigma'} \left|\sigma^* A_0^* A_0 \sigma'\right| \leq \left(\mathop{\mathbb{E}}_{\sigma,\sigma'} (\sigma^* A_0^* A_0 \sigma')^2\right)^{1/2} = \|A_0^* A_0\|_F \leq \|A_0\|_F \|A_0\| \leq \rho_F(\mathcal{A}) \cdot \rho_{\ell_{2\to 2}}(\mathcal{A}).$$

Thus,

$$\mathop{\mathbb{E}}_{\sigma,\sigma'} \sup_{A\in\mathcal{A}} |\langle A\sigma, A\sigma'\rangle| \leq \rho_F(\mathcal{A}) \cdot \rho_{\ell_{2\to 2}}(\mathcal{A}) + \mathop{\mathbb{E}}_{\sigma,\sigma'} \sup_{A\in\mathcal{A}} \sum_{r=1}^{\infty} |X_r(A)| + \mathop{\mathbb{E}}_{\sigma,\sigma'} \sup_{A\in\mathcal{A}} \sum_{r=1}^{\infty} |Y_r(A)|.$$

We focus on the second summand; handling the third summand is similar.

Note $X_r(A) = \langle(\Delta_r A)\sigma, (\pi_{r-1}A)\sigma'\rangle = \langle\sigma, (\Delta_r A)^*(\pi_{r-1}A)\sigma'\rangle$. Thus by the Khintchine inequality (namely $\| \langle\sigma, x\rangle \|_p \lesssim \sqrt{p} \cdot \|x\|$),

$$\mathbb{P}(|X_r(A)| > t2^{r/2} \cdot \|(\Delta_r A)^*(\pi_{r-1}A)\sigma'\|) \lesssim e^{-t^2 2^r/2}.$$

Let $\mathcal{E}(A)$ be the event that for all $r \geq 1$ simultaneously, $|X_r(A)| \leq t2^{r/2} \cdot \|\Delta_r A\| \cdot \sup_{A\in\mathcal{A}} \|A\sigma'\|$. Then

$$\mathbb{P}(\exists A \in \mathcal{A}\ s.t.\ \neg\mathcal{E}(A)) \lesssim \sum_{r=1}^{\infty} |T_r| \cdot |T_{r-1}| \cdot e^{-t^2 2^r/2}$$
$$\leq \sum_{r=1}^{\infty} 2^{2^{r+1}} \cdot e^{-t^2 2^r/2}.$$

Therefore

$$\mathop{\mathbb{E}}_{\sigma,\sigma'} \sup_{A\in\mathcal{A}} \sum_{r=1}^{\infty} |X_r(A)| = \mathop{\mathbb{E}}_{\sigma'} \int_0^{\infty} \mathop{\mathbb{P}}_{\sigma} \left(\sup_{A\in\mathcal{A}} \sum_{r=1}^{\infty} |X_r(A)| > t\right) dt,$$

which by a change of variables is equal to

$$\mathop{\mathbb{E}}_{\sigma'} \left(\sup_{A\in\mathcal{A}} \|A\sigma'\| \cdot \left(\sup_{A\in\mathcal{A}} \sum_{r=1}^{\infty} 2^{r/2} \|\Delta_r A\|\right)\right.$$

$$\begin{aligned}
&\times \cdot \int_0^\infty \mathop{\mathbb{P}}_{\sigma}\left(\sup_{A\in\mathcal{A}} \sum_{r=1}^\infty |X_r(A)| > t \sup_{A\in\mathcal{A}} 2^{r/2} \cdot \|\Delta_r A\| \cdot \sup_{A\in\mathcal{A}} \|A\sigma'\|\right) dt\Bigg) \\
&\le \left(\mathop{\mathbb{E}}_{\sigma'} \sup_{A\in\mathcal{A}} \|A\sigma'\|\right) \cdot \left(\sup_{A\in\mathcal{A}} \sum_{r=1}^\infty 2^{r/2}\|\Delta_r A\|\right) \cdot \left[3 + \sum_{r=1}^\infty \int_3^\infty 2^{2^{r+1}} e^{-t^2 2^r/2} dt\right] \\
&\lesssim \left(\mathop{\mathbb{E}}_{\sigma'} \sup_{A\in\mathcal{A}} \|A\sigma'\|\right) \cdot \sup_{A\in\mathcal{A}} \sum_{r=1}^\infty 2^{r/2}\|\Delta_r A\| \\
&\lesssim \left(\mathop{\mathbb{E}}_{\sigma'} \sup_{A\in\mathcal{A}} \|A\sigma'\|\right) \cdot \sup_{A\in\mathcal{A}} \sum_{r=1}^\infty 2^{r/2} \cdot \rho_{2\to 2}(A, T_r),
\end{aligned}$$

since $\|\Delta_r A\| \le \rho_{2\to 2}(A, T_{r-1}) + \rho_{2\to 2}(A, T_r)$ via the triangle inequality. Choosing admissible $T_0 \subseteq T_1 \subseteq \ldots \subseteq T$ to minimize the above expression,

$$E \lesssim \rho_F(\mathcal{A}) \cdot \rho_{\ell_{2\to 2}}(\mathcal{A}) + \gamma_2(\mathcal{A}, \|\cdot\|) \cdot \mathop{\mathbb{E}}_{\sigma'} \sup_{A\in\mathcal{A}} \|A\sigma'\|.$$

Now observe

$$\begin{aligned}
\mathop{\mathbb{E}}_{\sigma'}\left(\sup_{A\in\mathcal{A}} \|A\sigma'\|\right) &\le \left(\mathop{\mathbb{E}}_{\sigma'} \sup_{A\in\mathcal{A}} \|A\sigma'\|^2\right)^{1/2} \\
&\le \left(\mathop{\mathbb{E}}_{\sigma'}\left(\sup_{A\in\mathcal{A}} \left|\|A\sigma'\|^2 - \mathop{\mathbb{E}}_{\sigma'} \|A\sigma'\|^2\right| + \mathop{\mathbb{E}}_{\sigma'} \|A\sigma'\|^2\right)\right)^{1/2} \\
&= \left(\mathop{\mathbb{E}}_{\sigma'} \sup_{A\in\mathcal{A}}\left(\left|\|A\sigma'\|^2 - \mathop{\mathbb{E}}_{\sigma'} \|A\sigma'\|^2\right| + \|A\|_F^2\right)\right)^{1/2} \\
&\le \sqrt{E} + \rho_F(\mathcal{A})
\end{aligned}$$

Thus in summary,

$$E \lesssim \rho_F(\mathcal{A}) \cdot \rho_{\ell_{2\to 2}}(\mathcal{A}) + \gamma_2(\mathcal{A}, \|\cdot\|) \cdot (\sqrt{E} + \rho_F(\mathcal{A})).$$

This implies E is at most the square of the larger root of the associated quadratic equation, which gives the theorem.

Acknowledgments

I thank Oded Regev for pointing out that Dudley's inequality cannot yield a bound on the mean width of $B_{\ell_1^n}$ that is better than $O(\log^{3/2} n)$ (see Section 4.3), and I thank Mary Wootters for allowing me to include her argument for the admissible sequence for the ℓ_1 ball in Section 4. A preliminary version of this note appeared on the "Windows on Theory" blog. I thank commenters there, in addition to Assaf Naor and David Woodruff, for pointing out typographical errors and for other suggestions.

References

[1] Radosław Adamczak. A note on the sample complexity of the Er-SpUD algorithm by Spielman, Wang and Wright for exact recovery of sparsely used dictionaries. *CoRR*, abs/1601.02049, 2016.

[2] Nir Ailon and Edo Liberty. An almost optimal unrestricted fast Johnson-Lindenstrauss transform. *ACM Transactions on Algorithms*, 9(3):21, 2013.

[3] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *J. Comput. Syst. Sci.*, 58(1):137–147, 1999.

[4] Vladimir Braverman, Stephen R. Chestnut, Nikita Ivkin, Jelani Nelson, Zhengyu Wang, and David P. Woodruff. BPTree: an ℓ_2 heavy hitters algorithm using constant memory. *CoRR*, abs/1603.00759, 2016.

[5] Vladimir Braverman, Stephen R. Chestnut, Nikita Ivkin, and David P. Woodruff. Beating CountSketch for Heavy Hitters in Insertion Streams. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC)*, pages 740–753, 2016. Full version at arXiv abs/1511.00661.

[6] Jean Bourgain, Sjoerd Dirksen, and Jelani Nelson. Toward a unified theory of sparse dimensionality reduction in Euclidean space. *Geometric and Functional Analysis (GAFA)*, 25(4):1009–1088, July 2015. Preliminary version in STOC 2015.

[7] Jaroslaw Blasiok and Jelani Nelson. An improved analysis of the er-spud dictionary learning algorithm. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 44:1–44:14, 2016. Full version at arXiv abs/1602.05719.

[8] Jian Ding. Asymptotics of cover times via Gaussian free fields: Bounded-degree graphs and general trees. *Annals of Probability*, 42(2):464–496, 2014.

[9] Sjoerd Dirksen. Dimensionality reduction with subgaussian matrices: a unified theory. *Found. Comput. Math.*, pages 1–30, 2015. Full version at arXiv abs/1402.3973.

[10] Sjoerd Dirksen. Tail bounds via generic chaining. *Electron. J. Probab.*, 20(53):1–29, 2015.

[11] Jian Ding, James R. Lee, and Yuval Peres. Cover times, blanket times, and majorizing measures. *Annals of Mathematics*, 175:1409–1471, 2012.

[12] Victor de la Peña and Evarist Giné. *Decoupling: From dependence to independence*. Probability and its Applications. Springer-Verlag, New York, 1999.

[13] Richard M. Dudley. The sizes of compact subsets of Hilbert space and continuity of Gaussian processes. *J. Functional Analysis*, 1:290–330, 1967.

[14] Xavier Fernique. Regularité des trajectoires des fonctions aléatoires gaussiennes. *Lecture Notes in Math.*, 480:1–96, 1975.

[15] Yehoram Gordon. On Milman's inequality and random subspaces which escape through a mesh in $\mathbb{R}^n$. *Geometric Aspects of Functional Analysis*, pages 84–106, 1988.

[16] Zengfeng Huang, Wai Ming Tai, and Ke Yi. Tracking the frequency moments at all times. *CoRR*, abs/1412.1763, 2014.

[17] Piotr Indyk and Assaf Naor. Nearest-neighbor-preserving embeddings. *ACM Transactions on Algorithms*, 3(3), 2007.

[18] T. S. Jayram and David P. Woodruff. Optimal bounds for Johnson-Lindenstrauss transforms and streaming problems with subconstant error. *ACM Trans. Algorithms*, 9(3):26, 2013.

[19] Bo'az Klartag and Shahar Mendelson. Empirical processes and random projections. *J. Funct. Anal.*, 225(1):229–245, 2005.

[20] Felix Krahmer, Shahar Mendelson, and Holger Rauhut. Suprema of chaos processes and the restricted isometry property. *Comm. Pure Appl. Math.*, 2014.

[21] Ioannis Karatzas and Steven E. Shreve. *Brownian motion and stochastic calculus*, volume 113 of *Graduate Texts in Mathematics*. Springer-Verlag, 1991.

[22] Kyle Luh and Van Vu. Random matrices: l_1 concentration and dictionary learning with few samples. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 1409–1425, 2015.

[23] Raghu Meka. A PTAS for computing the supremum of gaussian processes. In *53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 217–222, 2012.

[24] Shahar Mendelson, Alain Pajor, and Nicole Tomczak-Jaegermann. Reconstruction and subgaussian operators in asymptotic geometric analysis. *Geometric and Functional Analysis*, 17:1248–1282, 2007.

[25] Samet Oymak, Benjamin Recht, and Mahdi Soltanolkotabi. Isometric sketching of any set via the restricted isometry property. *CoRR*, abs/1506.03521, 2015.

[26] Atri Rudra and Mary Wootters. Every list-decodable code for high noise has abundant near-optimal rate puncturings. In *Proceedings of the 46th ACM Symposium on Theory of Computing (STOC)*, pages 764–773, 2014.

[27] Daniel A. Spielman, Huan Wang, and John Wright. Exact recovery of sparsely-used dictionaries. In *Proceedings of the 25th Annual Conference on Learning Theory (COLT)*, pages 37.1–37.18, 2012.

[28] Michel Talagrand. Are many small sets explicitly small? In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC)*, pages 13–36, 2010.

[29] Michel Talagrand. *Upper and lower bounds for stochastic processes: modern methods and classical problems*. Springer, 2014.

[30] Ramon van Handel. Probability in high dimensions. Manuscript, 2014. Available at `https://www.princeton.edu/~rvan/ORF570.pdf`. Version from June 30, 2014.

[31] David P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10(1-2):1–157, 2014.

[32] Alex Zhai. Exponential concentration of cover times. *CoRR*, abs/1407.7617, 2014.

The Computational Complexity Column

BY

Vikraman Arvind

Institute of Mathematical Sciences, CIT Campus, Taramani
Chennai 600113, India
arvind@imsc.res.in
http://www.imsc.res.in/~arvind

Perhaps the oldest algorithmic technique used for the Graph Isomorphism prob-lem is the Weisfeiler-Lehman procedure. Invented in the late 1960's, it has attracted research in several directions over the last five decades, and continues to be an actively researched topic. Algorithmists invariably use this procedure in combination with others tools for Graph Isomorphism. Logicians interested in descriptive complexity have found logical characterizations for it. There is a linear programming connection to the Weisfeiler-Lehman procedure, and the Sherali-Adams hierarchy for the natural LP relaxation of a 0-1 integer linear pro-gram for Graph Isomorphism turns out to be intimately connected to "higher dimensional" versions of the procedure.

This brief essay, meant as an invitation to the topic, will touch upon these as-pects of the Weisfeiler-Lehman procedure. However, it is by no means a complete survey.

The Weisfeiler-Lehman Procedure

V. Arvind
Institute of Mathematical Sciences, Chennai, India
arvind@imsc.res.in

1 Introduction

Two simple, undirected n-vertex graphs $X = (V, E)$ and $X' = (V', E')$ are *isomorphic* if there is a bijection $\pi : V \to V'$ that maps edges to edges and non-edges to non-edges. I.e.

$$\{u^\pi, v^\pi\} \in E' \leftrightarrow \{u, v\} \in E.$$

The Graph Isomorphism problem is to test if a given pair of graphs X, X' are isomorphic. A generic procedure for the Graph Isomorphism problem builds on a simple *color refinement* procedure. It is an iterative procedure for coloring the vertices of a graph X:

- To begin with, all vertices have the same color.
- In each color refinement step, if two vertices u and v have the same color but their neighborhoods are differently colored (counting color multiplicity), then u and v get fresh different colors.

This iterative procedure stops when the coloring does not refine any further, i.e. it becomes a *stable coloring*.

Color refinement yields a simple isomorphism test when applied to the disjoint union $X \sqcup X'$ of X and X'. In the stable coloring for $X \sqcup X'$, if the number of vertices colored c, for some color c is different in X and X' then they are clearly not isomorphic. But the converse is not always true. For example, if X and X' are two regular nonisomorphic graphs then the stable coloring is just the initial coloring which does not distinguish between any two vertices. Nevertheless, this simple method is a basic tool in many of the current algorithms for Graph Isomorphism. Even practical Graph Isomorphism testing tools like NAUTY [18] are based on color refinement.

The color refinement method dates back to a 1968 paper by Weisfeiler and Lehman [23], where they actually proposed a stronger method of coloring pairs of vertices. This was subsequently generalized to the k-dimensional Weisfeiler-Lehman method (k-WL for short) for a graph $X = (V, E)$. The k-WL procedure colors k-tuples of vertices of X. Two k-tuple $(u_1, \ldots, u_k)$ and $(v_1, \ldots, v_k)$ are i-adjacent if $u_j = v_j$ for all $j \neq i$.

- Initially all k-tuples of vertices $(u_1, u_2, \ldots, u_k)$ are colored by the isomorphism type of the induced ordered subgraphs. I.e. $(u_1, \ldots, u_k)$ and $(v_1, \ldots, v_k)$ are colored the same if and only if the map $u_i \mapsto v_i$ is an isomorphism between the respective induced k-vertex subgraphs.

- In a general refinement step, the k-tuples $(u_1, \ldots, u_k)$ and $(v_1, \ldots, v_k)$ are colored different if for some index i, there are a different number of k-tuples colored c that are i-adjacent to $(u_1, \ldots, u_k)$ and $(v_1, \ldots, v_k)$.

The above procedure stops when a stable coloring is reached, which will clearly be in at most $|V|^k$ refinement steps. It turns out that 2-WL coincides with the color refinement procedure.

A graph X is said to be *identified* by color refinement if for any nonisomorphic graph X', the above procedure distinguishes them. For instance, trees (and forests) are identified by color refinement. Furthermore, Babai, Erdös and Selkow [4] have shown that a random graph is identified by color refinement with high probability; in fact, the stable coloring gives distinct colors to all vertices in a mere two rounds. For larger k, the k-WL method is known to be more powerful. For instance, it follows from [16] that 3-WL succeeds with high probability on random regular graphs. There is a fixed k such that k-WL succeeds on all planar graphs [11]. Furthermore, Grohe has extended this result to all graph classes characterized by excluded minors [12].

When precisely are two graphs X and X' *indistinguishable* by color refinement? I.e. for which pair of graphs X and X' does it hold that when we run color refinement on their disjoint union $X \sqcup X'$ and obtain the stable coloring, for any color c the number of vertices colored c is the same in X and X'? It turns out that there is are two nice characterizations of this property. One is logical and descriptive complexity-theoretic. The other is a geometric characterization, based on linear programming. We first explain the linear programming based characterization.

1.1 A linear programming characterization

There is a natural integer programming formulation of the Graph Isomorphism problem.

Let X and X' be two undirected simple graphs on vertex set $[n]$, with adjacency matrices A and B respectively. Suppose the graphs X and X' are isomorphic witness by permutation $\pi : [n] \to [n]$ that maps X to X'. Let P be the permutation matrix corresponding to π:

$$P_{ij} = 1 \leftrightarrow \pi(i) = j.$$

Then $AP = PB$. Conversely, if $AP = PB$ for a permutation matrix P, then the permutation π corresponding to P is an isomorphism from X to X'. We can express this as a 0-1 integer linear program given in the statement below.

Fact 1. *Let X and X' be two n-vertex graphs with adjacency matrices A and B respectively. An $n \times n$ permutation matrix P encodes an isomorphism π from X to X' if and only if P is a solution to the following* 0-1 *integer linear program:*

$$\begin{aligned} AP &= PB, \\ \sum_{i=1}^{n} P_{ij} &= 1, 1 \leq j \leq n, \\ \sum_{j=1}^{n} P_{ij} &= 1, 1 \leq i \leq n, \\ P_{ij} &\in \{0,1\}, 1 \leq i, j \leq n. \end{aligned}$$

The above Integer Linear Program (ILP) is feasible if and only if the graphs corresponding to A and B are isomorphic. Notice that the row and column sum being 1, and the P_{ij} taking 0-1 values forces P to be a permutation matrix. A natural LP relaxation of the above ILP is

$$\begin{aligned} AP &= PB, \\ \sum_{i=1}^{n} P_{ij} &= 1, 1 \leq j \leq n, \\ \sum_{j=1}^{n} P_{ij} &= 1, 1 \leq i \leq n, \\ P_{ij} &\geq 0, 1 \leq i, j \leq n, \end{aligned}$$

and the solutions to this LP are called *fractional isomorphisms*. Clearly, the fractional isomorphisms P are all doubly stochastic matrices. If we drop the equality constraints given by $AP = PB$, the resulting system of linear inequalities defines the polytope of all doubly stochastic matrices which, by Birkhoff's theorem, has as its extreme points all the $n!$ many permutation matrices.

We say that the graphs X and X' are fractionally isomorphic if the above LP relaxation has a fractional solution, where A and B are the adjacency matrices of X and X'. The following theorem by Ramana, Schneierman, and Ullman [19] shows a remarkable connection between fractional isomorphisms and the color refinement procedure.

Theorem 2. [19] *The graphs X and X' are fractionally isomorphic if and only if they are indistinguishable by color refinement.*

The proof relies on the Perron-Frobenius theorem and the notion of *equitable partitions* of a graph $X = (V, E)$. It is a partition of the vertex set

$$V = C_1 \sqcup C_2 \sqcup \cdots \sqcup C_r$$

such that the subgraph $X[C_i]$ induced by C_i is regular and the bipartite graph $X[C_i, C_j]$ is semi-regular for all $i \neq j$. For instance the *discrete partition* in which each C_i is a singleton is equitable. As it turns out, at the other extreme we have the equitable partition given by the stable coloring computed by color refinement, which is actually the *coarsest equitable partition*: any other equitable partition is a refinement of the stable coloring.

Let A be the adjacency matrix of graph X. A doubly stochastic matrix P is a *fractional automorphism* of X if $AP = PA$. We can interpret the matrix P as the adjacency matrix of a directed graph G_P with nonnegative weights. An important observation of [19] is that the strongly connected components of the directed graph G_P must form an equitable partition of X. As a consequence, it follows that any fractional automorphism P has to be block diagonal, where the blocks correspond to the stable coloring partition of V.

1.2 The convex set of fractional automorphisms

Let $X = (V, E)$ be an undirected graph with adjacency matrix A. The set of all fractional automorphisms P forms a convex set defined by the LP: $AP = PA$ such that $\sum_i P_{ij} = 1$, $\sum_j P_{ij} = 1$, and $P_{ij} \geq 0$, for $1 \leq i, j \leq n$.

Proposition 3. *The fractional automorphisms of X forms a semigroup* $\mathrm{FracA}(X)$ *under matrix multiplication.*

Proof. If $P_1, P_2 \in \mathrm{FracA}(X)$ are fractional automorphisms of X then we have $AP_1 = P_1A$ and $AP_2 = P_2A$. It follows that

$$AP_1P_2 = P_1AP_2 = P_1P_2A.$$

The semigroup also has the identity matrix I which is the identity element. $\square$

Since $\mathrm{FracA}(X)$ is a convex set, it is also closed under convex combinations. I.e. if $P_i \in \mathrm{FracA}(X), 1 \le i \le t$ and $\alpha_i, 1 \le i \le t$ are nonnegative such that $\sum_i \alpha_i = 1$ then $\sum_i \alpha_i P_i \in \mathrm{FracA}(X)$.

Let $\mathrm{Aut}(X)$ denote the automorphism group of X (we use the same notation $\mathrm{Aut}(X)$ whether we treat its elements as permutations on the vertices or $|V| \times |V|$ permutation matrices). Clearly, $\mathrm{Aut}(X) \subseteq \mathrm{FracA}(X)$.

Proposition 4. $\mathrm{Aut}(X)$ *coincides with* $\mathrm{FracA}(X)$ *if and only if the stable coloring of the graph X yields the discrete partition.*

Proof. If the stable coloring yields the discrete partition, then by Theorem 2 stated above it follows that the only fractional automorphism of X is the identity matrix which implies $\mathrm{Aut}(X) = \mathrm{FracA}(X)$.

Conversely, suppose $\mathrm{Aut}(X) = \mathrm{FracA}(X)$. Now, suppose the stable coloring is the equitable partition

$$V = C_1 \sqcup C_2 \sqcup \cdots \sqcup C_r,$$

is not discrete. Consider the block diagonal doubly stochastic matrix P, with blocks defined by subsets $C_1, C_2, \ldots, C_r$, such that for all $u, v \in C_k$ we have $P_{uv} = \frac{1}{|C_k|}$. As the stable coloring is an equitable partition, it follows that P is a fractional automorphism. Furthermore, P is not in $\mathrm{Aut}(X)$ which contradicts the assumption. $\square$

If X is a regular graph with no nontrivial automorphisms then $\mathrm{Aut}(X)$ is a proper subset of $\mathrm{FracA}(X)$, because $\mathrm{Aut}(X) = \{1\}$ and $\mathrm{FracA}(X)$ contains $\frac{1}{d}A$, where d is the degree of each vertex in X.

By Birkhoff's theorem we know that the extreme points of the polytope of doubly stochastic matrices are precisely the $n!$ many permutation matrices. As a consequence, for any graph X, all matrices in $\mathrm{Aut}(X)$ are extreme points of $\mathrm{FracA}(X)$. However, in general, the fractional automorphism polytope $\mathrm{FracA}(X)$ may have non-integral extreme points.

For graphs X and X', let $\mathrm{FracI}(X, X')$ denote the (possibly empty) convex polytope of all fractional isomorphisms from X to X' given by Equation 1. We note that $\mathrm{FracA}(X)P \subseteq \mathrm{FracI}(X, X')$, where P is some fractional solution to Equation 1. Now, there are graphs X (like forests, for instance) such that $\mathrm{FracI}(X, X') \neq \emptyset$ if and only if there is an integral solution in $\mathrm{FracI}(X, X')$ (i.e., X and X' are isomorphic). This will happen if the set of all extreme points of the convex polytope $\mathrm{FracA}(X)$ is precisely $Aut(X)$. This property was noticed by Tinhofer [21, 22], and he called such graphs X *compact*. For example, forests are compact. If X is compact, Tinhofer [21, 22] gives an algorithm to compute an isomorphism from X to another graph X', if it exists, by computing an extreme point solution for the linear program given by Equation 1 [21, 22].

2 Logical perspective

Immerman and Lander [15] wrote a seminal paper introducing a first-order logic based approach to understanding the color refinement procedure. In order to state their results precisely, we will require some basic definitions from their paper.

The first-order language of graphs is built from variables x_i, the binary edge relation E and equality $=$, along with the usual logical connectives and quantifiers $\forall$ and $\exists$. The quantifiers range over the vertex set of a given graph. Occasionally, it is useful to consider vertex colored graphs, where the colors are defined by unary predicates.

For any given language $\mathcal{L}$ (either first-order or a suitable extension of it, usually), we say that graphs G and H are $\mathcal{L}$-equivalent iff for all sentences $\varphi \in \mathcal{L}$ we have

$$G \models \varphi \leftrightarrow H \models \varphi.$$

A *k-valuation* over graph G is an assignment u of vertices to variables $x_1, x_2, \ldots, x_k$. Suppose u and v are k-valuations for graphs G and H respectively. We say that G, u and H, v are $\mathcal{L}$-equivalent iff for all formulas $\varphi \in \mathcal{L}$ with free variables from $x_1, \ldots, x_k$

$$G, u \models \varphi \leftrightarrow H, v \models \varphi.$$

We say that $\mathcal{L}$ k-characterizes G iff for all graphs H, and all k-valuations u and v over G and H respectively, if G, u and H, v are $\mathcal{L}$-equivalent then there is an isomorphism extending the correspondence given by (u, v).

2.1 The first-order language $\mathcal{C}_k$

The language $\mathcal{L}_k$ is defined to be first-order formulas which use k variables. The language $\mathcal{C}_k$ is defined to be first-order formulas with k variables, where the formulas use *counting quantifiers*: For example, the formula $(\exists^i x)\varphi(x)$ means there are at least i vertices v such that $\varphi(v)$ is true.

It turns out that the language $\mathcal{C}_2$ precisely corresponds to color refinement.

Theorem 5. [15] *Given a graph $G = (V, E)$, let $\overline{f}$ denote the stable coloring of V produced by color refinement. let v_1 and v_2 be two vertices of G. The following conditions are equivalent:*

- $\overline{f}(v_1) = \overline{f}(v_2)$.
- *For each formula $\varphi(x) \in \mathcal{C}_2$*

$$G \models \varphi(v_1) \leftrightarrow G \models \varphi(v_2).$$

The proof is by an inductive argument on the number of color refinement rounds which corresponds to the quantifier depth of the formula φ. I.e. v_1 and v_2 are indistinguishable by color refinement in r rounds precisely when $\mathcal{C}_2$ formulas $\varphi(x)$ of quantifier depth r cannot distinguish between v_1 and v_2. An important tool in analyzing the power of $\mathcal{C}_2$ is the following *two-player pebble game*. For a pair of graphs G and H the $\mathcal{C}_2$-game on them is defined as follows: there are two pairs of pebbles $(g_1, h_1), (g_2, h_2)$:

1. The first player takes a pebble, say g_i, and chooses a subset A of vertices from one of the graphs. The second player has to choose a subset B of vertices from the other graph such that $|A| = |B|$.
2. The first player places h_i on some vertex in B and the second player has to respond by placing g_i on some vertex in A.

The first player wins iff the subgraph induced by g_1, g_2 is not the same as that induced by h_1, h_2. Otherwise, the second player wins.

The above theorem actually shows that $\overline{f}(v_1) = \overline{f}(v_2)$ iff the first player has a winning strategy in the above game played on two copies of G with g_1 placed on v_1 in the first copy and h_1 placed on v_2 in the second copy.

The Immerman-Lander theorem combined with Theorem 2 gives a beautiful three-way characterization of graphs G and H that are indistinguishable by color refinement. This can be briefly summarized as below:

Theorem 6. [19, 15] *The following statements are equivalent:*

- *Graphs G and H are indistinguishable by color refinement.*
- *Graphs G and H are indistinguishable by formulas in $\mathcal{C}_2$.*
- *Graphs G and H are fractionally isomorphic.*

Immerman and Lander [15] also generalize their result to show that graphs G and H are indistinguishable in $\mathcal{C}_k$ if and only if they are indistinguishable by $(k-1)$-WL (the $(k-1)$-dimensional Weisfeiler-Lehman procedure).

2.2 Weisfeiler-Lehman and Graph Isomorphism

Coming back to the color refinement procedure as an algorithm for Graph Isomorphism, it is natural to ask for which graphs does it give the correct answer. We say that color refinement succeeds on a graph G iff for any nonisomorphic graph H, color refinement distinguishes between G and H. Equivalently, G and H are isomorphic iff they are indistinguishable in $\mathcal{C}_2$. As already noted, color refinement succeeds on forests, and also on random graphs with high probability. In [2, 20] the class of graphs on which color refinement succeeds is completely characterized. In particular, these graphs can also be eciently recognized. However, the problem of precisely characterizing the class of graphs on which k-WL succeeds remains open, for $k \geq 3$.

In this connection, it is natural to wonder if k-WL for some k could be powerful enough to solve Graph Isomorphism on all instances. It is easy to see that $k = n$ suces. We have already noted that for each n-vertex graph G there is a first-order formula using n variables that is true on G and not on any nonisomorphic graph H. Whether a smaller k suces remained open until the seminal paper by Cai, Fürer, and Immerman [7] in which they showed a lower bound of $k = \Omega(n)$. More precisely, they proved the following result.

Theorem 7. [7] *There exists a sequence of nonisomorphic graph pairs* $\{G_n, H_n\}_n$ *such that* G_n *and* H_n *have* $\Theta(n)$ *vertices, but* G_n *and* H_n *are indistinguishable by* n*-WL.*

It is interesting to note that the graphs G_n and H_n, ingeniously constructed, are actually very simple instances of Graph Isomorphism. That is to say, they are vertex-colored graphs with at most 4 vertices of each color, and the problem is to check if G_n and H_n have a *color-preserving* isomorphism. Such instances of Graph Isomorphism (with bounded size color classes) are easily solved in polynomial time by simple group-theoretic techniques from [10].

Since the k-WL procedure takes essentially n^k time, it is clear from this theorem that the Weisfeiler-Lehman procedure alone is not enough to get an ecient algorithm for Graph Isomorphism.

Nevertheless, it is often a crucial component in many algorithms for Graph Isomorphism. For instance, Lindell's logspace algorithm for Tree Isomorphism (and Canonization) [17] is essentially a clever logspace implementation of color refinement on trees. Another appealing paper in this direction is due to Grohe and Verbitsky [14]. They note that if graphs from a graph class C can be identified in $\mathcal{C}_k$ for small k using a formula of logarithmic quantifier depth, then it is possible to find efficient *parallel algorithms* for isomorphism (and even canonization) for such graphs. Using their method they could show that planar graph isomorphism can be solved in the parallel circuit class AC^1 (which is contained in NC^2). Similarly, they show that bounded treewidth graph isomorphism is in the circuit class TC^1 (also contained in NC^2). Recent work with much more complicated algorithms has improved these upper bounds to logarithmic space.

Remark 8. *The Immerman-Lander theorem has also driven a lot of research in the study of new extensions of first-order logic. There is interesting work of Dawar et al [8] using a rank operator based extension of first-order logic.*

Remark 9. *Finally, we briefly mention that the technique of* Individualization *of vertices combined with Weisfeiler-Lehman is a powerful tool for obtaining efficient ismorphism algorithms. Originally, it was introduced by Zemlyachenko as a "degree reduction trick" yielding the* $2^{O(\sqrt{n \ln n})}$ *time isomorphism algorithm [5]. It also plays a significant role in Babai's recent breakthrough algorithm [6].*

3 More linear programming

This section is essentially based on the work of Atserias and Maneva [3] in which they consider the different levels of the Sherali-Adams LP relaxation hierarchy of Equation 1 and shows a close relationship to k-dimensional Weisfeiler-Lehman for $1 \leq k \leq n$. We will state their main result and point to some questions that arise from their work. We begin with defining the Sherali-Adams relaxation.

Consider any 0-1 integer linear program

$$\begin{aligned} Ax &\geq b, \\ x_i &\in \{0,1\}, \forall\, x_i. \end{aligned}$$

Let P_{int} denote the convex polytope which is the convex hull of solutions $x \in \{0,1\}^n$ of the above integer linear program.

Its LP relaxation, where the integral constraints are replaced by $0 \leq x_i \leq 1$, also defines a convex polytope P. Clearly,

$$P \supseteq P_{int}.$$

In the special case when $P = P_{int}$ we can use linear programming to find integral solutions. Otherwise, one approach to understanding P_{int} is by defining a sequence of relaxations that "interpolate" P and P_{int}. In particular, the *Sherali-Adams* hierarchy is defined by a sequence of "approximating" polytopes

$$P = P_1 \supseteq P_2 \supseteq \cdots \supseteq P_n = P_{int},$$

where the k^{th} polytope in the sequence is obtained as follows from P:

For all subsets $I \in \binom{[n]}{\leq k-1}$, and all partitions $I = I_1 \sqcup I_2$, each inequality $\sum_{j=1}^{n} A_{ij} x_j \geq 0$ is multiplied by $\prod_{i \in I_1} x_i \prod_{j \in I_2} (1 - x_j)$. Overall, this yields a system of polynomial inequalities, where each polynomial's degree is at most k in each inequality.

Next, in each polynomial inequality, each monomial is "flattened" into a multilinear monomial by repeatedly replacing x_i^2 with x_i. This yields a system of multilinear polynomial inequalities, where each polynomial has degree at most k. Then, each monomial $\prod_{i \in A} x_i$ is replaced by a new variable y_A, and a fresh constraint $y_\emptyset = 1$ is included. This results in an LP Q_k in variables $y_A, A \in \binom{[n]}{\leq k}$. Finally, the polytope $P_k \subseteq \mathbb{R}^n$ is defined by projecting Q_k to the n variables y_A such that $|A| = 1$. Clearly, the LP defining P_k requires the variables $y_A, A \in \binom{[n]}{\leq k}$ and is of size $n^{O(k)}$.

By examining the integral solutions it is clear that $P_{int} \subseteq P_k$. Since the constraints of P_k are only tighter we have $P_k \subseteq P_1$. Furthermore, it turns out that $P_{int} = P_n$. Thus, the hierarchy is finite. Thus, if $P_k = P_{k+1}$ for some $k < n$, it follows that $P_k = P_{int}$.

Following [3], we consider the Sherali-Adams hierarchy corresponding to the 0-1 integer linear program defined in Fact 1. Let that polytope be denoted by P_{int}^{gi}, and let P^{gi} denote its LP relaxation given in Section 1, which defines fractional isomorphisms. The Sherali-Adams relaxations yields the following sequence of polytopes:

$$P^{gi} = P_1^{gi} \supseteq P_2^{gi} \supseteq \cdots \supseteq P_n^{gi} = P_{int}.$$

Let X and X' be two n-vertex graphs with adjacency matrices A and B, and consider the above Sherali-Adams relaxation hierarchy. Say that X and X' are *fractionally k-isomorphic* if and only if the polytope P_k^{gi} is nonempty. Thus, X and X' are fractionally n-isomorphic precisely when they are isomorphic. We now state the main result of Atserias and Maneva [3].

Theorem 10. [3] *Let X and X' be two n-vertex graphs with adjacency matrices A and B. For any $k \geq 0$, if X and X' are fractionally $k+1$-isomorphic then X and X' are indistinguishable by $k+1$-dimensional Weisfeiler-Lehman. Furthermore, if X and X' are indistinguishable by $k+1$-dimensional Weisfeiler-Lehman, then X and X' are fractionally k-isomorphic.*

In [13] it is shown that the interleaving of fractional k-isomorphism and k-WL for different k is, in fact, a strict interleaving, except for equality at the first level, given by Theorem 2.

We recall from Section 1 Tinhofer's definition of compact graphs. A graph X is compact iff $P^{gi} = P^{gi}_{int}$.

An intriguing open question is the complexity of recognizing compact graphs. For the general case we have the following simple complexity-theoretic upper bound.

Fact 11. *Given an n-vertex graph X as input, we can decide if X is compact in* coNP.

Proof. This follows because testing integrality of every vertex of the polytope P^{gi} for X is in coNP. We use the fact that, as the polytope P^{gi} is itself defined by a small LP, testing if a point is a vertex can be done in polynomial time. □

It is shown in [1] that the problem of checking if X is compact is P-hard under logspace reductions. Apart from this we do not have any complexity lower bound for the problem. It is open whether the problem is coNP-hard.

Similar to compactness, we can define a notion of k-compactness w.r.t. the k-level of the Sherali-Adams relaxation.

Definition 12. *A graph X is* k-compact *if $P^{gi}_k = P^{gi}_{int}$.*

Analogous to Tinhofer's observation [21, 22] we note the following.

Theorem 13. *If X is an n-vertex graph that is k-compact then given any other n-vertex graph X' there is an $n^{O(k)}$ time algorithm to check if X and X' are isomorphic.*

Proof. Let A and B be the adjacency matrices of X and X' respectively. Let $P^{gi}_k(A)$ and $P^{gi}_k(B)$ denote the polytopes given by the k^{th} level of the Sherali-Adams hierarchy for X and X'. As per definition $P^{gi}_k(A)$ is the projection of another polytope $\hat{P}^{gi}_k(A)$, where $\hat{P}^{gi}_k(A)$ is defined by an LP of size $n^{O(k)}$ on the variables y_S, for every subset $S \subseteq [n]$ of size at most $k-1$. The variables $y_{\{i\}}$ equal $x_i, 1 \le i \le n$ and $P^{gi}_k(A)$ is defined by projection to these n variables. Suppose X and X' are isomorphic and π is an isomorphism. Let Q be the corresponding permutation matrix. Then we have

$$Q^T A Q = B.$$

I.e. $AQ = QB$. The permutation π extends to all subsets of $[n]$ naturally, where $\pi(S) = \{\pi(i) \mid i \in S\}$. It is easy to see that an $\binom{[n]}{\le k}$-vector v (of values to $y_S, |S| \leq k-1$) is in $\hat{P}_k^{gi}(A)$ iff the vector $u = \pi(v)$ is in $\hat{P}_k^{gi}(B)$, where $u_S = v_{\pi(S)}$ for all $S : |S| \leq k-1$. As $P_k^{gi}(A)$ and $P_k^{gi}(B)$ are obtained by projecting to the n variables $y_{\{i\}}$, it follows that

$$\pi(P_k^{gi}(A)) = P_k^{gi}(B).$$

Hence, if X is k-compact and X' is isomorphic to X, then X' is also k-compact.

Now, consider the polytope $S_k(X, X')$ obtained as the k^{th} level of the Sherali-Adams relaxation of the integer linear program in Fact 1. Assuming X is k-compact, we show that if $S_k(X, X')$ is nonempty that all its vertices are integral. That would immediately yield an $n^{O(k)}$ time isomorphism test because the size of the LP is $n^{O(k)}$. Let P be an extreme point of $S_k(X, X')$. Suppose P is not integral. We know that $AP = PB$ and P is doubly stochastic. Since $Q^T AQ = B$, it follows that $APQ^T = PQ^T A$. Hence, PQ^T is a fractional automorphism of A (which is not integral because Q is a permutation matrix and P is not integral). Since X is compact, we can write PQ^T as a convex combination of integral automorphisms of X. I.e.

$$PQ^T = \sum_{i=1}^{N} \lambda_i P_i,$$

where $\sum_i \lambda_i = 1$ and $0 \leq \lambda_i < 1$ for all i, P_i are permutation matrices and $AP_i = P_i A$ for all i. The nonzero λ_i are strictly less than 1 because we have assumed PQ^T is fractional.

Hence,

$$P = \sum_{i=1}^{N} \lambda_i P_i Q.$$

Now, $AP_iQ = P_iAQ = P_iQB$ for all i. Hence, P_iQ are all integral isomorphisms from X to X'. This contradicts the extremality of P for the polytope $S_k(X, X')$. □

Characterizing k-compact graphs is an interesting open problem.

References

[1] V. Arvind, J. Köbler, G. Rattan, and O. Verbitsky. On Tinhofer's linear programming approach to isomorphism testing. *Proc. 40th MFCS Conference, Part II* LNCS 9235, 26–37, 2015.

[2] V. Arvind, J. Köbler, G. Rattan, and O. Verbitsky. On the power of color refinement. *Proc. 20th FCT Conference,* LNCS 9210, 339-350, 2015.

[3] A. Atserias and E. N. Maneva. Sherali-adams relaxations and indistinguishability in counting logics. *SIAM J. Comput.*, 42(1):112–137, 2013.

[4] L. Babai, P. Erdös, and S. M. Selkow. Random graph isomorphism. *SIAM J. Comput.*, 9(3):628–635, 1980.

[5] L. Babai and E. Luks. Canonical labeling of graphs. In *Proceedings of the 15th STOC Conference,* 171–183, 1983.

[6] L. Babai. Graph isomorphism in quasipolynomial time. *Proc. 48th STOC Conference,*, 684-697, 2016.

[7] J. Cai, M. Fürer and N. Immerman. An optimal lower bound on the number of variables for graph identifications. *Combinatorica* 12(4): 389-410, 1992.

[8] A. Dawar, M. Grohe, B. Holm, and B. Laubner. Logics with Rank Operators. *Proceedings 24th Annual IEEE Symposium on Logic in Computer Science,* 11-14 August 2009, 113–122.

[9] C. Godsil. Compact graphs and equitable partitions. *Linear Algebra and its Applications*, 255(1âĂŞ3):259 – 266, 1997.

[10] M. Furst, J. E. Hopcroft, and E. M. Luks. Polynomial-time algorithms for permutation groups. Proc. 21st IEEE Symp. on Foundations of Comp. Sci. (1980), 36-41.

[11] M. Grohe. Fixed-Point Logics on Planar Graphs. *13th IEEE LICS Conference,* 1998, 6–15.

[12] M. Grohe. Fixed-point definability and polynomial time on graphs with excluded minors. *Journal of the ACM,* 59(5), article 27, 2012.

[13] M. Grohe and M. Otto. Pebble games and linear equations. *26th Computer Science Logic Conference,* volume 16 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 289–304, Dagstuhl, Germany, 2012.

[14] M. Grohe, O. Verbitsky. Testing Graph Isomorphism in Parallel by Playing a Game. *Proc. 33rd ICALP,* 2006, 3 – 14.

[15] N. Immerman and E. Lander. Describing graphs: A first-order approach to graph canonization. In A. Selman, editor, *Complexity Theory Retrospective*, pages 59–81. Springer, 1990.

[16] Ludek Kucera. Canonical Labeling of Regular Graphs in Linear Average Time. *28th Annual FOCS Conference,*1987, 271–279.

[17] S. Lindell. A Logspace Algorithm for Tree Canonization. *Proc. 24th ACM Symposium on Theory of Computing,* 1992, pp. 400–404.

[18] B. McKay. Nauty User's Guide (version 1.5). *Tech. Rpt. TR-CS-90-02,* Dept. Computer Science, Austral. Nat. Univ. (1990).

[19] M. V. Ramana, E. R. Scheinerman, and D. Ullman. Fractional isomorphism of graphs. *Discrete Mathematics*, 132(1-3):247–265, 1994.

[20] S. Kiefer, P. Schweitzer, E. Selman. Graphs identified by logics with counting. *Proc. 40th MFCS Conference,* LNCS 9234, 319-330, 2015.

[21] G. Tinhofer. Graph isomorphism and theorems of birkhoff type. *Computing*, 36:285–300, 1986.

[22] G. Tinhofer. A note on compact graphs. *Discrete Applied Mathematics*, 30(2-3):253–264, 1991.

[23] B. Weisfeiler, A.A. Lehman. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsiya,* Ser. 2, no. 9 (1968), 12-16.

The Distributed Computing Column

BY

Stefan Schmid

Aalborg University
Selma Lagerlöfs Vej 300, DK-9220 Aalborg, Denmark

This time, the Distributed Computing Column features two articles:

1. Gianlorenzo D'Angelo presents an interesting optimization perspective on graph centralities and surveys the state-of-the-art approximation bounds. Gianlorenzo D'Angelo also receives the Best Young Italian TCS Researcher Award for 2016 of the Italian Chapter of the EATCS. Congratulations!

2. In an effort to shed light on the consequences of Artificial Intelligence and Computerization on employment, Philipp Brandes and Roger Wattenhofer present an interesting refinement of a seminal study by the economists Frey and Osborne. In particular, Brandes and Wattenhofer's probabilistic model accounts for the unique tasks of each job, allowing us to look inside the Frey/Osborne blackbox, and giving rise to a number of interesting insights and discussions.

Approximation bounds for centrality maximization problems

Gianlorenzo D'Angelo
Gran Sasso Science Institute (GSSI).
Viale F. Crispi, 7, 67100, L'Aquila, Italy.
gianlorenzo.dangelo@gssi.infn.it

Abstract

Determining what are the most important nodes in a network is one of the main problems in the field of network analysis. Several so-called *centrality indices* have been defined in the literature to try to quantitatively capture the notion of importance (or centrality) of a node within a network. It has been experimentally observed that being central for a node, according to some centrality index, leads to several benefits to the node itself.

In this paper, we study the problem of maximizing the centrality index of a given node by adding a limited number of edges incident to it. We survey on some recent results on this problem by focusing on four well-known centrality indices, namely harmonic centrality, betweenness centrality, eccentricity, and page-rank.

1 Introduction

In the past decades, there has been an increasing interest in the analysis of real-world complex networks in diverse research areas from sociology to computer science, going through biology and economy. Relevant examples of networks are autonomous-systems networks within the Internet, the World Wide Web, networks deriving from transportation infrastructures like roads or public transport, networked energy systems, social networks, coauthorship networks, and financial systems. It is somewhat surprising to observe that several networks originating from different contexts exhibit similar structural properties.

One of the most studied network properties goes under the name of *centrality* of a node in a network. Informally speaking, a node is considered "central" if it is important within the network and it is believed that the importance that a node has within a network reflects, to some extent, the position of the node in the network and, more in general, the network structure. However, researchers do not agree on a common definition of centrality, instead several *centrality indices* have been proposed in the literature to try to quantitatively capture this notion. Most of the centrality indices are based on distances between nodes (like the *closeness centrality* [2]), on the number of shortest paths passing thorough a node (like the *betweenness centrality* [17]), or on spectral properties (like the *page-rank* [8]). For more details on centrality indices, see [5, 30]. What is the right definition of centrality of a node is not clear and the choice depends on the application domain.

On the other hand, it has been experimentally observed that being central for a node, according to some centrality index, has several benefits for the node itself. For example, closeness centrality is significantly correlated with citation counts of an author in author-citation networks [36], betweenness centrality is correlated with the efficiency of an airport in transportation networks [28], and both closeness and betweenness are correlated with the efficiency of an individual to propagate the information in a social network [27]. Therefore, a lot of research effort has been done on the problems of computing the centrality indices of a given node or determining the most central nodes of a network, according to some index.

In this paper we look at centrality indices from a *proactive* point of view, that is we want to *modify* an existing network with the aim of improving the centrality of a given node. A network can be modified by adding or removing edges and nodes. By performing these operations the centrality of a node can increase, while the centrality of other nodes can decrease. For example by adding edges, the distances between nodes decreases and hence the closeness centrality of some node increases, while by removing edges the closeness centrality might decrease.

Which "strategy" should a node adopt in order to increase its own centrality value as much as possible? In this paper we formulate this question as an optimization problem which consists in finding a limited amount of edges to be added in a graph in order to maximize the centrality of a given node within a network.

Generally speaking, adding edges incident to a given node v reduces the distances between v and the other nodes and hence it increases the centrality of v in some centrality indices. Moreover, looking at social networks from a user (node) perspective, it is not difficult to imagine scenarios in which a node can only add edges incident to itself and hence it is reasonable to consider such constraint in our optimization problem. More specifically, we consider the problem of efficiently determining, for a given vertex v, the set of k edges incident to v that, when added to the original graph, maximizes the centrality of v, according to some index. We denote this optimization problem as Centrality Maximization problem (CM). In this paper we survey some recent results on the CM problem in which we use four relevant centrality indices to be maximized: harmonic centrality, betweenness centrality, eccentricity and page-rank. The results outlined in this paper are reported in Table 1.

Structure of the paper. In the next section, we give the notation used in the paper, define the centrality indices that we aim at maximizing, and give the problem statement. In Section 3, we survey on the known results on the CM problem. Finally, in Section 4, we outline some future research directions that deserve further investigation.

2 Preliminaries

Let $G = (V, E)$ be a directed or undirected graph. For each node v, if G is directed, N^i_v and N^o_v denote the set of in-neighbors and out-neighbors of v, respectively, i.e. $N^i_v = \{u \mid (u, v) \in E\}$ and $N^o_v = \{u \mid (v, u) \in E\}$. If G is undirected, N_v denotes the set of all neighbors of v, $N_v = \{u \mid \{u, v\} \in E\}$. Given two nodes s and t, we denote by d_{st}, σ_{st}, and σ_{stv} the distance from s to t in G, the number of shortest paths from s to t in G, and the number of shortest paths from s to t in G that contain v, respectively. When we discuss about page-rank, we will assume that the graph is strongly connected.

2.1 Centrality indices

A *centrality index* c (also called *centrality metrics* or *centrality measures*) is a function $c : V \to \mathbb{R}$ that associates a number to each node according to the importance of the node, that is if node v is at least as important as node u, then $c_v \geq c_u$. A centrality index induces a partial ordering of the nodes in V. The *ranking* of a node v according to some centrality index c is the placement of v in the ordering induced by c and it is defined as

$$r^c_v = |\{u \in V \mid c_u > c_v\}| + 1.$$

According to [4], centrality indices can be classified into three non-disjoint categories: *geometric* indices, *path-based* indices, and *spectral* indices. The first category includes all those measures that evaluate the importance of a node on the basis of a function of the distances from the node to any other node, more in details, a geometric index depends only on how many nodes exist at every distance from the given node. Examples of geometric indices are: node degree, closeness centrality [2], Lin's index [26], harmonic centrality [4], and eccentricity. Instead of considering distances to a node, path-based indices take into account all the shortest paths (or all simple paths) passing through a node. Examples in this category are stress centrality [34], betweenness centrality [6, 17] and its variants [7]. Spectral indices evaluate the importance of a node on the basis of the left dominant eigenvector of a matrix derived from the graph. Examples of spectral indices are: Katz' index [22], page-rank [8], and HITS [25].

In this paper we study the problem of augmenting a graph in order to maximize the centrality of a node according to some index. We focus on four relevant centrality indices that are representative of the above categories. In what follows we define such centrality indices.

- The *harmonic centrality* [4] of a node v is defined as the harmonic mean of the distances from all the other nodes to v, formally:
$$h_v = \sum_{\substack{s \in V \setminus \{v\} \\ d_{sv} < \infty}} \frac{1}{d_{sv}}.$$
- The *betweenness centrality* [6, 17] of a node v is defined as the sum over all pairs of nodes (s, t) of the ratio between the number of shortest path from s to t passing through v and all the shortest paths from s to t that is:
$$b_v = \sum_{\substack{s,t \in V \\ s \neq t; s,t \neq v \\ \sigma_{st} \neq 0}} \frac{\sigma_{stv}}{\sigma_{st}}.$$
- The *eccentricity* of a node v is the maximum distance between v and any other node, that is
$$e_v = \max_{u \in V}\{d_{uv}\}.$$
Note that, in this case a node is central if its eccentricity is small.
- In a directed graph, the *page-rank* of a node v is the probability that a *random surfer walk* that starts at a random node in a graph is at v at a given point in time. A random surfer walk with parameter α, is a walk in the graph defined as follows: start at a random node in G, given by a starting probability distribution; with probability α, move to an edge chosen uniformly at random from those outgoing the current node; with probability $1 - \alpha$, move directly to another node that might be not connected to the current node. In this latter case, the next node node is chosen by according to the starting probability distribution.

  Formally, let us assume that G is a strongly connected directed graph. Let M be a $|V| \times |V|$ matrix where each element m_{uv} is defined as $m_{uv} = \frac{1}{|N_u^o|}$ if $(u, v) \in E$ and $m_{uv} = 0$ otherwise. For a given parameter α, the page-rank is the eigenvector $\bar{p}$ associated to the largest eigenvalue of the matrix
$$Q = \frac{1 - \alpha}{|V|}\mathbb{1} + \alpha M.$$
  The page rank of a node v is the element p_v in the position associated to v in $\bar{p}$.

2.2 Problem statement

Given a set S of edges not in E, we denote by $G(S)$ the graph augmented by adding the edges in S to G, i.e. $G(S) = (V, E \cup S)$. For a parameter x of G, we denote by $x(S)$ the same parameter in graph $G(S)$, e.g. the distance from s to t in $G(S)$ is denoted as $d_{st}(S)$. The centrality index of a node v clearly depend on the graph structure: if we augment a graph by adding a set of edges S incident to v, then the centrality of v might change. Generally speaking, adding edges incident to some node v can only increase the centrality of v. We are interested in finding a set S of edges incident to a particular node v that maximizes such an increment. Therefore, given a centrality index c, we define the following optimization problem.

	Centrality Maximization (CM)
Given:	A directed or undirected graph $G = (V, E)$; a node $v \in V$; and an integer $k \in \mathbb{N}$
Solution:	A set S of edges incident to v, $S = \{(u, v) \mid u \in V \setminus N_v^i\}$ ($S = \{\{u, v\} \mid u \in V \setminus N_v\}$, if G is undirected), such that $\lvert S \rvert \leq k$
Goal:	Maximize $c_v(S)$

We study the CM problem by using harmonic centrality, betweenness centrality, eccentricity, and page-rank as indices, obtaining problems CM-H, CM-B, CM-E, CM-P.

2.3 Maximizing monotone submodular functions

Some of the algorithms reported in this paper, exploit the results of Nemhauser et al. on the approximation of monotone submodular objective functions [29]. A function z defined on subsets of a ground set N, $z : 2^N \to \mathbb{R}$, is *submodular* if the following inequality holds for any pair of sets $S \subseteq T \subseteq N$ and for any element $e \in N \setminus T$

$$z(S \cup \{e\}) - z(S) \geq z(T \cup \{e\}) - z(T).$$

In other words, a submodular function exhibits decreasing marginal gains: the marginal value of adding a new element to a set decreases as the set increases. Let us consider the following optimization problem: given a finite set N, an integer k', and a real-valued function z defined on the set of subsets of N, find a set $S \subseteq N$ such that $\lvert S \rvert \leq k'$ and $z(S)$ is maximum. If z is *monotone and submodular*, then the following greedy algorithm exhibits an approximation of $1 - \frac{1}{e}$ [29]: start with the empty set, and, for k' iterations, add an element that gives the maximal marginal gain, that is if S is a partial solution, choose the element $j \in N \setminus S$ that maximizes $z(S \cup \{j\})$.

Theorem 1 ([29]). *For a non-negative, monotone submodular function z, let S be a set of size k obtained by selecting elements one at a time, each time choosing*

Algorithm 1: Greedy algorithm for CM on directed graphs.
Input : A directed graph $G = (V, E)$; a node $v \in V$; and an integer $k \in \mathbb{N}$
Output: Set of edges $S \subseteq \{(u, v) \mid u \in V \setminus N_v^i\}$ such that $|S| \leq k$

$S := \emptyset$;
for $i = 1, 2, \ldots, k$ **do**
  foreach $u \in V \setminus N_v^i(S)$ **do**
    Compute $c_v(S \cup \{(u, v)\})$
  $u_{\max} := \arg\max\{c_v(S \cup \{(u, v)\}) \mid u \in V \setminus N_v^i(S)\}$;
  $S := S \cup \{(u_{\max}, v)\}$;
return S;

---

an element that provides the largest marginal increase in the value of z. Then S provides a $\left(1 - \frac{1}{e}\right)$*-approximation.*

In this paper, we exploit such results by showing that some centrality indices c are monotone and submodular with respect to the possible set of edges incident to a given node v. Hence, the greedy algorithm in Algorithm 1 provides a $\left(1 - \frac{1}{e}\right)$-approximation for CM.[1] Algorithm 1 iterates k times and, at each iteration, it adds to an initially empty solution S an edge (u, v) (or $\{u, v\}$ in the case of undirected graph) that, when added to $G(S)$, gives the largest marginal increase in the centrality of v, that is $c(S \cup \{(u, v)\})$ ($c(S \cup \{\{u, v\}\})$, respectively) is maximum among all the possible edges not in $E \cup S$ incident to v. This technique will be used for harmonic centrality, betweenness centrality, and page-rank.

3 Centrality maximization

In this section we study the CM problem for harmonic centrality, betweenness centrality, eccentricity, and page-rank. For each problem we will give both hardness of approximation results and approximation algorithms. In order to highlight the main ideas and techniques, we will give only proof sketches and references to the complete proofs.

3.1 Harmonic centrality

We now report the results for the CM-H problem, more details on these results can be found in [10]. We first show the hardness of approximation results for the

[1] Algorithm 1 can be easily modified to work in the case of undirected graphs.

undirected and directed graph cases and then give an approximation algorithm for both cases.

To derive an approximation hardness result for the undirected case, we make use of the *Minimum Dominating Set* (in short, MDS) problem, which is defined as follows: given an undirected graph $G = (V, E)$, find a *dominating set* of minimum cardinality, that is, a subset D of V such that $V = D \cup \bigcup_{u \in D} N_u$. It is known that, for any r with $0 < r < 1$, it cannot exist a $(r \ln |V|)$-approximation algorithm for the MDS problem, unless $P = NP$ [14]. We now use this result in order to show that the CM-H problem does not admit a polynomial-time approximation scheme. To this aim, we design an algorithm A' that, given an undirected graph $G = (V, E)$ and given the size k of the optimal dominating set of G, by using an approximation algorithm A for the CM-H problem returns a dominating set of G whose approximation ratio is at most $(r \ln |V|)$. Clearly, we do not know the value of k, but we know that this value must be at least 1 and at most $|V|$: hence, we run algorithm A' for each possible value of k, and return the smallest dominating set found. Algorithm A' runs the approximation algorithm A for the CM-H problem multiple times. Each time A finds k nodes $u \in V$ which are the "new" neighbours of the node whose centrality has to be increased: we then add these nodes to the dominating set and create a smaller instance of the CM-H problem (which contain, among the others, all the nodes in V not yet dominated). We continue until all nodes in V are dominated.

Algorithm A' is specified in Fig. 2, where k denotes a "guess" of the size of an optimal solution for MDS with input the graph G. In the following, ω denotes the number of times the while loop is executed. Since, at each iteration of the loop, we include in the dominating set at most k nodes, at the end of the execution of algorithm A' the set D includes at most $k \cdot \omega$ nodes. Hence, if k is the correct guess of the value of the optimal solution for the MDS instance, then D is a ω-approximate solution for the MDS problem (as we have already noticed, we do not know the correct value of k, but algorithm A' can be executed for any possible value of k, that is, for each $k = 1, 2, \dots, |V|$).

The first instruction of the while loop of algorithm A' computes a transformed graph G' (to be used as part of the new instance for CM-H) starting from the current graph $G = (V, E_V)$, which is the subgraph of the original graph induced by the set $\{u_1, \dots, u_n\}$, where $n = |V|$, of still not dominated nodes. This computation is done as follows. We add a new node z and two new nodes x_i and y_i, for each i with $1 \le i \le n$. Moreover, we add to E_V the edges $\{z, y_i\}$, $\{x_i, y_i\}$, and $\{x_i, u_i\}$, for each i with $1 \le i \le n$. As it is shown in the second line of the while loop, z is the node whose harmonic centrality h_z has to be increased by adding at most k edges: that is, the CM-H instance is formed by G', z, and k. We can assume that the solution S computed at the second line of the while loop of algorithm A' contains only edges connecting z to nodes in V (see [10] for details).

Algorithm 2: The approximation algorithm A' for the MDS problem, given a γ-approximation algorithm A for the CM-H problem and a "guess" k for the optimal value of MDS.

Input : an undirected graph $G = (V, E)$ and an integer k
Output: a dominating set D

$D := \emptyset$;
while $V \neq \emptyset$ **do**
  Compute graph G' starting from G;
  $S := A(G', z, k)$;
  $D' := \{u : \{z, u\} \in S\}$
  $D := D \cup D'$;
  $V := V \setminus (D' \cup \bigcup_{u \in D'} N_u)$;
  $G :=$ subgraph of G induced by V;
return D;

First of all, note that, since k is (a guess of) the measure of an optimal solution D^* for MDS with input G, we have that the measure $h^*(G', z, k)$ of an optimal solution S^* for CM-H with input G' satisfies the following inequality:

$$h^*(G', z, k) \geq k + \frac{1}{2}(n - k) + \frac{3}{2}n = \frac{1}{2}k + 2n.$$

This is due to the fact that, by connecting z to all the k nodes in D^*, in the worst case we have that k nodes in G are at distance 1, $n - k$ nodes in G are at distance 2 (since D^* is a dominating set), the n nodes y_i are at distance 1, and the n nodes x_i are at distance 2 from z.

Given the solution S computed by the approximation algorithm A for CM-H, let a and b denote the number of nodes in G at distance 2 and 3, respectively, from z in $G'(S)$. Since all nodes in G' are at distance at most 3 from z, we have that $n = k + a + b$ (we can assume, without loss of generality, that $n \geq k$): hence, $a = n - b - k$. Since A is a γ-approximation algorithm for CM-H, we have that $h_z(S) \geq \gamma h^*(G', z, k)$. That is, $k + \frac{1}{2}a + \frac{1}{3}b + \frac{3}{2}n \geq \gamma\left(\frac{1}{2}k + 2n\right)$. From this inequality, by doing some algebraic computation that use the fact that $a = n - b - k$ and $k \leq n$, we obtain $b \leq 15n(1 - \gamma)$.

Assuming $\gamma > 1 - \frac{1}{15e} > \frac{14}{15}$ (which implies $15(1 - \gamma) < 1$), then after one iteration of the while loop of algorithm A', the number of nodes in G decreases by a factor $15(1 - \gamma)$. Hence, after $\omega - 1$ iterations, the number n of nodes in the graph G is at most a fraction $[15(1-\gamma)]^{\omega-1}$ of the number N of nodes in the original graph. Since we can stop as soon as $n < k$, we need to find the maximum value of ω such that $k \leq N[15(1 - \gamma)]^{\omega-1}$. By solving this inequality and by recalling that

$15(1-\gamma) < 1$, we obtain

$$\omega - 1 \le \log_{15(1-\gamma)} \frac{k}{N} \le \log_{15(1-\gamma)} \frac{1}{N} = \frac{\ln(N)}{\ln \frac{1}{15(1-\gamma)}}.$$

One more iteration might be necessary to trivially deal with the remaining nodes, which are less than k. Hence, the total number ω of iterations is at most $\frac{\ln(N)}{\ln \frac{1}{15(1-\gamma)}} + 1$. If $\gamma > 1 - \frac{1}{15e}$, then the solution reported by algorithm A' is an $(r' \ln N + 1)$-approximate solution, where $r' = \frac{1}{\ln \frac{1}{15(1-\gamma)}} < 1$. Clearly, for any r with $0 < r' < r < 1$, there exists a number $N^{(r)}$ sufficiently large, such that for any $N > N^{(r)}$, $r' \ln N + 1 \le r \ln N$: hence, algorithm A' would be an $r \ln N$-approximation algorithm for MDS, and, because of the result of [14], P would be equal to NP. Thus, we have that, if $P \neq NP$, then γ has to be not greater than $1 - \frac{1}{15e}$. The next theorem follows.

Theorem 2 ([10]). *The* CM-H *problem on undirected graphs cannot be approximated within a factor greater than* $1 - \frac{1}{15e}$*, unless* $P = NP$.

We now focus on the directed case and show that also in this case the CM-H problem cannot be approximated within a certain constant upper bound, unless $P = NP$. We make use of the *Maximum Set Coverage* (in short, MSC) problem, which is defined as follows: given a set X, a collection $\mathcal{F}\}$ of subsets of X, and an integer k, find a sub-collection $\mathcal{F}' \subseteq \mathcal{F}$ such that $|\mathcal{F}'| \le k$ and $s(\mathcal{F}') = |\cup_{S_j \in \mathcal{F}'} S_j|$ is maximized. It is known that the MSC problem cannot be approximated within a factor greater than $1 - \frac{1}{e}$, unless $P = NP$ [16].

In this case we follow the scheme of L-reductions [35, Chapter 16]. In detail, we will give a polynomial-time algorithm that transforms any instance I_{MSC} of MSC into an instance $I_{\text{CM-H}}$ of CM-H and a polynomial-time algorithm that transforms any solution S for $I_{\text{CM-H}}$ into a solution $\mathcal{F}'$ for I_{MSC} such that the following two conditions are satisfied for some constants a and b:

$$OPT(I_{\text{CM-H}}) \le aOPT(I_{\text{MSC}}) \tag{1}$$

$$OPT(I_{\text{MSC}}) - s(\mathcal{F}') \le b\,(OPT(I_{\text{CM-H}}) - h_v(S))\,. \tag{2}$$

where OPT denotes the optimal value of an instance of an optimization problem. If the above conditions are satisfied and there exists a α-approximation algorithm for CM-H, then there exists a $(1 - ab(1-\alpha))$-approximation algorithm for MSC [35, Chapter 16]. Since MSC is hard to approximate within a factor greater than $1 - \frac{1}{e}$, then $1 - ab(1-\alpha) < 1 - \frac{1}{e}$, unless $P = NP$. This implies that, if $P \neq NP$, $\alpha < 1 - \frac{1}{abe}$.

Given an instance $I_{\text{MSC}} = (X, \mathcal{F}, k)$ of MSC, we define an instance $I_{\text{CM}} = (G, v, k)$, where $G = (V, E)$, $V = \{v\} \cup \{v_{x_i} \mid x_i \in X\} \cup \{v_{S_j} \mid S_j \in \mathcal{F}\}$, and $E = \{(v_{x_i}, v_{S_j}) \mid x_i \in S_j\}$.

Without loss of generality, we can assume that any solution S of CM-H contains only edges (v_{S_j}, v) for some $S_j \in \mathcal{F}$ (see [10] for details). Given a solution S of CM-H, let $\mathcal{F}'$ be the solution of MSC such that $S_j \in \mathcal{F}'$ if and only if $(v_{S_j}, v) \in S$. We now show that $h_v(S) = \frac{1}{2}s(\mathcal{F}') + k$. To this aim, let us note that the distance from a vertex v_{x_i} to v is equal to 2 if an edge (x_{S_j}, v) such that $x_i \in S_j$ belongs to S, and it is ∞ otherwise. Similarly, the distance from a vertex v_{S_j} to v is equal to 1 if $(x_{S_j}, v) \in S$, and it is ∞ otherwise. Moreover, the set of elements x_i of X such that $d_{v_{x_i}v}(S) < \infty$ is equal to $\{x_i \mid x_i \in S_j \wedge (v_{S_j}vu) \in S\} = \bigcup_{S_j \in \mathcal{F}'} S_j$. Therefore,

$$\begin{aligned} h_v(S) = \sum_{\substack{u \in V \setminus \{v\} \\ d_{uv}(S) < \infty}} \frac{1}{d_{uv}(S)} &= \sum_{\substack{x_i \in X \\ d_{v_{x_i}v}(S) < \infty}} \frac{1}{d_{v_{x_i}v}(S)} + \sum_{\substack{S_j \in \mathcal{F} \\ d_{v_{S_j}v}(S) < \infty}} \frac{1}{d_{v_{S_j}v}(S)} \\ &= \frac{1}{2}|\{x_i \in X \mid d_{v_{x_i}v}(S) < \infty\}| + |\{S_j \in \mathcal{F} \mid d_{v_{S_j}v}(S) < \infty\}| \\ &= \frac{1}{2}\left|\bigcup_{S_j \in \mathcal{F}'} S_j\right| + |\{S_j \mid (v_{S_j}, v) \in S\}| = \frac{1}{2}s(\mathcal{F}') + k. \end{aligned}$$

It follows that Conditions (1) and (2) are satisfied for $a = \frac{3}{2}$ and $b = 2$. Indeed, $OPT(I_{\text{CM-H}}) = \frac{1}{2}OPT(I_{\text{MSC}}) + k \leq \frac{3}{2}OPT(I_{\text{MSC}})$, where the inequality is due to the fact that $OPT(I_{\text{MSC}}) \geq k$, since otherwise the greedy algorithm would find an optimal solution for I_{MSC}. Moreover, $OPT(I_{\text{MSC}}) - s(\mathcal{F}') = 2\,(OPT(I_{\text{CM-H}}) - k) - 2\,(h_v(S) - k) = 2\,(OPT(I_{\text{CM-H}}) - h_v(S))$. The next theorem follows by plugging the values of a and b into $\alpha < 1 - \frac{1}{abe}$.

Theorem 3 ([10])**.** *The* CM-H *problem on directed graphs cannot be approximated within a factor greater than* $1 - \frac{1}{3e}$*, unless* $P = NP$.

In the following we show that h_u is monotone and submodular in the case of undirected graphs, the proof can be easily adapted to the case in which the graphs are directed. To simplify the notation, we assume that $\frac{1}{\infty} = 0$. To show that h_v is monotone increasing, it is enough to observe that, for each solution S to CM-H, for each edge $\{u, v\} \notin E \cup S$, and for each node $x \in V \setminus \{v\}$, $d_{vx}(S \cup \{\{u, v\}\}) \leq d_{vx}(S)$ (since adding an edge cannot increase the distance between two nodes) and, therefore, $\frac{1}{d_{vx}(S \cup \{\{u,v\}\})} \geq \frac{1}{d_{vx}(S)}$.

To prove that h_v is submodular, we show that, for each pair S and T of solutions for CM-H such that $S \subseteq T$, and for each edge $\{u, v\} \notin T \cup E$,

$$h_v(S \cup \{\{u, v\}\}) - h_v(S) \geq h_v(T \cup \{\{u, v\}\}) - c_u(T).$$

To this aim, we prove that each term of h_u is submodular, that is, for each vertex $x \in V \setminus \{v\}$,

$$\frac{1}{d_{vx}(S \cup \{\{u, v\}\})} - \frac{1}{d_{vx}(S)} \geq \frac{1}{d_{vx}(T \cup \{\{u, v\}\})} - \frac{1}{d_{vx}(T)}. \tag{3}$$

Let us consider the shortest paths from v to x in $G(T \cup \{\{u, v\}\})$, and let us distinguish the following two cases.

1. The first edge of a shortest path from v to x in $G(T \cup \{\{u, v\}\})$ is $\{u, v\}$ or belongs to $S \cup E$. In this case, such a path is a shortest path also in $G(S \cup \{\{u, v\}\})$, as it cannot contain edges in $T \setminus S$ (since these edges are all incident to v). Then, $d_{vx}(S \cup \{\{u, v\}\}) = d_{vx}(T \cup \{\{u, v\}\})$ and $\frac{1}{d_{vx}(S\cup\{\{u,v\}\})} = \frac{1}{d_{vx}(T\cup\{\{u,v\}\})}$. Moreover, $d_{vx}(S) \geq d_{vx}(T)$ (since $S \subseteq T$) and, therefore, $-\frac{1}{d_{vx}(S)} \geq -\frac{1}{d_{vx}(T)}$.

2. The first edge of all shortest paths from v to x in $G(T \cup \{\{u, v\}\})$ belongs to $T \setminus S$. In this case, $d_{vx}(T) = d_{vx}(T \cup \{\{u, v\}\})$ and, therefore, $\frac{1}{d_{vx}(T\cup\{\{u,v\}\})} - \frac{1}{d_{vx}(T)} = 0$. As $\frac{1}{d_{vx}(S)}$ is monotone increasing, then $\frac{1}{d_{vx}(S\cup\{\{u,v\}\})} - \frac{1}{d_{vx}(S)} \geq 0$.

In both cases, we have that the inequality (3) is satisfied and, hence, the next theorem follows.

Theorem 4 ([10])**.** *In both directed and undirected graphs, for each vertex u, function h_u is monotone and submodular with respect to any feasible solution for* CM-H*.*

Theorems 1 and 4 imply the next corollary.

Corollary 5. *The* CM-H *problem is approximable within a factor $1 - \frac{1}{e}$ in both directed and undirected graphs.*

3.2 Betweenness centrality

We now show that problem CM-B is hard to be approximated within a certain constant upper bound, that in the case of directed graphs the objective function is monotone and submodular, and that there are instances of the undirected case for which the greedy algorithm exhibits an arbitrarily small approximation ratio. We omit proof sketches for the first two results as the arguments are similar to those of Theorems 3 and 4, respectively. Full proofs of the results stated in this section can be found in [9, 12].

We observe that the next result for undirected graphs has been proven only for the case in which edges are weighted [12].

Theorem 6 ([9, 12])**.** *The* CM-B *problem on both directed and undirected graphs cannot be approximated within a factor greater than $1 - \frac{1}{2e}$, unless $P = NP$.*

Theorem 7 ([9])**.** *In directed graphs, for each vertex v, function b_v is monotone and submodular with respect to any feasible solution for* CM-B*.*

Corollary 8. *In directed graphs, the* CM-B *problem is approximable within a factor* $1 - \frac{1}{e}$.

We now prove that, differently from the directed case and from the case of harmonic centrality, the approximation ratio of a greedy solution for CM-B in the case of undirected graphs does not have a constant lower bound. To this aim, let us consider the following instance of CM-B.

- Graph $G = (V, E)$.
- $V = \{v, t, a, b, c, a', b', c'\} \cup A \cup B \cup C$, where $A = \{a_i\}_{i=1}^{y}$, $B = \{b_i\}_{i=1}^{x}$, $C = \{c_i\}_{i=1}^{y}$, and $y = x - 2$, for some $x > 2$;
- $E = \{\{v, t\}, \{a, b\}, \{b, c\}, \{a, a'\}, \{b, b'\}, \{c, c'\}, \{a', t\}, \{b', t\}, \{c', t\}\} \cup \{\{a_i, a\} \mid a_i \in A\} \cup \{\{b_i, b\} \mid b_i \in B\} \cup \{\{c_i, c\} \mid c_i \in C\}$;
- $k = 2$.

The initial value of b_v is zero. The greedy algorithm first chooses edge $\{b, v\}$ and then edge $\{a_i, v\}$, for some $a_i \in A$ (or equivalently $\{c_i, v\}$, for some $c_i \in A$). The value of $b_v(\{b, v\}, \{a_i, v\})$ is $2x + 3$. In fact, the following pairs have shortest paths passing through v in $G(\{b, v\}, \{a_i, v\})$: nodes in $B \cup \{b\}$ and t ($x + 1$ shortest paths), a_i and t (1 shortest path), a_i and nodes in $B \cup \{b\}$ ($\frac{x+1}{2}$ shortest paths), a_i and nodes in $C \cup \{c\}$ ($\frac{y+1}{2}$ shortest paths), and a_i and c' (1 shortest path). An optimal solution, instead, is made of edges $\{a, v\}$ and $\{c, v\}$ where $b_v(\{\{a, v\}, \{c, v\}\}) = \frac{x^2+3x-2}{2}$, where the quadratic term comes from the fact that there are $(y+1)^2$ paths passing through v between nodes in $A \cup \{a\}$ and nodes in $C \cup \{c\}$. Therefore, the approximation ratio of the greedy algorithm tends to be arbitrarily small as x increases. We observe that the bad approximation ratio of the greedy algorithm is due to the fact that it does not consider the shortest paths that pass through v by using both edges. The next proposition follows.

Proposition 9 ([12]). *In undirected graphs, the greedy algorithm exhibits an unbounded approximation ratio.*

3.3 Eccentricity

We now report the results on the CM-E problem, more details can be found in [13, 32]. Note that in this case a node is considered central if its eccentricity is small, therefore the CM-E problem is a minimization problem, that is we want to find the set of edges S that, when added to G, minimizes the value of $e_v(S)$, for some given node v. We first show that, unless $P = NP$, the problem cannot be approximated within a certain constant lower bound, we then give an algorithm that guarantees a constant approximation ratio and an algorithm that guarantees an arbitrarily small approximation ratio if an higher number of edges is allowed.

To derive an approximation hardness result for the undirected case, we make use of the *Set Cover* (in short, sc) problem, which is defined as follows: given a set X, a collection $\mathcal{F}\}$ of subsets of X, and an integer B, find a sub-collection $\mathcal{F}' \subseteq \mathcal{F}$ such that $\cup_{S_j \in \mathcal{F}'} S_j = X$ and $|\mathcal{F}'| \leq B$. It is known that the set cover problem is NP-hard [18].

Given an instance $(X, \mathcal{F})$ of sc, we compute a graph $G = (V, E)$, where $V = \{v, v'\} \cup \{v_{x_i} \mid x_i \in X\} \cup \{v_{S_j} \mid S_j \in \mathcal{F}\}$ and $E = \{\{v, v'\}\} \cup \{\{v', v_{S_j}\} \mid S_j \in \mathcal{F}\} \cup \{\{v_{x_i}, v_{S_j}\} \mid x_i \in S_j\}$. Initially, the eccentricity of v is equal to 3. We prove that there exists a feasible solution for an instance $I_{\text{sc}} = (X, \mathcal{F})$ of sc if and only if there exists a solution S for the instance $I_{\text{cm-e}} = (G, v, k)$, where $k = B$, of cm-e such that $e_v(S) = 2$.

If I_{sc} admits a feasible solution $\mathcal{F}'$, then let us consider the solution $S = \{\{v, v_{S_j}\} \mid S_j \in \mathcal{F}'\}$ to $I_{\text{cm-e}}$. Since $|\mathcal{F}'| \leq B$, then $|S| \leq k$. Moreover, $\cup_{S_j \in \mathcal{F}'} S_j = X$ and then all the nodes v_{x_i} are at distance 2 to v. Therefore, $e_v(S) = 2$.

Let us now assume that $I_{\text{cm-e}}$ admits a solution S such that $e_v(S) = 2$, without loss of generality, we can assume that S contains only edges $\{v, v_{S_j}\}$ for some $S_j \in \mathcal{F}$ (see [13] for details). Let $\mathcal{F}'$ be the solution of sc such that $S_j \in \mathcal{F}'$ if and only if $\{v, v_{S_j}\} \in S$. Since $e_v(s) = 2$, the distance between v and all the nodes v_{x_i} is at most 2 and then for each v_{x_i} there exists an edge $\{v, v_{S_j}\} \in S$ such that $x_i \in S_j$. This implies that $\cup_{S_j \in \mathcal{F}'} S_j = X$. Moreover, since $|S| \leq k$, then $|\mathcal{F}'| \leq B$.

Let us assume that there exists an approximation algorithm A for cm-e that guarantees an approximation factor $\alpha < \frac{3}{2}$ and let S be the solution obtained by applying algorithm A to $I_{\text{cm-e}}$ derived from I_{sc}. We have that $e_v(S) < \frac{3}{2} OPT$. This implies that, if $(X, \mathcal{F})$ admits a feasible solution, then $e_v(S) < \frac{3}{2} \cdot 2 = 3$, that is $e_v(S) = 2$; otherwise, if $(X, \mathcal{F})$ does not admit a feasible solution, then $e_v(S) = 3$. Therefore, we can determine whether an instance of sc is feasible or not by means of algorithm A. The next theorem follows.

Theorem 10 ([13])**.** *The* cm-e *problem on undirected graphs cannot be approximated within a factor smaller than $\frac{3}{2}$, unless $P = NP$.*

In what follows we describe the algorithm given in [32] to solve the cm-e problem in undirected graphs. The algorithm is based on a former solution to the problem of minimizing the diameter of a graph by adding a limited number of edges [3].

The algorithm is reported in Algorithm 3 and works as follows: first node v is inserted into a set U, then, a for loop of k iteration is run. At each iteration $i = 1, 2, \ldots, k$, a node u_i that maximizes the minimum distance in G between u_i

---

Algorithm 3: Approximation algorithm for CM-E.

Input : An undirected graph $G = (V, E)$; a node $v \in V$; and an integer $k \in \mathbb{N}$

Output: Set of edges $S \subseteq \{\{u, v\} \mid u \in V \setminus N_v\}$ such that $|S| \leq k$

$S := \emptyset$;
$U := \{v\}$;
for $i = 1, 2, \ldots, k$ **do**
  $u_i := \arg\max_{u \in V} \min_{u_j \in U} d_{uu_j}$;
  $U := U \cup \{u_i\}$;
  $S := S \cup \{\{u_i, v\}\}$;
return S;

---

and a vertex in U is selected and inserted into U. The solution S returned is made of edges that connect nodes u_i in $U \setminus \{v\}$ to v, $S = \{\{u_i, v\} \mid u_i \in U \setminus \{v\}\}$.

To analyze the algorithm, we need some further notation. Let $IS(G)$ be the size of a *maximum independent set* of graph $G = (V, E)$, that is the size of a maximum subset of nodes $V' \subseteq V$ such that no two nodes in V' are joined by an edge in E. Given a subset of nodes $U \subseteq V$, the *radius* of U is defined as $r_U = \min_{x \in V} \max_{u \in U} d_{xu}$. Given a graph G and an integer $d \geq 0$, $G^d = (V, E^d)$ is the graph with the same nodes as G and an edge (x, y) if the distance in G between x and y is at most d.

Let S^* be an optimal solution for the instance of CM-E and let OPT denote $e_v(S^*)$. The diameter of $G(S^*)$ is at most $2OPT$ and therefore $IS((G(S^*))^{2OPT}) = 1$. The next lemma implies that $IS((G(S^*))^{2OPT}) \geq IS(G^{2OPT}) - |S^*|$.

Lemma 11 ([3])**.** *Let G be a graph and let $d \geq 0$. For each $e \in V \times V \setminus E$, $IS((G(\{e\}))^d) \geq IS(G^d) - 1$.*

It follows that $IS(G^{2OPT}) \leq k + 1$. Let $u_0 = v$. We partition the set of nodes V into $k + 1$ clusters $U_0, U_1, \ldots, U_k$ as follows: for each $i = 0, 1, \ldots, k$, a node u belongs to U_i if $d_{uu_i} \leq d_{uu_j}$, for each $j = 0, 1, \ldots, k$, ties are arbitrarily broken in order to form a partition. Sets $U_0, U_1, \ldots, U_k$ are called the *clusters* induced by Algorithm 3. The next lemma implies that, for each $i = 0, 1, \ldots, k$, $r_{U_i} \leq 2OPT$.

Lemma 12 ([3])**.** *Let G be a graph, let $d \geq 0$, and let $U_0, U_1, \ldots, U_k$ be the clusters induced by Algorithm 3 on G. If $IS(G^d) \leq k + 1$, then for each $i = 0, 1, \ldots, k$, $r_{U_i} \leq d$.*

Clearly $|S| \leq k$ and the distance between each node $u \in V$ and v in $G(S)$ is at most $2OPT + 1$ to v, therefore, $e_v(S) \leq 2OPT + 1$. The approximation factor guaranteed by Algorithm 3 is then $2 + \frac{1}{OPT}$.

Theorem 13 ([32])**.** *In undirected graphs, the* CM-E *problem is approximable within a factor* $2 + \frac{1}{OPT}$*, where* OPT *is the value of an optimal solution.*

The next theorem shows that if we allow a number of added edges that is higher than k, then we can obtain a solution that is at most $1 + \epsilon$ far from the optimal solution of the case in which only k additional edges are allowed.

Theorem 14 ([13])**.** *For any* $\epsilon > 0$*, there exists a polynomial-time algorithm that adds* $O(k \log |V|)$ *edges to reduce the eccentricity of* v *to at most* $1 + \epsilon$ *times the optimum eccentricity for the case in which k additional edges are allowed.*

3.4 Page-rank

We first show that the CM-P problem does not admit a polynomial-time approximation scheme, and then we show that a variant of the greedy algorithm guarantees a constant approximation ratio. More details on these results can be found in [1, 31].

The next theorem states that there exist no polynomial-time approximation scheme for the CM-P problem, unless $P = NP$. The proof is quite technical and hence it is omitted here, see [31] for details.

Theorem 15 ([31])**.** *The* CM-P *problem does not admit an FPTAS, unless* $P = NP$*.*

Let I denote the $|V| \times |V|$ identity matrix and let us consider matrix the matrix $Z = (I - \alpha M)^{-1}$. Then, the entry z_{uv} of Z is the expected number of visits to node v for a random surfer walk starting at node u [1]. The value $g_v = \frac{p_v}{z_{vv}}$ is the *overall reachability* of node v from all the other nodes, that is the probability that node v is reached by a random surfer walk that starts at some node u, for all $u \in V$ [31]. Let us consider a variant of the CM-P problem where the function to maximize is g_v and let us denote such problem as CM-G. The next theorem implies that problem CM-G can be approximated by the greedy algorithm with an approximation factor of $1 - \frac{1}{e}$.

Theorem 16 ([31])**.** *In directed graphs, for each vertex v, function* g_v *is monotone and submodular with respect to any feasible solution for* CM-G*.*

Let S be the solution of the greedy algorithm for problem CM-G and let $OPT = \frac{p_v^{OPT}}{z_{vv}^{OPT}}$ denote the value of an optimal solution for CM-G. The previous theorem implies that

$$\frac{p_v(S)}{z_{vv}(S)} \geq \left(1 - \frac{1}{e}\right) \frac{p_v^{OPT}}{z_{vv}^{OPT}}.$$

Finally, the next theorem follows by the observation that, for any solution S', $z_{vv}(S') \leq \sum_{i=0}^{\infty} \alpha^{2i} = \frac{1}{1-\alpha^2}$ and $z_{vv}(S') \geq 1$ [31].

Theorem 17 ([31])**.** *In directed graphs, the* CM-P *problem is approximable within a factor* $\left(1 - \alpha^2\right)\left(1 - \frac{1}{e}\right)$*.*

Centrality index	Graph type	Inapproximability Upper/Lower bound	Approximation algorithms
Harmonic	Undir.	$1 - \frac{1}{15e}$	$1 - \frac{1}{e}$
	Dir.	$1 - \frac{1}{3e}$	$1 - \frac{1}{e}$
Betweenness	Undir.	$1 - \frac{1}{2e}$	OPEN
	Dir.	$1 - \frac{1}{2e}$	$1 - \frac{1}{e}$
Eccentricity	Undir.	$\frac{3}{2}$	$2 + \frac{1}{OPT}$ $1 + \epsilon$, with $O(k \log \lvert V \rvert)$ edges
	Dir.	OPEN	OPEN
Page-rank	Undir.	OPEN	OPEN
	Dir.	NO FPTAS	$\left(1 - \alpha^2\right)\left(1 - \frac{1}{e}\right)$

Table 1: Summary of results.

4 Summary of results and open problems

In this paper we summarized some recent results on the CM problem which consist in finding a limited amount of edges to be added incident to a given node v in a graph in such a way that the centrality of v is maximized. In particular, we used harmonic centrality, betweenness centrality, eccentricity and page-rank as centrality indices. The results outlined in this paper are reported in Table 1. It is worth to note that for all the problems, except for CM-E, the approximation algorithm used is basically the same greedy algorithm.

In the following we list some research directions that deserve further investigation.

- First of all, it would be worth to close open cases pointed out in Table 1 and to close the gaps between approximation and inapproximability results. Moreover, it would be interesting to study the CM problem with other centrality indices. Note that not always the greedy algorithm given in this paper exhibits a bounded approximation ratio, see the case of CM-B for undirected graphs, and hence new algorithms could be required.

- A centrality index c induces a ranking of the nodes which is the placement of a node v according to c and it is denoted as r_v^c. It has been experimentally observed that increasing the centrality of a given node v has the consequence of increasing the ranking of v [9, 10, 12]. Therefore, maximizing the centrality of v like in the CM problem decreases a lot r_v^c. However, it could be worth to directly study the problem of optimizing the position of v in the ranking, that is find a set S of edges incident to v that minimizes $r_v^c(S)$ or maximizes the possible increment in the ranking of v, $r_v^c - r_v^c(S)$.

- The notion of centrality index of a node can be extended to a set of nodes in the graph. The aim is to capture the centrality of a particular class of individuals within a large community (e.g. a specific department inside a large company). Informally, given a centrality index c and a subset of nodes $U \subseteq V$, the *group centrality index* c on U is the centrality of a "virtual" node u that collapses all the nodes in U. Relevant group centrality indices are degree, closeness, betweenness, and flow betweenness [15]. It could be interesting to extend the CM problem to group centrality indices, i.e. maximizing the group centrality of a given group of individuals in a network by adding a limited amount of edges incident to one or more nodes in the group.

- It is not difficult to figure out scenarios in which two or more nodes try to increase their centrality by adding new edges. In such a scenario the best strategy that each node should adopt might be different from the greedy one. It would then be interesting to study this scenario from a game theoretic perspective.

- In the field of complex networks, different models of information diffusion have been introduced in the literature in order to model the dynamics that regulate the diffusion of information in a network. Important examples are the *Linear Threshold Model* [21, 23, 33] and the *Independent Cascade Model* [19, 20, 23, 24]. In such models, we can distinguish between active, or infected, nodes which spread the information and inactive ones. At the beginning of the process a small percentage of nodes of the graph is set to active in order to let the information diffusion process start. Recursively, currently infected nodes can infect their neighbours with some probability. After a certain number of such cycles, a large number of nodes might become infected in the network. The *influence maximization problem* consists in finding a set A of k nodes such that if we initiate the spreading of information by activating the nodes in A, the number of nodes that is active at the end of the process is maximum. Nodes in A are called seeds. A possible extension of the work in this paper is to determine a limited number of edges to be added incident to the seeds in order to maximize the eventual number of active nodes. Preliminary results for the case of the independent cascade model have been presented in [11].

- Assuming that $k = o(|E|)$, the time complexity of the greedy algorithm is $O(k|V|f_c(|V|, |E|))$, where $f_c(|V|, |E|)$ is the time complexity of computing c for a node v in graph $G = (V, E \cup S)$ for some solution S such that $|S| \leq k$. In many cases, $f_c(|V|, |E|)$ is at least linear in $|E|$ and therefore, the time complexity of the greedy algorithm is at least quadratic. When graphs are huge, with billions of nodes and edges, the greedy algorithm requires to much time. However, we do not actually need to recompute the centrality of a node for each possible added edge and at each iteration but we can exploit the so-called *dynamic algorithms* for centrality measures that compute $c_v(S \cup \{e\})$ starting from $c_v(S)$ in a smaller computational time compared to $f_v(|V|, |E \cup S \cup \{e\}|)$. This approach has been adopted for the case of harmonic centrality [10].

- In many applications one wants to minimize the centrality of a given node rather then maximize it. Examples are applications in which one wants to reduce the traffic flow in nodes of a road or a communication networks or reduce the spread of disease in epidemic and social networks. In general this can be done by deleting edges from the graph. Therefore it would be interesting to study the "dual" problem of CM in which we want to minimize the centrality of a given node by deleting edges incident to that node.

References

[1] K. Avrachenkov and N. Litvak. The effect of new links on google pagerank. *Stochastic Models*, 22(2):319–331, 2006.

[2] A. Bavelas. Communication patterns in task-oriented groups. *The Journal of the Acoustical Society of America*, 22(6):725–730, 1950.

[3] D. Bilò, L. Gualà, and G. Proietti. Improved approximability and non-approximability results for graph diameter decreasing problems. *Theoretical Computer Science*, 417:12–22, 2012.

[4] P. Boldi and S. Vigna. Axioms for centrality. *Internet Mathematics*, 10(3–4):222–262, 2014.

[5] S. P. Borgatti. Centrality and network flow. *Social networks*, 27(1):55–71, 2005.

[6] U. Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(2):163–177, 2001.

[7] U. Brandes. On variants of shortest-path betweenness centrality and their generic computation. *Social Networks*, 30(2):136–145, 2008.

[8] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998.

[9] P. Crescenzi, G. D'Angelo, L. Severini, and Y. Velaj. Greedily improving our own centrality in a network. In *Proceedings of the 14th International Symposium on Experimental Algorithms (SEA 2015)*, volume 9125 of *Lecture Notes in Computer Science*, pages 43–55. Springer, 2015.

[10] P. Crescenzi, G. D'Angelo, L. Severini, and Y. Velaj. Greedily improving our own closeness centrality in a network. *ACM Transactions on Knowledge Discovery from Data*, 11(1):9:1–9:32, 2016.

[11] G. D'Angelo, L. Severini, and Y. Velaj. Influence maximization in the independent cascade model. In *Proceedings of the 17th Italian Conference on Theoretical Computer Science (ICTCS16)*, CEUR Workshop Proceedings. CEUR-WS.org, 2016. To appear.

[12] G. D'Angelo, L. Severini, and Y. Velaj. On the maximum betweenness improvement problem. *Electronic Notes in Theoretical Computer Science*, 322:153 – 168, 2016. Proceedings of the 16th Italian Conference on Theoretical Computer Science (ICTCS15).

[13] E. D. Demaine and M. Zadimoghaddam. Minimizing the diameter of a network using shortcut edges. In *Proceedings of the 12th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2010)*, volume 6139 of *Lecture Notes in Computer Science*, pages 420–431. Springer, 2010.

[14] I. Dinur and D. Steurer. Analytical approach to parallel repetition. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC 2014)*, pages 624–633. ACM, 2014.

[15] M. G. Everett and S. P. Borgatti. The centrality of groups and classes. *The Journal of mathematical sociology*, 23(3):181–201, 1999.

[16] U. Feige. A threshold of ln n for approximating set cover. *Journal of the ACM*, 45(4), 1998.

[17] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977.

[18] M. R. Garey and D. S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York, 1979.

[19] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing letters*, 12(3):211–223, 2001.

[20] J. Goldenberg, B. Libai, and E. Muller. Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata. *Academy of Marketing Science Review*, 2001(9):1, 2001.

[21] M. Granovetter. Threshold models of collective behavior. *American journal of sociology*, 83(6):1420–1443, 1978.

[22] L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

[23] D. Kempe, J. Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. *Theory of Computing*, 11(4):105–147, 2015.

[24] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2003)*, pages 137–146. ACM, 2003.

[25] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[26] N. Lin. *Foundations of social research*. McGraw-Hill, New York, 1976.

[27] B. Macdonald, P. Shakarian, N. Howard, and G. Moores. Spreaders in the network SIR model: An empirical study. *CoRR*, abs/1208.4269, 2012.

[28] P. Malighetti, G. Martini, S. Paleari, and R. Redondi. The impacts of airport centrality in the EU network and inter-airport competition on airport efficiency. Technical Report MPRA-7673, 2009.

[29] G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of approximations for maximizing submodular set functions–I. *Mathematical Programming*, 14(1):265–294, 1978.

[30] M. Newman. *Networks: an introduction*. Oxford university press, 2010.

[31] M. Olsen and A. Viglas. On the approximability of the link building problem. *Theoretical Computer Science*, 518:96–116, 2014.

[32] S. Perumal, P. Basu, and Z. Guan. Minimizing eccentricity in composite networks via constrained edge additions. In *Proceedings of the 32th IEEE Military Communications Conference (MILCOM)*, pages 1894–1899. IEEE, 2013.

[33] T. C. Schelling. *Micromotives and macrobehavior*. Norton & Company, 2006.

[34] A. Shimbel. Structural parameters of communication networks. *The bulletin of mathematical biophysics*, 15(4):501–507, 1953.

[35] D. Williamson and D. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.

[36] E. Yan and Y. Ding. Applying centrality measures to impact analysis: A coauthorship network analysis. *Journal of the Association for Information Science and Technology*, 60(10):2107–2118, 2009.

WHICH TASKS OF A JOB ARE SUSCEPTIBLE TO COMPUTERIZATION?

Philipp Brandes and Roger Wattenhofer

ETH Zurich, Switzerland

Abstract

In their paper, the two economists Frey and Osborne quantified the automation of jobs, by assigning each job in the O*NET database a probability to be automated. In this paper, we refine their results in the following way: Every O*NET job consists of a set of tasks, and these tasks can be related. We use a linear program to assign probabilities to tasks, such that related tasks have a similar probability and the tasks can explain the computerization probability of a job. Analyzing jobs on the level of tasks helps comprehending the results, as experts as well as laymen can more easily criticize and refine what parts of a job are susceptible to computerization.

1 Introduction

Computerization is considered to be one of the biggest socio-economic challenges. What is the foundation of the recent worries about many jobs being affected by automation [7, 8, 11, 12]?

Why did the last few years see dramatic technological progress regarding self-driving cars [14], board games [20], automatic language translation [1], or face recognition [16]? One reason is big data. While "intelligent algorithms" in the past were restricted to learning from data sets with a few thousand examples, we now have exabytes of data. Learning becomes even more powerful if you combine big data with a highly parallel hardware, stirred by the success of graphics processing units (GPUs). However, both of these technological advancements needed to be harvested, and they are with the advent of so-called deep learning algorithms, which have blown the competition away, starting with voice recognition [10]. As a consumer, you can already witness some of these advancements on your smartphone, but a lot more will come soon. We believe that these advancements will revolutionize white collar work and (with a little help from sensors and robotics) also blue collar work. In contrast to previous waves of innovation, this time new emerging jobs might not be able to compensate jobs endangered by the new technology.

Computer scientists have been pondering the consequences of AI for a long time; starting with the possibly most famous paper in computer science "Computing Machinery and Intelligence" by Alan Turing, where he proposed the Turing test [21]. An early pioneer of AI, Marvin Minsky, claimed already 1967 that "within a generation ... the problem of creating 'artificial intelligence' will substantially be solved" [19]. More recent work in computer science focuses on the effects on the economy and the society [22–24].

In their paper, Frey and Osborne [12], two economists, quantitatively study job automation, predicting that 47% of US employment is at risk of automation. In order to calculate this number, Frey and Osborne labeled 70 of the 702 jobs from the O*NET OnLine job database[1] manually as either "automatable" or "not automatable". Then, for the remainder of the jobs in the O*NET database, they computed the automation probability as a function of the distance to the labeled jobs.

But the results of Frey and Osborne are opaque, one either believes their "magic" computerization percentages, or one has doubts. We want anybody to be able to easily understand and argue about our results, by incorporating the unique tasks of each job. This additional depth will help laymen as well as job experts to argue about potential flaws in our methodology.

If we know that a job is 100% automatable, we also know that every task of that job must be completely automatable. But what if a job is 87% automatable? Is every task 87% automatable? Or are 87% of the tasks completely automatable, and 13% not at all? We want to forecast which tasks of a job are safe and which tasks are automatable.

In order to calculate the automation probability for a task, we first need to determine its share of a job (Section 4). Based on this, we are able to assign each task a probability to be automated such that the weighted average of the probabilities is equal to the probability of the corresponding job (Section 6.1). During our evaluation (Section 6.2), we discover a few suspicious results in the probabilities by Frey and Osborne, e.g., a surprisingly high automation probability of 96% for the job *compensation and benefits managers*.

We analyze the correlation between various properties of a job and its probability to be automated (Section 7). E.g., we show that there is a strong negative correlation between the level of education required for a job and its probability to

[1] O*NET is an application that was created for the general public to provide broad access to the O*NET database of occupational information. The site is maintained by the National Center for O*NET Development, on behalf of the U.S. Department of Labor, Employment and Training Administration (USDOL/ETA); see https://www.onetonline.org/

be automated. In Section 8, we try to determine whether the effects of automation can already be seen in our data.

Our complete results can be found at `http://jobs-study.ethz.ch`.

2 Related Work

The current effects of automation have been studied intensively in economics. Most studies agree that some routine tasks have already fallen victim to automation [2, 4, 13]. A task is routine if "it can be accomplished by machines following explicit programmed rules" [4]. With computers being able to do routine tasks, the demand for human labor performing these tasks has decreased. But on the other hand, the demand for college educated labor has increased over the last decades [6, 25, 26]. The effect is more pronounced in industries that are computer-intensive [3]. As a consequence of this, the employment share of the highest skill quartile has increased. In addition to more people being employed in the highest skill quartile, the real wage for this quartile has increased faster than the average real wage. Service occupations, which are non-routine, but also not well paid, have also seen an increase in employment share and in real hourly wage. Thus, both, employment share and real wage, are U-shaped with respect to the skill level [2]. This employment pattern is a phenomenon that is called polarization. This is not unique to the US, but can also be observed, e.g., in the UK [13]. These papers make important observations about the effects that automation already has. Until now, routine tasks are the ones most affected, but more and more tasks can nowadays be performed by a computer. We focus on the future and try to predict which tasks will be automated next.

John Keynes predicted already in 1933 that there will be widespread technological unemployment "due to the means of economising the use of labour outrunning the pace at which we can find new uses for labour" [15]. Automation might be the technology, where this becomes true [7, 8, 11, 12]. "Automation of knowledge work", "Advanced robotics", and "Autonomous and near-autonomous vehicles" are considered to be 3 out of 12 potentially economically disruptive technologies [17]. Computer labor and human labor may no longer be complements, but competitors. Automation might be the cause for the current stagnation [7]. There might be too much technological progress, which causes high unemployment. A trend that could be going on for years, but was hidden by the housing boom [9].

The paper by Frey and Osborne is the first to make quantitative claims about the future of jobs [12]. Together with 70 machine learning experts, Frey and Osborne first manually labeled 70 out of 702 jobs from the O*NET database as either "automatable" or "non automatable". This labeling was, as the authors admit, a subjective assignment based on "eye balling" the job descriptions from

O*NET. Labels were only assigned to jobs where the whole job was considered to be (non) automatable, and to jobs where the participants of the workshop were most confident. To calculate the probability for non-labeled jobs, Frey and Osborne used a probabilistic classification algorithm. They chose 9 properties from O*NET as features for their classifier, namely "Finger Dexterity", "Manual Dexterity", "Cramped Work Space, Awkward Positions", "Originality", "Fine Arts", "Social Perceptiveness", "Negotiation", "Persuasion", and "Assisting and Caring for Others".

The results from Frey and Osborne for the US job market were adopted to other countries, e.g., Finland, Norway, and Germany [5, 18]. This was done by matching each job from O*NET to the locally used standardized name. Due to differences in the economies, a different percentage of people will be affected by this change, e.g., only one third in Finland and Norway are at risk compared to 47% in the US.

3 Model

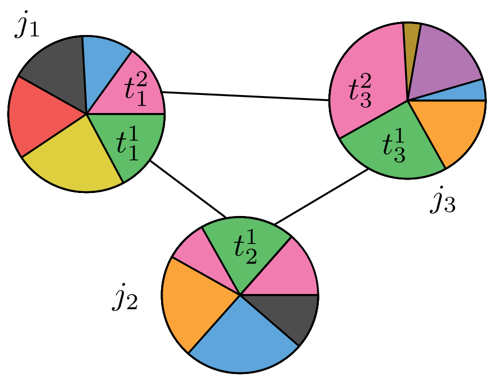
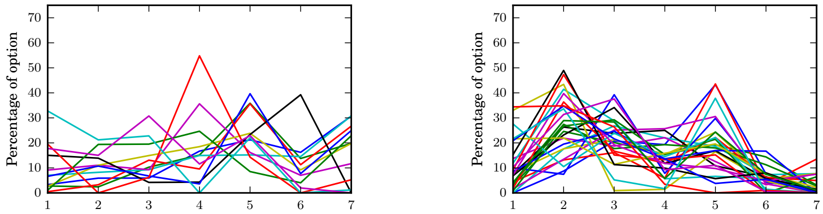


Figure 1: This figure shows a small example consisting of three jobs represented by circles. The share of each task of a job is shown by its sector. Two tasks are connected by a line if and only if they are related, i.e., similar according to O*NET. Task t_1^1 from job j_1 and task t_2^1 from job j_2 are related as indicated by the line connecting them. Note that this relationship is not transitive. Thus, tasks t_1^1 and t_3^1 do not need to be related.

We are given a set of jobs $J = \{j_1, \ldots, j_n\}$. Each job j_i consists of a set of tasks $T_i = \{t_i^1, \ldots, t_i^m\}$, where every task belongs to exactly one job. We call two tasks t_i^k and $t_{i'}^{k'}$ related if and only if these tasks are similar according to O*NET. Two jobs



(a) Electro-Mechanical Technicians (b) Environmental Engineering Technicians

Figure 2: The number of tasks and the frequencies assigned to them can differ significantly even for related jobs.

with related tasks are also called related. An example with 3 jobs is depicted in Figure 1.

O*NET provides us for each task t_i^k with the information how often it is performed. This information was gathered by asking job incumbents and occupational experts. The options are "yearly or less", "more than yearly", "more than monthly", "more than weekly", "daily", "several times daily", and "hourly or more". O*NET provides a percentage for each of the 7 options. We denote these frequencies of task t_i^k with $f_1(t_i^k), \ldots, f_7(t_i^k)$. Since these values are percentages, for every task t_i^k they sum up to 100%, i.e., $\sum_{\ell=1}^{7} f_\ell(t_i^k) = 1$.

Each job j_i has a given probability $p(j_i)$ to be automated. We want to use $p(j_i)$ to calculate a probability to be automated for each task of this job.

4 From Task Frequencies to Task Shares

We use the frequencies with which a task is performed to assign each task t_i^k its share $s(t_i^k)$. For every task t_i^k, the share denotes how much time is spent doing this task, such that $\sum_{t_i^k \in T_i} s(t_i^k) = 1$. The frequency values from O*NET do not fulfill this property; their values are very consistent for one job, but they can vary a lot between different jobs and might even seem to contradict each other. An extreme example can be seen in Figure 2. The seven frequency options provided by O*NET are on the x-axis and on the y-axis is the corresponding value of each option.

To make use of the high consistency within a job, we decided that the share of a task is a weighted average of its frequencies, i.e., $s(t_i^k) := \sum_{\ell=1}^{7} x_i^\ell f_\ell(t_i^k)$. We want to calculate the job specific coefficients x_i^ℓ. Let us illustrate these coefficients with a simple example. If $x_i^7 = 0.1$, then a task t_i^k that is done exclusively "hourly or more" (i.e., $f_7(t_i^k) = 1$) makes up 10% of job j_i.

We want these coefficients to satisfy a few assumptions. If O*NET states that a task is done "hourly or more", then the share of this task should be higher than the share of a task that is done "several times daily". This translates to $x_i^\ell \leq x_i^{\ell+1}$ $\forall \ell \in \{1, \ldots, 6\}$ and $0 \leq x_i^1$.

These constraints neither use that jobs are related nor do they define the coefficients uniquely. Both issues are solved if we require the coefficients x_i^ℓ and $x_{i'}^\ell$ for two related jobs j_i and $j_{i'}$ to be similar. The intuition behind this is that the frequencies of O*NET for related jobs are not independent of each other either, but rather should be similar as well. Occupational experts who have rated the frequency in which a task is done for one job, are likely to have rated the frequencies of related jobs.

The coefficients cannot be identical without violating the other constraints. Jobs have a different number of tasks and the frequencies are task specific. The example in Figure 2 highlights this. It is therefore easy to see that we cannot have the same coefficients for two related jobs and fulfill the equality $\sum_{t_i^k \in T_i} s(t_i^k) = 1$ for both jobs simply because the number of tasks can differ a lot.

Thus, we allow a bit of slack in the coefficients of related jobs. We use the variable $x_{i,i'}^\ell$ to express the difference between the coefficients x_i^ℓ and $x_{i'}^\ell$ for two related jobs $j_i, j_{i'}$. Formally, we define it as $x_{i,i'}^\ell := \max\{x_i^\ell - x_{i'}^\ell, x_{i'}^\ell - x_i^\ell\}$. This yields the following linear program, which minimizes the overall slack:

$$\begin{array}{rcll}
\text{minimize} & \sum x_{i,i'}^\ell & & \\
\text{s.t.} & & & \\
x_{i,i'}^\ell \geq & x_i^\ell - x_{i'}^\ell & \forall \ell & \\
 & & \forall j_i, j_{i'} \in J \text{ that are related} & \\
x_{i,i'}^\ell \geq & x_{i'}^\ell - x_i^\ell & \forall \ell & \\
 & & \forall j_i, j_{i'} \in J \text{ that are related} & \\
\displaystyle\sum_{t_i^k \in T_i} \sum_{\ell=1}^{7} x_i^\ell f_\ell(t_i^k) \leq & 1 + \varepsilon & \forall j_i \in J & \\
\displaystyle\sum_{t_i^k \in T_i} \sum_{\ell=1}^{7} x_i^\ell f_\ell(t_i^k) \geq & 1 - \varepsilon & \forall j_i \in J & \\
x_i^1 \geq & 0 & \forall j_i \in J & \\
x_i^\ell \geq & x_i^{\ell-1} & \forall j_i \in J & \ell \in \{2, \ldots, 7\}
\end{array}$$

We set ε to 0.01. The resulting LP has 169,372 variables in its objective function. Since there are 735 jobs,[2] this means that a job is related to approximately 32.9

[2]We consider slightly more jobs than Frey and Osborne, since we use the finest granularity available from O*NET.

other jobs on average. The value of the objective function is 24.6, i.e., for two related jobs the coefficients differ only by 0.000145 on average. For comparison, the average value of a coefficient is 0.060. Our complete results can be found online at `http://jobs-study.ethz.ch`.

5 From Jobs to Tasks

Knowing the shares of the tasks enables us to set up a linear program that calculates for each task the probability to be automated. We want that the weighted average of the automation probabilities $p(t_i^k)$ of the tasks of a job j_i can explain the automation probability $p(j_i)$ of the job, i.e., $\sum_{t_i^k \in T_i} p\left(t_i^k\right) \cdot s\left(t_i^k\right) \approx p\left(j_i\right)$. Furthermore, we want to assign related tasks similar automation probabilities. To do this, we define a variable $t_{i,i'}^{k,k'}$ for each pair of related tasks t_i^k and $t_{i'}^{k'}$. It denotes the probability difference that we assign to the two tasks. Formally, it is defined as $t_{i,i'}^{k,k'} := \max\{p(t_i^k) - p(t_{i'}^{k'}), p(t_i^k) - p(t_{i'}^{k'})\}$. We want to minimize the sum of these variables, i.e., the sum of the probability difference of all related tasks.

Combining these requirements with necessary conditions to have meaningful probabilities, i.e., $0 \leq p(t_i^k) \leq 1$, yields the following linear program:

$$
\begin{array}{rcl}
\text{minimize} & \sum t_{i,i'}^{k,k'} & \\
\text{s.t.} & & \\
p(t_i^k) - p(t_{i'}^{k'}) \leq & t_{i,i'}^{k,k'} & \forall t_i^k, t_{i'}^{k'} \text{ that are related} \\
p(t_{i'}^{k'}) - p(t_i^k) \leq & t_{i,i'}^{k,k'} & \forall t_i^k, t_{i'}^{k'} \text{ that are related} \\
\sum_k p(t_i^k) \cdot s\left(t_i^k\right) \leq & p(j_i)\,(1+\varepsilon) & \forall j_i \in J \\
\sum_k p(t_i^k) \cdot s\left(t_i^k\right) \geq & p(j_i)\,(1-\varepsilon) & \forall j_i \in J \\
p(t_i^k) \geq & 0 & \forall j_i \in J, t_i^k \in T_i \\
p(t_i^k) \leq & 1 & \forall j_i \in J, t_i^k \in T_i \\
t_{i,i'}^{k,k'} \geq & 0 & \forall t_{i,i'}^{k,k'} \\
t_{i,i'}^{k,k'} \leq & 1 & \forall t_{i,i'}^{k,k'}
\end{array}
$$

We set ε to 0.01.

6 Linear Program Results

We now analyze the results of the linear program as described above. Later on, we will look at a small refinement to automatically detect outliers in our results.

6.1 Task Probabilities

The linear program as described in Section 5 has 105,748 variables in its objective function and it has a minimal value of 9,846. This means that two related tasks differ, on average, with regard to their probability by 9.3%. The complete results can be found online at `http://jobs-study.ethz.ch`. The histogram of the probability difference between two related tasks is shown in Figure 3. A majority of related tasks is assigned a similar probability. A small fraction of related tasks is assigned diametrically opposed probabilities, which seems startling. It can be reconciled by considering that neither the classification by Frey and Osborne nor the classification of tasks being related by O*NET are perfect.

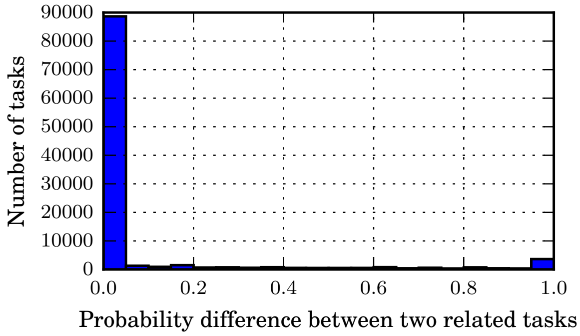


Figure 3: A histogram of the probability with which two related tasks differ.

One example that highlights this are the two jobs *computer programmer* and *software developers, applications*. These two jobs have many related tasks, but the probabilities of these jobs differ a lot (4% for *software developers, applications*, 48% for *computer programmers*). Hence, the diametrically opposed probabilities are necessary to meet the constraints of the linear program.

In the following, we present a few selected jobs to illustrate our results. The first example is *chemists*. This job has an automation probability of 10% according to Frey and Osborne. Only one task has, according to our linear program, a high probability of being automated: "Induce changes in composition of substances by introducing heat, light, energy, or chemical catalysts for quantitative or qualitative

Task Description	p	Share
Write decisions on cases.	1	5.1
Instruct juries on applicable laws, direct juries to deduce the facts from the evidence presented, and hear their verdicts.	1	3.4
Monitor proceedings to ensure that all applicable rules and procedures are followed.	1	8.0
Advise attorneys, juries, litigants, and court personnel regarding conduct, issues, and proceedings.	1	6.2
Interpret and enforce rules of procedure or establish new rules in situations where there are no procedures already established by law.	1	5.4
Conduct preliminary hearings to decide issues such as whether there is reasonable and probable cause to hold defendants in felony cases.	1	3.9
Rule on admissibility of evidence and methods of conducting testimony.	0.94	5.3
Preside over hearings and listen to allegations made by plaintiffs to determine whether the evidence supports the charges.	0.46	5.9
Perform wedding ceremonies.	0.39	2.7
Read documents on pleadings and motions to ascertain facts and issues.	0	10.1
Research legal issues and write opinions on the issues.	0	6.5
Settle disputes between opposing attorneys.	0	4.6
Participate in judicial tribunals to help resolve disputes.	0	6.6
Rule on custody and access disputes, and enforce court orders regarding custody and support of children.	0	6.3
Sentence defendants in criminal cases, on conviction by jury, according to applicable government statutes.	0	4.0
Grant divorces and divide assets between spouses.	0	4.7
Award compensation for damages to litigants in civil cases in relation to findings by juries or by the court.	0	3.8
Supervise other judges, court officers, and the court's administrative staff.	0	8.5

Table 4: The automation probability and the share of each task of *Judges, Magistrate Judges, and Magistrates"*. The automation probability of this job is 40%.

analysis." Other simple mechanical tasks have been assigned low automation probabilities. We will revisit this job in Section 6.2.

Next up: *judges*. Their automation probability is 40%. The tasks, their probabilities, and their shares are shown in Table 4. The tasks that can be automated can be grouped in two sets: preliminary hearings which includes making first assessments, and ensuring that the procedures in court are followed. The tasks that involve sentencing (or the preparation thereof) have been assigned low automation probabilities.

Figure 5 shows the histogram of the probabilities from our linear program. The probabilities for most tasks are either very high or very low and only a few tasks have a probability in-between. This desired side effect of our linear program helps us to achieve our goal of allowing job experts (and laymen) to argue about the validity of our results. We invite the reader to have a look at other jobs at `http://jobs-study.ethz.ch`.

6.2 Outlier Detection

To evaluate our approach and check for outliers, we use a variant of cross-validation. For every job j_i, we create a linear program without job j_i. This yields a probability

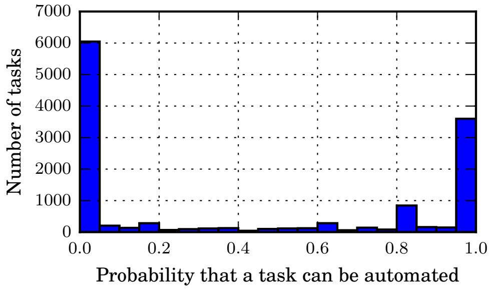


Figure 5: A histogram of the probabilities that tasks can be automated. Nearly all tasks are assigned either 0 or 1. This simplifies arguing whether our classification is correct.

$p_i(\cdot)$ for every task but the tasks from j_i. Afterward, we calculate the new probability $p'(j_i)$ that job j_i can be automated. We do this by setting the probability $p'(t_i^k)$ of each task t_i^k to the average of all tasks that are related to it. We denote the set of related tasks by $N(t_i^k)$. Formally, we set $p'(t_i^k) := \frac{1}{|N(t_i^k)|} \sum_{t_{i'}^{k'} \in N(t_i^k)} p_i(t_{i'}^{k'})$.

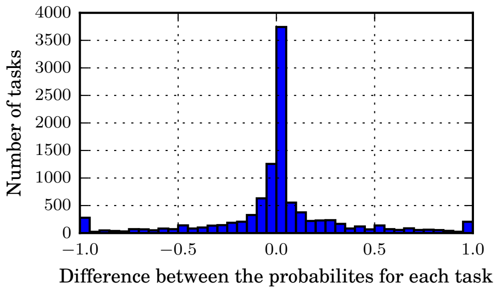


Figure 6: A histogram of $p(t_i^k) - p'(t_i^k)$ for all tasks. This plot shows that most tasks have a similar probability in both approaches. The values are sharply distributed around 0.

We first compare $p'(t_i^k)$ with $p(t_i^k)$. The difference between these two probabilities should be small for the majority of the tasks. This is indeed what can be seen in Figure 6. The histogram of $p'(t_i^k) - p(t_i^k)$ shows that nearly all tasks have similar probabilities in both approaches. The average absolute difference is less than 20%. The distribution is centered around 0. Its mean is less than 0.05%.

By combining the new probability of each task with its share, we can calculate the new probability of job j_i by using a weighted average. This allows us to compare $p'(j_i)$ with $p(j_i)$.

We have plotted this difference, i.e., $p(j_i) - p'(j_i)$, in Figure 7. We can see that the difference is centered around 0%; with the average absolute difference being less than 20%. For more than half of the jobs our probability differs by less than 20% from Frey and Osborne [12]. Most interesting are the jobs whose probability differs significantly. We now have a look at a few of them.

There are jobs where our probability is more than 80% smaller than the one by Frey and Osborne. One job is *compensation and benefits managers*. We assigned it a probability $p'(j)$ to be automated of 9.1%; compared to $p(j) = 96\%$ by Frey and Osborne. We do not claim to know the true value, but we can look at the job and compare it to the probabilities of jobs we consider similar. Notice that this is conceptually similar to what our linear program does and thus might be biased. The tasks of this job are shown in Table 8. If we manually compare them to related tasks, we conclude that they do not seem to be automatable in the next few decades. We do favor our result over the result of Frey and Osborne.

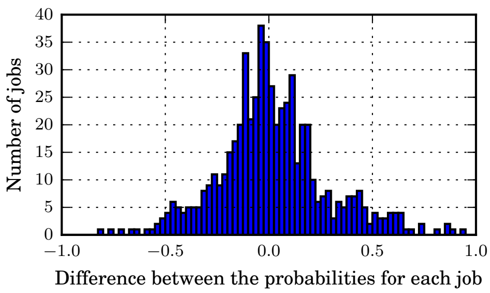


Figure 7: A histogram of the difference between the probability by Frey and Osborne with our probability. The distribution is centered around 0 and a majority of the jobs differs by less than 20%.

There is only one job that we assign a much higher probability than Frey and Osborne. The job *First-Line supervisors of production and operating workers* has been assigned a 83% automation probability by us and only 1.6% by Frey and Osborne. A close inspection of the tasks makes us believe that the true value is between these extremes. Quite a few of the tasks are clearly automatable, e.g., "Keep records of employees' attendance and hours worked." and "Observe work and monitor gauges, dials, and other indicators to ensure that operators conform to production or processing standards." Others, e.g., "Read and analyze charts, work orders, production schedules, and other records and reports to determine

Task Description	p	p'
Advise management on such matters as equal employment opportunity, sexual harassment and discrimination.	1	0.15
Study legislation, arbitration decisions, and collective bargaining contracts to assess industry trends.	1	0
Fulfill all reporting requirements of all relevant government rules and regulations, including the Employee Retirement Income Security Act (ERISA).	1	0.20
Investigate and report on industrial accidents for insurance carriers.	1	0.12
Represent organization at personnel-related hearings and investigations.	1	0
Analyze compensation policies, government regulations, and prevailing wage rates to develop competitive compensation plan.	1	0.5
Mediate between benefits providers and employees, such as by assisting in handling employees' benefits-related questions or taking suggestions.	1	0.42
Prepare detailed job descriptions and classification systems and define job levels and families, in partnership with other managers.	1	0
Prepare personnel forecasts to project employment needs.	1	0
Direct preparation and distribution of written and verbal information to inform employees of benefits, compensation, and personnel policies.	1	0
Manage the design and development of tools to assist employees in benefits selection, and to guide managers through compensation decisions.	1	0
Design, evaluate and modify benefits policies to ensure that programs are current, competitive and in compliance with legal requirements.	1	0
Administer, direct, and review employee benefit programs, including the integration of benefit programs following mergers and acquisitions.	1	0
Prepare budgets for personnel operations.	1	0.03
Maintain records and compile statistical reports concerning personnel-related data such as hires, transfers, performance appraisals, and absenteeism rates.	1	0
Contract with vendors to provide employee services, such as food services, transportation, or relocation service.	1	0.38
Identify and implement benefits to increase the quality of life for employees, by working with brokers and researching benefits issues.	1	0
Plan, direct, supervise, and coordinate work activities of subordinates and staff relating to employment, compensation, labor relations, and employee relations.	1	0
Negotiate bargaining agreements.	1	0.67
Plan and conduct new employee orientations to foster positive attitude toward organizational objectives.	1	0
Conduct exit interviews to identify reasons for employee termination.	1	0
Develop methods to improve employment policies, processes, and practices, and recommend changes to management.	0.51	0
Formulate policies, procedures and programs for recruitment, testing, placement, classification, orientation, benefits and compensation, and labor and industrial relations.	0.23	0.01

Table 8: The tasks and their corresponding probability that they will be automated for *Compensation and Benefits Managers* according to our original linear program and the cross-validation.

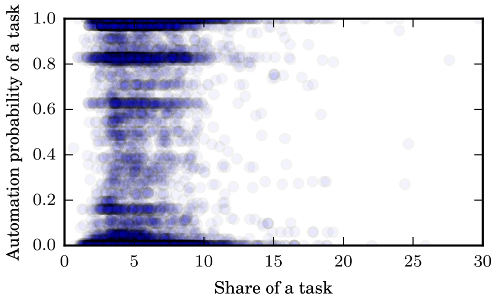


Figure 9: Our results show almost no correlation between the share of a task and its probability that it will be automated.

production requirements and to evaluate current production estimates and outputs." seem difficult to automate. The complete results for this can job be found at `http://jobs-study.ethz.ch`.

We continue by comparing the previous results with the approach described in this section. To do this, we return to the jobs that we have looked at previously. First off is the job *chemists*. The automation probability of most tasks has increased. Consequently, the automation probability of this job has increased from 10% to 42%. Due to the large difference, this job should be analyzed in-depth by job experts.

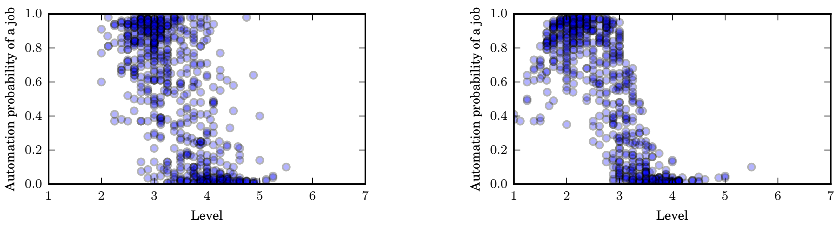
The changes in the automation probability of the tasks of *judges* are much smaller. Most tasks have a similar automation probability as before and the overall probability of this job has changed marginally, i.e., increased only from 40% to 50%. Therefore, we are confident that the classification by Frey and Osborne is correct.

We conclude that our approach can also be used to detect outliers in the results of Frey and Osborne. We can then manually inspect the automation probabilities of the tasks of such an outlier to determine the truth. We think our results allow us to fine tune the results from Frey and Osborne, but not replace it, as we need their results to bootstrap our linear program.

7 Further Analysis

In addition to inspecting every task of every job, we consider a broader picture. We do this by looking at general properties of a job that correlate with the probability that it can be automated.

7.1 Tasks



(a) Deductive Reasoning

(b) Originality

Figure 10: The probability that a job can be automated over the level of the abilities "deductive reasoning" and "originality" used in this job. These levels are defined by O*NET and for each they provide an anchor point. Level 2 of "deductive reasoning" means "knowing that a stalled car can coast downhill" and level 5 "deciding what factors to consider in selecting stocks". Level 2 of "originality" means "using a credit card to open a locked door" and level 6 "inventing a new type of man-made fiber". Every point represents one job. A higher level of either of these two abilities correlates, as expected, with a lower probability to be automated.

We first analyze the share of a task. The higher the share, the more often a task is performed. Hence, from a machine learning perspective this means that much more training data is available. This might lead to the conclusion that such a task is easier to automate. To disprove this claim, we plotted the share of a task over the probability that a task can be automated according to our linear program. The resulting graph is shown in Figure 9. Every dot represents one task, with its share on the x-axis and its probability on the y-axis. We see that there is barely any correlation between these two. We conclude that tasks that are done more frequently are not more likely to be automated.

7.2 Jobs

We continue our analysis by looking at the correlation between the properties that a job has, e.g., what kind of degree is necessary to do a job, and the probability that this job can be automated. Correlation does not imply causation, but nevertheless, these results reveal some interesting nuggets.

O*NET provides the level that the ability "deductive reasoning" is used in a job. The level ranges from 1 to 7, where for example level 2 means "knowing that a stalled car can coast downhill" and level 5 "deciding what factors to consider

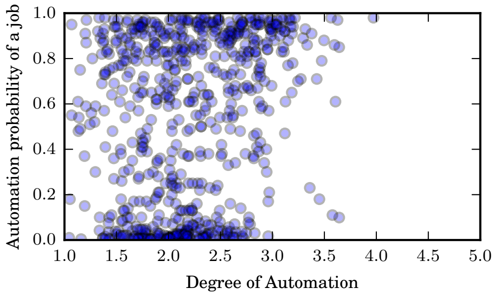
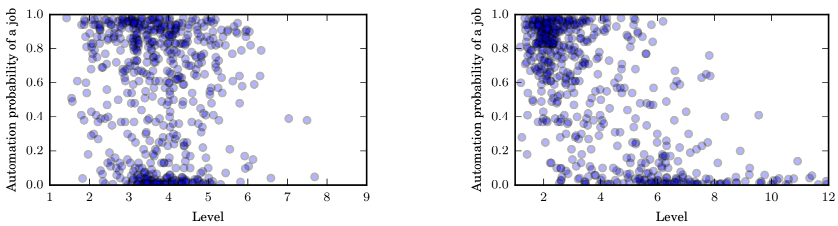


Figure 11: The probability that a job can be automated over the level of the current "Degree of Automation". This level ranges from 1 (Not at all automated) to 5 (Completely automated). The rather small correlation of 0.23 implies that different jobs will soon be affected.

in selecting stocks". For every job, we have one value between 1 and 7. The resulting graph can be seen in Figure 10. Every job is represented by one dot; its x-coordinate being its level and the y-coordinate its probability. We can see that jobs that require a high level of deductive reasoning tend to have a lower probability of being automated. A similar result can be seen for "originality" (see Figure 10).



(a) On the Job Training

(b) Required Level of Education

Figure 12: The probability that a job can be automated over the amount of "On the Job Training", which ranges from 1 (none or short demonstration) to 9 (over 10 years) and the "Required of Level of Education", which ranges from 1 (less than a high school diploma) to 12 (post-doctoral training).

Level 2 of "originality" means "using a credit card to open a locked door" and level 6 means "inventing a new type of man-made fiber". This confirms our expectation that these abilities will remain difficult for a computer.

O*NET even has an explicit value for the current level of the "degree of automation" for each job. This level ranges from 1 (not at all automated) to 5 (completely automated). As depicted in Figure 11, the already existing level of automation barely correlates with the probability that this job will be automated. This indicates that not only jobs that are already affected by automation are in danger, but also a whole new set of jobs. This is aligned with the recent worries about many new jobs soon being affected by computerization.

We conclude this section by looking at the effect that the level of required education for a job has on the probability to be automated. Jobs that require only very little education (level 1, i.e., less than a high school diploma) tend to have a higher probability than jobs that require an associate degree (level 5) which in turn have a higher probability than jobs that require post-doctoral training (level 12). Most jobs that require little education are in danger. It is noteworthy that the effect of training before the job is much stronger than the effect of on the job training. Jobs that require more on the job training only have a marginally smaller probability to be automated. Both plots are shown in Figure 12.

8 Existing Trends

In this section we analyze to what extent automation is already happening. To investigate this, we used historical job data from 2001 until 2015. For every year we know for every job in the O*NET database how many people were employed. Instead of using the absolute values (which tend to increase for most jobs since more and more people live and work in the US), we consider the relative values and look at the percentage of how many people are employed in this job. We consider the fraction a job j_i has in 2001 and focus on the factor c_i by which this fraction has changed in 2015. Note that due to the financial crisis, where a lot of jobs have been lost, the data shows to some extend how recession-proof a job is.

We plot the factor c_i against the automation probability of this job for every job j_i. Every job is represented by a dot; its x-coordinate being the factor c_i and the y-coordinate being its automation probability. We expect that jobs with a high automation probability tend to have a factor smaller than one, i.e., its fraction of the labor market has decreased and vice versa. The result can be seen in Figure 13. Even though there are quite a few outliers, the overall trend follows our expectation. The most extreme outlier is the job *service unit operators, oil, gas, and mining* whose fraction has increased by more than a factor of 4, but is deemed 93% automatable. This can easily be explained due to the rise of fracking. A job that is considered non-automatable, but whose fraction has been reduced to about a third of its initial value, is *advertising and promotions managers*. We suspect that this job is highly susceptible to recessions, i.e., the financial crisis in 2008/2009.

The above analysis shows that the demand for jobs that are highly automatable

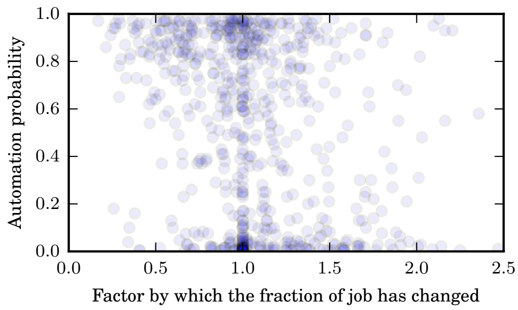
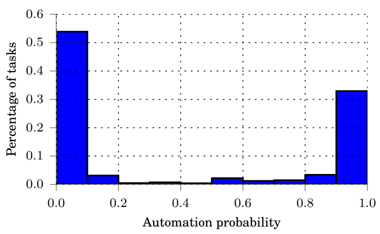


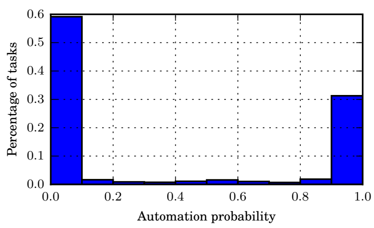
Figure 13: The factor c_i by which the fraction of a job has changed over the automation probability. We can see that jobs that have a factor smaller than one tend to have a high automation probability.

has already started to decrease. We once again look at the problem from a task level perspective. We assume that the demand for a job decreases because the demand for highly automatable tasks decreases since some of these tasks are already being automated.

Our analysis works as follows: We initially set the share $s\left(t_i^k\right)$ of a task t_i^k as described in Section 4. We allow the share $s(t_i^k)$ of a task t_i^k to change by a factor which we denote by c_i^k. The weighted average of the adjusted share should be equal to c_i, i.e., $\sum_{k \in T_i} c_i^k \cdot s\left(t_i^k\right) \approx c_i$ to reflect the change in demand for a job. We assume that if one task can already be automated, i.e., has a factor smaller than one, then related tasks also tend to decrease and vice versa. We want to minimize the overall difference $s_{i,i'}^{k,k'} := |c_i^k - c_{i'}^{k'}|$ between two related tasks $t_i^k, t_{i'}^{k'}$.



(a) Tasks whose share decreases by at least a factor of 0.5



(b) Tasks whose share increases by at least a factor of 2

Figure 14: The histogram of the automation probability of all tasks whose share either decreases by at least factor of 0.5 or increases by at least a factor of 2.

This is modeled in the following linear program:

$$\begin{array}{rcll} \text{minimize} & \sum s_{i,i'}^{k,k'} & & \\ \text{s.t.} & & & \\ c_i^k - c_{i'}^{k'} \leq & s_{i,i'}^{k,k'} & \forall t_i^k, t_{i'}^{k'} \text{ that are related} \\ c_{i'}^{k'} - c_i^k \leq & s_{i,i'}^{k,k'} & \forall t_i^k, t_{i'}^{k'} \text{ that are related} \\ \sum_k c_i^k \cdot s\left(t_i^k\right) \leq & c_i\,(1+\varepsilon) & \forall j_i \in J \\ \sum_k c_i^k \cdot s\left(t_i^k\right) \geq & c_i\,(1-\varepsilon) & \forall j_i \in J \\ c_i^k \geq & 0 & \forall j_i \in J, t_i^k \in T_i \end{array}$$

To verify our claim, we first look at tasks whose factor c_i^k is less than 0.5, i.e., whose share has more than halved. We expect that these tasks to have a high automation probability. The histogram is depicted in Figure 14. Similarly, tasks whose share has increased by more than a factor of 2 should have a low automation probability. The results are also shown in Figure 14. Despite various other economic factors that come into play like economic cycles or major political events, we can see that tasks whose share has more than doubled tend to have a smaller automation probability.

9 Conclusion

We believe that automation is one of the main challenges for society. In our opinion, the seminal work of Frey and Osborne did an excellent job of getting the discussion going. In this paper we dug a bit deeper, by looking not only at jobs – but at the tasks that make up a job. We hope that opening the Frey/Osborne black box will contribute to the discussion. The professionals that are actually doing a job are the main experts to decide what parts of their job can or cannot be computerized. The Frey/Osborne work only tells these experts that their job is 87% automatable, but what does it actually mean? With our work, job experts can look inside the box, and understand which tasks of their job are at risk. Our hope is that the job experts have a discussion which results are believable and which are not, and why. To facilitate this discussion, we have created a web page (`http://jobs-study.ethz.ch`) that allows users to comment upon our results.

References

[1] Michael Auli, Michel Galley, Chris Quirk, and Geoffrey Zweig. Joint language and translation modeling with recurrent neural networks. In *EMNLP*, volume 3, page 0, 2013.

[2] David H. Autor and David Dorn. The growth of low skill service jobs and the polarization of the us labor market. Technical report, National Bureau of Economic Research, 2009.

[3] David H. Autor, Lawrence F Katz, and Alan B Krueger. Computing inequality: have computers changed the labor market? Technical report, National Bureau of Economic Research, 1997.

[4] David H Autor, Frank Levy, and Richard J Murnane. The skill content of recent technological change: An empirical exploration. Technical report, National Bureau of Economic Research, 2001.

[5] Holger Bonin, Terry Gregory, and Ulrich Zierahn. Übertragung der Studie von Frey/Osborne (2013) auf Deutschland. Technical report, ZEW Kurzexpertise, 2015.

[6] John Bound and George E Johnson. Changes in the structure of wages during the 1980's: An evaluation of alternative explanations. Technical report, National Bureau of Economic Research, 1989.

[7] Erik Brynjolfsson and Andrew McAfee. *Race against the machine: How the digital revolution is accelerating innovation, driving productivity, and irreversibly transforming employment and the economy*. Brynjolfsson and McAfee, 2012.

[8] Erik Brynjolfsson and Andrew McAfee. *The second machine age: work, progress, and prosperity in a time of brilliant technologies*. WW Norton & Company, 2014.

[9] Kerwin Kofi Charles, Erik Hurst, and Matthew J Notowidigdo. Manufacturing decline, housing booms, and non-employment. Technical report, National Bureau of Economic Research, 2013.

[10] L Deng, D Yu, and G Hinton. Deep learning for speech recognition and related applications. In *NIPS Workshop*, 2009.

[11] Martin Ford. *Rise of the Robots: Technology and the Threat of a Jobless Future*. Basic Books, 2015.

[12] Carl Benedikt Frey and Michael A. Osborne. The future of employment: How susceptible are jobs to computerisation?, 2013.

[13] Maarten Goos and Alan Manning. Lousy and lovely jobs: The rising polarization of work in britain. *The review of economics and statistics*, 89(1):118–133, 2007.

[14] Erico Guizzo. How google's self-driving car works. *IEEE Spectrum Online, October*, 18, 2011.

[15] John Maynard Keynes. Economic possibilities for our grandchildren (1930). *Essays in persuasion*, pages 358–73, 1933.

[16] Quoc V Le. Building high-level features using large scale unsupervised learning. In *ICASSP*, pages 8595–8598. IEEE, 2013.

[17] James Manyika, Michael Chui, Jacques Bughin, Richard Dobbs, Peter Bisson, and Alex Marrs. *Disruptive technologies: Advances that will transform life, business, and the global economy*, volume 12. McKinsey Global Institute New York, 2013.

[18] Pajarinen Mika and Anders Ekeland. Computerization and the future of jobs in norway. 2015.

[19] Marvin L Minsky. *Computation: finite and infinite machines*. Prentice-Hall, Inc., 1967.

[20] David Silver, Aja Huang, Christopher J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–503, 2016.

[21] Alan M Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.

[22] Moshe Y. Vardi. The great robotics debate. *Commun. ACM*, 56(7), July 2013.

[23] Moshe Y Vardi. Is information technology destroying the middle class? *Commun. ACM*, 58(2):5, 2015.

[24] Moshe Y. Vardi. The moral imperative of artificial intelligence. *Commun. ACM*, 59(5), April 2016.

[25] Adrian Wood. North-south trade, employment and inequality. *New York*, 1994.

[26] Adrian Wood. Globalisation and the rise in labour market inequalities. *The economic journal*, 108(450):1463–1482, 1998.

The Education Column

by

Juraj Hromkovič
Department of Computer Science
ETH Zürich
Universitätstrasse 6, 8092 Zürich, Switzerland
juraj.hromkovic@inf.ethz.ch

DEMYSTIFYING CODING FOR SCHOOLS — WHAT ARE WE ACTUALLY TRYING TO TEACH?

Tim Bell
University of Canterbury, NZ
tim.bell@canterbury.ac.nz

Abstract

As computer science enters school curricula around the world, many teachers are having to teach the topic of computer programming for the first time. For those who have little experience in the area it can be daunting, while those who are experienced programmers may be so familiar with the subject that it is hard to see what is difficult about it! This article explores the reality of what we need to be teaching in schools, and considers what the essence of the topic is.

1 Introduction

The idea of teaching computer science at all levels of the school system is gradually being adopted around the world (see for example [10, 16]), with the subject being introduced in the first year of school [6, 9].

Such changes are not usually driven by simply wanting to teach young children to be programmers, but to address broader issues such as helping students to be informed citizens in a digital world [8], and to reflect this, the subject usually has a broader name such as "computational thinking" or "computing". However, "coding" is becoming an important component in new curricula, and is a catchword for those promoting the discipline [3]. For those new to programming (especially established school teachers who have never programmed themselves), the prospect of teaching "coding" can be daunting. For those who are experienced programmers, it can be tempting to impose their own experience (perhaps as a self-taught programmer in their youth, or working at a professional level) onto the curricula being developed for use in schools.

When teaching a subject the focus can easily end up on details (such as print statement formats and while loop conditions), and these are indeed needed to prescribe a plan for teaching, but we need to stay aware of the big picture — what are the key concepts that students should take away from having learned about programming? What do we want them to remember a year later, or five years later? These are things that may be obvious to those who are heavily involved in computer science as a profession, but can be a great mystery to those from outside our culture who are having to prepare to teach this new topic.

This article reflects on what we really want students to take away from programming courses, and especially on what teachers will need to be aware of to achieve this. Some of the ideas discussed are misconceptions that we have commonly encountered while helping teachers to deliver the new subject, and others are ideas that may be so obvious to experienced computer scientists that we forget to articulate them to people who are new to our culture.

2 Teaching Programming

This section considers a number of issues that come up when teaching programming in schools, and looks at how they relate to the bigger picture of how programming is used in practice.

Learning to program isn't the same as learning to teach programming. Understanding the subject is clearly important, but to teach effectively there are many ideas that can be brought to bear on teaching programming, which even the most experienced programmer won't necessarily know about. Some ideas about teaching programming are well established, and others are the subject of great debate [14] — one particularly hotly debated issue is whether we teach objects first or later [1]. There are also various ideas about what is the best programming language for learners [17]. There is evidence that students can learn better working in pairs [18], and that it can be useful to use Parson's problems [13] while teaching programming. Ideas around teaching programming to young students are still being developed, as it is relatively recent that this has been done at scale in typical classrooms with typical teachers, rather than specialist clubs and outreach programs.

There are many books on how to program, but relatively few books on how to *teach* programming (although a few are now appearing e.g. [2, 11]). Because programming at all levels in schools is a new phenomenon for most countries, we are still at the early stages of research, and much of the knowledge about this is coming through conferences and journals that aren't where teachers would look for guidance.

Programs are used everywhere. A variety of terms, such as "app", "application", "code", "solution", "software" and "firmware", are used to refer to programs in various contexts, and a lay person may not realise that these things are all programs. Even simple-looking devices like a burglar alarm, WIFI router, or ticket vending machine may not be recognised as being based on a program, let alone devices like a smartphone or smart watch, which typically run many dozens of programs. This can be overcome to some extent when teaching the basics of programming by mentioning examples from everyday life when simple constructs are taught; for example, "count = count +1" might be used in a fitness tracking device to add one to the number of steps taken, or "if margin < 0" might be used in a word processor to determine if a value is out of range. This will help students to appreciate that what appears to be a meaningless exercise, possibly in an environment that doesn't look like anything used commercially, is actually the basis of many different forms of "program" that people develop for everyday use. And to make things even more interesting, a programming language itself is implemented by a program!

Meaningful experiences can help to understand jargon. A key outcome of learning programming and other computer science concepts early is that students become familiar with technical terms in a meaningful context. The new topics being introduced to schools come with a lot of jargon — words like "algorithm", "binary" and "coding" can make teachers genuinely fearful, and it's important to overcome this. Students may see these words as a lot of unnecessary jargon to memorise, and may be put off by their teacher's confusion with the terminology. We have found that it's effective to engage teachers new to the subject by first *using* the idea (such as showing an algorithm to calculate the checksum in a product barcode), and *then* labelling it (in this case distinguishing the "algorithm" from a program that implements the algorithm, and of course introducing the term "checksum") so that the mysterious terminology appears *after* the concept has been brought down to size.

For students, rather than memorising definitions of jargon like "debug", "algorithm", and so on, teachers can use the terms to label the experiences and tools that students are engaging with. If a student has regularly implemented even the simplest programs, they will have had to track down errors in it (debug), and by implementing even simple algorithms (like drawing a regular polygon in a turtle-based language, which can be acted out away from the computer and then implemented on a digital device) they can start to distinguish between concepts like an algorithm and a program. If the teacher uses this terminology regularly and appropriately when referring to something the student is doing, it will acquire an authentic meaning for them. Once when using this approach a young student asked "Why do you use such big words for such simple ideas?" Every discipline has its jargon, and allowing teachers and students to engage with the ideas first and *then* labelling them can help to have them seen as approachable concepts.

Programming is more about communicating with humans than with machines. Writing a computer program could be described as giving instructions to a computer, but much of the discipline of programming is around communication with humans via the computer, rather than the computer being the final recipient of what is written. The programmer is communicating to two sets of people: the users, and future programmers.

Almost every program involves human interaction, and the user experience is notoriously poor in many digital systems. Introductory programming is an excellent place to sensitise students to the importance of thinking about the user. Many beginners overlook giving any instructions to the user (e.g. which keys to press for a game, or what range of values is expected for an input), and the output can be jarring or confusing (e.g. a rather blunt "You are wrong" in response to a quiz answer, or an uninformative "Out of range, try again" for an input value).

The other important audience for a program is the next programmer who needs to work on it. Trying to read someone else's program helps a student to appreciate comments that explain what is happening at just the right level of detail, variable names that are unambiguous, and layout that follows conventions, so they can focus on the content and not the form. The future programmer may even be the original programmer, trying to make sense of what they wrote a year ago — or perhaps even a day ago!

In both cases it can be useful for a student to swap their program with a neighbour and see if they can figure out how to use it, or look at the code and work out how it works. Blackwell [4] points out that "many skills of a professional programmer are related to social context rather than the technical one", and even simple introductory programming environments can help students develop skills for this social context.

Programming integrates well with other subjects. Rather than teaching programming in isolation, it can be integrated with other areas of the curriculum [12], and this is particularly natural at primary school where the same teacher may be teaching the different subjects. For example, at a junior level students might learn about odd and even numbers, and writing a program to print the odd numbers forces them to articulate the meaning. At a more senior level, students might learn a definition of prime numbers, and again articulate it as a program that demonstrates their understanding. Turtle-graphics languages such as Scratch or Logo also naturally use mathematical concepts like positive and negative numbers, coordinates, angles, and other concepts from geometry.

Many other subjects can also be integrated: literacy becomes important if the program is telling a story or providing an understandable textual interface, music can be represented through programming, and physical fitness can be involved in acting out coordinate based instructions. These examples just scrape the surface of many examples of integrated learning that we have observed.

Programming is a skill that demands practice. This is related to the pedagogy of programming: some students (and teachers) have a model of learning a subject that there are some facts to learn, and once you know them you are competent in the subject. This view relates to working at a low level of Bloom's taxonomy, but programming is a very creative activity, where the programmer is generally operating at a high level of the taxonomy. This misconception can be reflected in wanting to learn programming by reading a book, or perhaps attending a short course, and hoping that from then on one is able to program.

A better model is to think of programming more like fitness training or learning a foreign language; regular exercises are far more important that trying to learn it in a hurry and hoping to know it some time later. Robins [15] introduces a "Learning Edge Momentum" model, which highlights that in programming it is particularly important to understand the basic concepts, as later concepts will make no sense if one basic concept is missing (for example, objects won't make sense if the concept of type isn't understood, or *for* loops will be confusing if the role of a variable isn't clear).

To support teachers new to programming, we encourage them to do small regular exercises to keep up their "fitness level"; this also applies to students, as it is much better to do a little regularly with programming, rather than, say, a short segment of a course where they write one large program. Of course, working on such rudiments needs to be balanced with more motivational large projects, but large projects alone can be frustrating if students don't have the skills to draw on.

Technology changes quickly, but the basics don't. A concern that teachers often raise is that the computing is changing so fast that they are worried that even if they learn to teach the subject, their knowledge will go out of date quickly. While it's true that technology keeps changing, the basic ideas around computer science and programming don't change so rapidly. Since teaching is more about laying foundations, focussing on the basics is appropriate.

For example, a new introductory language might become popular, but chances are it has very similar ingredients to ones that already exists. In principle, any language that is Turing-complete is sufficient to fully control any conventional computing device, and so from an educational point of view we don't need to be concerned that learning to program in one language will be of no use for learning to program a new one in the future. In fact, the key is that we are teaching *programming*, not a particular language. A cue for students is to call the subject *programming*, not Python, or Java, or Scratch.

The Böhm-Jacopini theorem [5] underpins the idea that teaching programming should cover three basic constructs: sequence, selection and iteration. In principle, these are the only control structures needed to program any computing device. In addition to these control structures, a language needs input, output and storage (variables) to give sufficient scope to program anything that is computable. An important consequence of this observation is that "toy" languages like Scratch are actually just as capable as the most advanced languages, or at least, they capture the key logic needed to make things happen on a computational device, and the differences between languages highlight features that make programming more convenient for particular applications, rather than some fundamental new capability. Also, not all introductory programming systems are Turing-complete — for example, the popular "Beebot" and "ScratchJr" teaching tools focus on sequence, but this can be seem as an important stepping stone that is aimed at an appropriate cognitive level for young students.

Blackwell [4] reflected on what programming is from a cognitive point of view. As well as highlighting the boundaries of programming (for example, writing HTML or setting a microwave oven isn't programming because the system isn't Turing-complete), Blackwell notes that programming involves more than just writing code; the programmer must identify requirements, derive a specification, design how it will work, code the commands, and debug it to be sure that it will function as intended. He highlights that programming reflects a loss of direct manipulation: the programmer must anticipate what will happen in advance (e.g. all combinations of user input), and account for these before the program is run. These skills can be exercised in the simplest of programming systems, and in this light, "coding" is a relatively small part of the whole process of writing a successful program.

Along with this is the need for persistence; writing a program is easy, but debugging it is the real challenge, and persistent work is required to make sure the program works properly, rather than making do with something that is almost correct [7].

Teaching core concepts well is better than covering every possible technique. When designing computing courses, we need to be careful to focus on quality rather than quantity. There are endless programming languages, environments (mobile, web, desktop, server) and toolkits that could be taught, but the goal should be to provide students with a good grounding, and inspire them to learn more, rather than overwhelm them with so many topics that it has the effect of putting them off the subject. The same applies to teachers: if a teacher is pressured to deliver a curriculum that beyond what they have had the professional development for, the students may end up getting a poor experience of learning to program, and may go away with the impression that programming is difficult and confusing.

Computer science is much more than programming. While this article has focussed on programming, it's important that computing courses take a much broader view of the discipline. Programming enables us to make things happen, but there is a lot to know before we can write effective and efficient programs, which is informed by the field of computer science. There are intriguing ideas in algorithm design — some algorithms are staggeringly more efficient than others for solving the same problem, while other problems have no known programmable solution that will work in a reasonable amount of time. Programs operate on data, and how that data is represented has an effect on how effectively it can be processed. Computers need to communicate with each other, and the programs that do this need to follow appropriate protocols to make sure that works well. When computers communicate with humans, the way the operate needs to be informed by a basic understanding of how humans think and perceive. And beyond the basics there are so many more questions: can we imitate human intelligence? Can we simulate processes from the physical world? Or can we create new virtual worlds? Are there things we could implement, but shouldn't?

These are all questions that the discipline of computer science is concerned with, and students can engage with these ideas even before they write programs e.g. using non-computer based activities such as "CS Unplugged" (`csunplugged.org`). In fact, it is valuable that they have such experiences because for some students this will provide the motivation to learn to program; while some may enjoy programming for its own sake, others will be more motivated if they can see how it can be applied, and that there are tools and concepts beyond programming that are exciting and relevant to our human world.

The epigram that "computer science is no more about computers than astronomy is about telescopes" captures this idea when applied to programming; it's a tool that is normally used to make things happen in a digital world, but it is a means, not the end in itself. Computer science courses often start (and sometimes end!) with programming, and this can give an inaccurate message to students of what the discipline is about. By keeping students aware of the bigger picture, we are more likely to capture their interest and give them a balanced view of what matters.

3 Conclusion

We are at an exciting point in education, where many countries are adding a whole new subject to their curriculum that hasn't been taught before. Empowering teachers to deliver this with enthusiasm is important, and the ideas shared above are intended to help us think about approaching this change in a way that neither underplays how significant the change is, but also doesn't make it so overwhelming that the value of the change is lost because schools are unable to deliver it effectively.

References

[1] Owen Astrachan, Kim Bruce, Elliot Koffman, Michael Kölling, and Stuart Reges. Resolved: objects early has failed. *ACM SIGCSE Bulletin*, 37(1):451–452, 2005.

[2] Phil Bagge. *How to Teach Primary Programming Using Scratch*. The University of Buckingham Press, 2015.

[3] Tim Bell. Surprising Computer Science. In Andrej Brodnik and Jan Vahrenhold, editors, *8th International Conference on Informatics in Schools: Situation, Evolution, and Perspective*, pages 1–11. Springer, 2015.

[4] A Blackwell. What is programming. In *14th workshop of the Psychology of Programming Interest Group*, pages 204–218, 2002.

[5] Corrado Böhm and Giuseppe Jacopini. Flow diagrams, Turing machines and languages with only two formation rules. *Communications of the ACM*, 9(5):366–371, may 1966.

[6] Neil C C Brown, Sue Sentance, Tom Crick, and Simon Humphreys. Restart: The Resurgence of Computer Science in UK Schools. *Trans. Comput. Educ.*, 14(2):9:1–9:22, jun 2014.

[7] Quintin Cutts, Emily Cutts, Stephen Draper, Patrick O'Donnell, and Peter Saffrey. Manipulating mindset to positively influence introductory programming performance. In *Proceedings of the 41st ACM technical symposium on Computer science education*, pages 431–435. ACM, 2010.

[8] Caitlin Duncan, Tim Bell, and Steve Tanimoto. Should your 8-year-old learn coding? In *Proceedings of the 9th Workshop in Primary and Secondary Computing Education - WiPSCE '14*, pages 60–69, New York, New York, USA, nov 2014. ACM Press.

[9] Katrina Falkner, Rebecca Vivian, and Nickolas Falkner. The Australian Digital Technologies Curriculum: Challenge and Opportunity. *Proceedings of the Sixteenth Australasian Computing Education Conference (ACE2014)*, pages 3–12, 2014.

[10] Walter Gander, Antoine Petit, Gérard Berry, Barbara Demo, Jan Vahrenhold, Andrew McGettrick, Roger Boyle, Avi Mendelson, Chris Stephenson, Carlo Ghezzi, and Others. Informatics education: Europe cannot afford to miss the boat. `http://europe.acm.org/iereport/ie.html`, 2013.

[11] Mark Guzdial. Learner-Centered Design of Computing Education: Research on Computing for Everyone. *Synthesis Lectures on Human-Centered Informatics*, 8(6):1–165, nov 2015.

[12] Irene Lee, Fred Martin, and Katie Apone. Integrating Computational Thinking Across the K–8 Curriculum. *ACM Inroads*, 5(4):64–71, dec 2014.

[13] Dale Parsons and Patricia Haden. Parson's programming puzzles: a fun and effective learning tool for first programming courses. In *Proceedings of the 8th Australasian Conference on Computing Education-Volume 52*, pages 157–163. Australian Computer Society, Inc., 2006.

[14] Jody Paul. "What First?" Addressing the Critical Initial Weeks of CS-1. In *Proceedings. Frontiers in Education. 36th Annual Conference*, pages 1–5. IEEE, 2006.

[15] Anthony Robins. Learning edge momentum: A new account of outcomes in CS1. *Computer Science Education*, 20:37–71, 2010.

[16] Josh Tenenberg and Robert McCartney. Editorial: Computing Education in (K-12) Schools from a Cross-National Perspective. *Trans. Comput. Educ.*, 14(2):6:1–6:3, jun 2014.

[17] David Weintrop and Uri Wilensky. To block or not to block, that is the question: students' perceptions of blocks-based programming. In *Proceedings of the 14th International Conference on Interaction Design and Children*, pages 199–208. ACM, 2015.

[18] Linda Werner and Jill Denning. Pair programming in middle school: What does it look like? *Journal of Research on Technology in Education*, 42(1):29–49, 2009.

The Logic in Computer Science Column

BY

Yuri Gurevich

Microsoft Research
One Microsoft Way, Redmond WA 98052, USA
gurevich@microsoft.com

The 1966 International Congress of Mathematicians: A micro-memoir

Yuri Gurevich

To the memory of Boris (Boaz) Trakhtenbrot,
a founding father of theoretical computer science
and a good friend who will be greatly missed

Why the memoir? Well, the author of the scheduled article became unwell. (He should be up and running soon, and his article will appear at some later time.) Facing the prospect of missing an issue, I thought about what I can do in a few days before the deadline. It occurred to me that 2016 is the 50th anniversary of the 1966 International Congress of Mathematicians. Maybe the impressions of a young Soviet mathematician would be of interest to at least some of the readers.

Figure 1: Boris Trakhtenbrot, Sophia Yanovskaya and Alexander Kuznetsov at the 1966 congress (first row, right to left)

1 A hole in the iron curtain

Quisani[1]**:** I guess an International Congress of Mathematicians is a big deal?

Author: It has an illustrious history [5]. The first congress was held in Zurich in 1897. Hilbert's famous list of problems was presented at the 1900 congress, in Paris. The 1966 congress was special for Soviet mathematicians because it was held in Moscow. We had been deprived of contacts with our Western colleagues.

Q: Were Western books and journals available in the USSR?

A: There were good mathematical libraries in Moscow, Leningrad and Novosibirsk. I know of no other place with a good mathematical library. To buy Western books or to subscribe to Western journals, you needed foreign currency. The Ural State University, where I was teaching at the time, had zero foreign currency for that purpose, even though it was the main university of the whole area of the Urals.

Q: Could you move to Moscow?

A: For all practical purposes, no. The Soviet system was feudal in some respects. The right to live in Moscow normally went from parents to their children. Even if some Moscow institution was eager to hire you, in most cases they would not be able to do that [6].

Q: Could you talk to foreigners?

A: That would be risky. Besides, there weren't many foreigners to talk to. The Urals were completely closed to foreigners.

Q: To all foreigners or only to Westerners?

A: I believe the Urals were closed to all foreigners, though there could have been some tightly controlled exceptions. I remember that, right after the congress, the Polish logician Bogdan Węglorz and I considered flying together to Novosibirsk, but I needed to stop in Sverdlovsk (the cultural center of the Urals, now called Yekaterinburg), and Bogdan was not allowed to stop there.

Q: At the congress, were your Western colleagues willing to talk to you?

A: Yes. They — and especially the Americans — were very smiley. Of course I smiled back, but my smiling muscles were not sufficiently developed. At the end of the congress, those muscles were aching.

[1] My former student.

Q: Were you particularly gloomy?

A: I don't think so. We, in Russia, just didn't smile that often, certainly not at strangers. "A laugh without a reason is a sign of folly," says a Russian adage[2].

A: Did you have time to speak to your Soviet colleagues?

Q: Of course. On one occasion, I was running from one building to another, and I met Anatoly I. Malcev, the father of Soviet model theory. Malcev was super polite, and we had a small talk about the difficulty of disseminating one's own results. "Find a famous guy," joked Malcev, "grab him by a button and force feed him your results."

2 Logic

Q: The congress, I am sure, was very interesting to you scientifically. Give me a couple of highlights.

A: One highlight for me was a beautiful lecture on nonstandard analysis given by Abraham Robinson, the originator of that field. Infinitesimals used to be a fundamental notion in mathematical analysis. Eventually, for the purpose of rigor, their use was replaced by ε-δ arguments. Intuitively a positive infinitesimal ι is an infinitely small real number different from zero, so that we have

$$0 < \iota < 1/n \quad \text{for all positive integers } n. \tag{1}$$

Q: But this makes no sense.

A: Indeed, it seems so. Yet nonstandard analysis provides a solid foundation for (a careful use of) infinitesimals.

Q: Explain.

A: Extend the first-order theory Th($\mathbb{R}$) of the ordered real field $\mathbb{R}$ with a constant symbol ι and the axioms (1). Every finite subset of these axioms is obviously consistent with Th($\mathbb{R}$). By the compactness theorem, the set of all these axioms is consistent with Th($\mathbb{R}$) and thus has a model $\mathbb{R}^*$. In that model, the value of ι is a positive nonstandard real number that is less than any standard positive real number.

[2]Смех без причины — признак дурачины.

Figure 2: Abraham Robinson at the congress

With infinitesimals, one can make rigorous the intuitive ideas underlying analysis. For example, a real-valued function f of a real variable is continuous at a real number a if and only if an infinitesimal change of the input produces an infinitesimal change of the output, i.e., for every infinitesimal ξ, the difference $f(a+\xi) - f(a)$ is infinitesimal. Moreover, if the ratio $\dfrac{f(a+\xi) - f(a)}{\xi}$ is infinitely close to a standard real number b for all nonzero infinitesimals ξ then b is the derivative of f at a.

At the end of his lecture, Robinson announced that he would give away a few copies of his book [7]. I approached him and asked whether I can have one. Only if you promise, he said, that you will read the book. I told him that I would do better, that I would conduct a seminar on nonstandard analysis. He gave me a copy of his book, and I indeed conducted such a seminar at the Ural State University.

Q: Was there anything at the congress that impressed you even more than nonstandard analysis?

A: Yes. This was the method of forcing presented at the congress by its inventor, Paul Cohen. Forcing allows you, given a set-theoretic world W, to construct new set-theoretic worlds by adding elements to W in a tightly controlled way.

Q: What is a set-theoretic world?

A: Typically, it is a model of ZFC, the first-order Zermelo-Fraenkel set theory with the axiom of choice.

Figure 3: Paul Cohen at the congress

Q: This seems similar to nonstandard analysis. There you extend a given model $\mathbb{R}$ of the first-order theory Th($\mathbb{R}$) while here you extend a given model W of the first-order theory ZFC. I guess the difference is that ZFC is more complicated than Th($\mathbb{R}$).

A: There is something else. You get a lot of mileage from an arbitrary nonstandard model $\mathbb{R}^*$ of Th($\mathbb{R}$). Any such $\mathbb{R}^*$ allows you to formalize nicely many intuitively appealing arguments that involve infinitesimals. In the set-theoretic case, an arbitrary extension of the given model doesn't buy you much. You need models with particular properties. Cohen used forcing to construct a model of ZFC where the continuum hypothesis (CH) fails [1]. Earlier, Gödel had constructed a model of ZFC + CH [2]. Thus Cohen proved the independence of the continuum hypothesis.

The independence result all by itself was an enormous achievement; at the congress, Cohen received a Fields medal [3] and in general was a big hero. But the forcing method was more important yet. Very quickly it radically changed set-theory. A great many problems in set theory, algebra, topology, etc. have been solved using forcing.

Q: Do you understand the method of forcing?

A: I learned forcing in the 1970s when I attended the logic seminar at the Hebrew University of Jerusalem. It was virtually impossible to be an active member of the seminar without understanding forcing.

But, during Cohen's lecture at the congress, I did not understand a thing. After his lecture, a group of Soviet mathematicians, about two dozen of us, asked Cohen to give us an informal lecture. Cohen spoke, in English, and took numerous questions. One of the Moscow mathematicians was translating. We stopped when, after a couple of hours, the translator was exhausted. We did not have a substitute translator from English. Cohen knew some French and German, but we did not have translators from those languages either.

As a result of that informal session I understood better why ZFC + CH was consistent. I had tried to study Gödel's book on the subject but it was too formal and dry. Cohen's explanation made Gödel's approach clear. But I did not understand why ZFC + the negation of CH was consistent. The whole idea of forcing looked mystical.

Q: Maybe I should read Cohen's book [1].

A: This will not be easy. Cohen gives a simple and beautiful proof of Gödel's result, but the proof of his own result is challenging. A streamlined exposition of Cohen's result was given by Shoenfield [8].

Q: Did you speak to Cohen?

A: I wouldn't dare to but he approached me and asked whether I am Yuri Gurevich. I said yes, I am, but I do not know anything about forcing. He smiled and said that he had been working on quantifier elimination for some field theories and that he wanted to know how I proved the decidability of the theory of ordered Abelian groups. I sketched my proof.

Q: Were there many logicians at the congress?

A: Everybody was there, it seems, except for Kurt Gödel, The authors of the logic books used to be just names for me. But there they were: Alonzo Church, Stephen Kleene, Alfred Tarski.

Q: Did you give a talk at the congress?

A: I gave a 15 minute talk, on the classical decision problem [4]. Translators had not been provided for short presentations but Haim Gaifman, of the Hebrew University of Jerusalem, helped me out. I split the talk into several pieces and explained every piece to Haim beforehand. During the presentation, every piece was presented first in Russian by me and then — loud and clear — in English by Gaifman.

Figure 4: Stephen Kleene, Alonzo Church and Anatoly Malcev at the congress

Figure 5: Anatoly Malcev and Alfred Tarski at the congress

Figure 6: Sol Feferman and Haim Gaifman at the congress

3 Language

Q: How did you communicate with Robinson, Cohen and Gaifman? Did they know some Russian?

A: No, they didn't. Russian is an important language in mathematics, and many Western mathematicians could read Russian mathematical texts. But, in my experience, very few of them spoke Russian. One exception was Alfred Tarski who spoke flawless Russian. Warsaw, his home town, was in the Russian Empire during his first 14 years.

Q: How was your English?

A: Nonexistent. I was a monolingual Russian speaker who never spoke any other language whatsoever. But burning desire is worse than fire[3]. I used an unprincipled concoction of German that I studied in school but never used and Yiddish that I heard at home. My parents spoke Yiddish between themselves but not to me. At the congress, my communication skills grew with use. On at least one occasion, I even translated a conversation between a Russian and an American. One of my American interlocutors joked that there were three universal languages: mathematics, music and Yiddish.

One peculiar problem was related to the fact that G and H are often rendered by the same Russian letter Г. "Hegel" becomes "Гегель." You have to guess whether it was "Gegel, Gehel, Hegel," or "Hehel," At one point, I asked Paul Cohen, in a written form, whether he met "Hödel". He looked puzzled but then figured out my intent and said that he did meet Gödel.

Q: Why the written form?

A: I mentioned above my conversation with Cohen. We spoke for a while, and then Cohen said that he wanted to go to a certain lecture. I went along, and we sat together. After a minute or two, he lost interest in the lecture, and I initiated the written exchange.

4 Epilog

Q: Were you afraid of possible consequences of your interaction with Westerners at the congress?

A: I was, of course. The Soviet Union was changing unpredictably after Stalin's death in 1953. The Khrushchev Thaw was a welcome development but it was

[3] Охота пуще неволи.

inconsistent. Besides, Khrushchev was removed from power in 1964. It wasn't clear at all in 1966 which way the country would go.

Q: Were all foreigners equally dangerous for you to talk to?

A: At the time I thought that, for me, it was most dangerous to talk to Israelis. I was happy that Haim Gaifman agreed to help me with my talk; that gave me an excuse to talk to him. By the way, I asked him whether he had a car. He said yes but not right now because his car hit some object. He named that object in many languages, and I still had no idea what it was until he waved his hands: it was a camel.

Q: After the congress, could you correspond with your Western colleagues?

A: The rules were unclear. You were supposed to ask for permission. Now imagine yourself being a Soviet academic bureaucrat. My letter arrives to you with a request to send it to the West. Since the rules are unclear, it is safer for you to deny the request or to postpone the decision indefinitely.

Instead of seeking permission to correspond, I went to the post office and sent my letters, hoping for the best. Amazingly, the letters usually made it through, and I received some letters from the West.

Q: I guess that the 1966 congress remains an unforgettable event for you.

A: Oh yes, it was a great learning experience. Also, I made many friends. When I was leaving the USSR in 1973, Sol Feferman, whom I met at the congress, invited me to Stanford "if you don't go to Israel." When I was seeking my first job and needed recommendation letters, my Soviet colleagues were unavailable. Abraham Robinson and Alfred Tarski graciously wrote for me.

Figure 7: The author shortly before the congress

Acknowledgement

Pictures 1 and 6, that is the pictures at Figures 1 and 6 respectively, are from the archive of Boris Trakhtenbrot. Pictures 2–5 were taken, at the congress, by Prof. Sergey V. Smirnov (1911-1979) of Ivanovo State Pedagogical Institute (now Ivanovo State University), and are published here with kind permission of his daughter Olga Smirnova. The university keeps Smirnov's *Nachlass*, and additional congress pictures can be found at [9].

I am grateful to Andreas Blass and Victor Marek for commenting on the draft of this micro-memoir. (Victor Marek was one of the young Polish logicians that I met at the congress.)

References

[1] Paul J. Cohen, "Set theory and the continuum hypothesis," W. A. Benjamin, 1966.

[2] Kurt Gödel, "The consistency of the continuum hypothesis," Princeton University Press, 1940

[3] Fields Medals 1966, `http://www.mathunion.org/publications/o/General/Prizes/Fields/1966`

[4] Yuri Gurevich, "The decision problem for predicate logic," *Soviet Mathematics* 7 (1966), 669–670

[5] International Congress of Mathematicians, *Wikipedia*, `https://en.wikipedia.org/w/index.php?oldid=711467665`

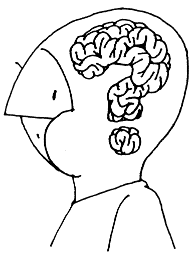
[6] Residency permits in the Soviet Union, *Wikipedia*, `https://en.wikipedia.org/w/index.php?oldid=738779705`

[7] Abraham Robinson, "Non-standard analysis," North-Holland, 1966

[8] Joseph R. Shoenfield, "Unramified forcing," in *Axiomatic set theory*, Proc. Sympos. Pure Math., XIII, Part I, American Math. Society, 1971, 357–381

[9] Sergey V. Smirnov, `http://math.ivanovo.ac.ru/hist/mos_66/mosk_66.htm`.

News and Conference Reports



Report on The EATCS Council Meeting

ICALP 2016, Rome, 12–13 July 2016

Luca Aceto
ICE-TCS, School of Computer Science, Reykjavik University

The EATCS Council meeting took place over lunch on the 12th and the 13th of July 2016 during ICALP 2016 in Rome This piece provides a brief and informal report on the discussions that took place at the meeting for the benefit of our members.

1 Tuesday, 12 July 2015: 13:00–14:50: Agenda and executive summary

1.1 Appointment of the new leadership of the EATCS

Josep Diaz presented the work of the search committee for the new president of the EATCS and the rationale behind the committee's proposal. The search committee suggested that Paul Spirakis be elected as new president of the EATCS. Paul proposed Leslie Ann Goldberg, Antonin Kucera and Giuseppe Persiano as vice-presidents. The council asked Paul for a short description of his programme as president of the EATCS and Paul answered some other questions from the council members. After this discussion, the council approved the new leadership, which will start its term of service after ICALP 2016 in Rome.

1.2 Report from the secretary

Ioannis Chatzigiannakis presented a report from the secretary office[1].

The president (Luca Aceto) thanked Efi Chita and Ioannis for their sterling work.

1.3 Report from the treasurer

Dirk Janssens gave his report on the finances of the EATCS[2]. After 27 years of sterling service, Dirk left his post as treasurer of the EATCS. The president thanked Dirk for his service to the association.

[1] http://icetcs.ru.is/luca/files/2016-secretary-report-GA.pdf

[2] http://icetcs.ru.is/luca/files/FinancialReport16.pdf

1.4 Approval of the accounts and discharging Dirk of financial obligations

Following Dirk's financial report, the EATCS Council approved the accounts of the association and formally discharged Dirk of financial obligations

1.5 Appointment of the new treasurer of the EATCS

The EATCS Council appointed Jean-Francois Raskin as the new treasurer of our association. The president thanked Jean-Francois for his willingness to serve.

1.6 Moving the seat of the association to Universit© Libre de Bruxelles

Dirk discussed the steps that remain to be taken in order to move the seat of the EATCS to Universit© Libre de Bruxelles (ULB). The EATCS Secretary Office is collecting the required information on all the changes to the EATCS Council since 1988.

Dirk also said that we will need to have an official translation of the statutes of the EATCS in French and that he has sought some professional help in order to settle the legal matters related to preserving the EATCS as a non-profit association under Belgian law and to moving its seat to ULB.

1.7 Journal of the EATCS: outcome of the feasibility study

Tony Kucera presented the report on the feasibility study conducted by a committed appointed by the president of the EATCS. The president of the EATCS had circulated the documentation prepared by the committee via email before the meeting.

Tony's presentation was followed by a very interesting and thought-provoking discussion. Some council members expressed worries related to the need of having another journal and to the fact that it was not clear that the journal could maintain a high standard and could compete with established journals. Some others said that, while perhaps there was no need for another journal, there was a need for a high-quality, open-access outlet for journal papers in TCS, broadly construed. Some council members expressed some doubt about the relationship that such a journal, if ever it sees the light of day, should have with the EATCS. Paul Spirakis suggested that we should find out in what way, for instance, the Journal of the ACM is related to the ACM as an association.

The president said that, in his opinion, the EATCS should establish the proposed journal only as a service to the TCS community, if the community thinks that there is indeed a need for a high-quality, open-access outlet for journal papers in TCS, broadly construed. The poll amongst the members of the EATCS that was held some time ago and email exchanges with prominent members of the community seemed to indicate that the community was supportive of the idea.

In order to address some of the worries related to the quality of the proposed journal in the first few years of its existence, if the EATCS ends up establishing it, the Council decided that the committee responsible for the feasibility study should prepare a manifesto for the journal and accompany it with a list of high profile TCS researchers who pledge to support it by submitting some of their best work to it in the five years of its existence and when it won't have an impact factor. Timothy Gower's Discrete Analysis journal got off the ground by publishing high profile papers from top class mathematicians and, in particular, a paper by Terence Tao on his solution of the Erdös Discrepancy Problem.

2 Wednesday, 19 July, from 13:00–14:50: Agenda and executive summary

2.1 One-off payment to the Helmut Veith Award fund

The president proposed that the EATCS contribute to the award in memory of Helmut Veith, established by the University of Vienna to support promising students. The Council approved the proposal to donate 500 Euros to the award fund.

2.2 Dealing with the increase in the LIPIcs APC

The president proposed that, for future editions of ICALP, the LIPIcs APC be paid by the ICALP participants as done this year. The proposal was approved by the EATCS Council.

2.3 ICALP 2016 report from the conference organizers

Tiziana Calamoneri presented the report from the organizers[3]. She also discussed the possible causes for the lack of workshops at this year's ICALP.

Tiziana's presentation was followed by a very lively discussion. Yuval Rabani said that a conference such as ICALP should have about 300 participants each time. He speculated that the cost of attending the conference are too high in general. Pierre Fraigniaud said that a way to keep the registration fees as low as possible would be to fix the fees beforehand and see what can be done with the

[3]http://icetcs.ru.is/luca/files/ICALP2016summary_Short.pdf

available funding. This is the model that has been adopted in the organization of HALG, where the registration fee was 100 Euros.

Jukka Suomela suggested that ICALP consider soliciting and accepting "brief announcements" like PODC does. There was some discussion on this point, with pros and cons of Jukka's proposal being aired. Jukka was given the task to prepare a proposal for "brief announcements" at ICALP that will be considered by the Council.

The council also discussed why ICALP is not considered an A* conference by some widely used rankings, such as CORE. The Council agreed that it was important that its members submit their best work to ICALP and that ICALP is advertised as the best, broad theory conference.

2.4 ICALP 2016 report PC chairs

Yuval Rabani delivered the report from the PC chairs[4]. The report was largely factual. During his report, Yuval mentioned the relatively low number of submission in learning to ICALP 2016.

2.5 Report on the organization of ICALP 2017

Mikolaj Bojanczyk presented a report on the status of the organization of ICALP 2017. The organization is very much on track and there are already four workshops that will co-locate with the conference. Three of the four invited speakers have been selected.

Tony Kucera asked whether ICALP should consider implementing a rebuttal phase during the PC meeting like other conferences do. During the ensuing discussion, several opinions were aired. Some felt that having a rebuttal phase would help to ensure that all the reports on the papers are delivered on time and would increase their quality. Others felt that papers for which a rebuttal phase might help clarify matters could by handled by other, lightweight means, such as contacting the authors for clarifications on controversial points, even if this might be seen as introducing a little unfairness in the process. The main constraint is the very short time that the ICALP PCs have for their reviews and for selecting a conference programme.

The Council should keep discussing this matter.

[4]http://icetcs.ru.is/luca/files/PC-report.pdf

2.6 ICALP 2018 bid presented

Ji™ Sgall presented a bid to host ICALP 2018 in Prague in the period 9–13 July 2018[5]. The bid was accepted by the General Assembly of the EATCS held on Thursday, 14 July.

2.7 The present and the future of ICALP Track C

Michael Mitzenmacher gave his opinion on the current status of Track C. Despite the best efforts of the PC chairs of the last few years to "brand" this track as a "theory of networking" track, it is fair to say that Track C is still being seen as a less competitive version of Track A. (Note: This opinion was independently reiterated by Mikkel Thorup during the general assembly. Mikkel asked: "What is the role of the current Track C rather than allowing PC members for Track A to submit to the conference?")

The president reminded the Council that Track C was meant to cover "emerging areas" and that its scope should therefore be regularly considered. During the ensuing discussion, Paul Spirakis suggested that perhaps Track C could be solely devoted to Algorithmic Game Theory.

The Council will continue discussing the future of Track C and what its focus should be from 2018 onwards.

2.8 EATCS Young Researcher Schools

Tony Kucera gave a report on the status of the EATCS Young Researcher Schools and presented a new joint series with ETAPS, SIGLOG and SIGPLAN that will kick off next year with a school on probabilistic programming. This joint series of schools will be held in 2017, 2018 and every two years thereafter. The council approved the establishment of this new school series and the EATCS sponsorship for it. From now on, the council will formally approve the sponsorship of EATCS schools.

Tony encouraged the Volume A community to submit proposals for Young Researcher Schools, especially in odd years. It was suggested that such schools might be co-located with HALG. Pierre Fraigniaud contacted Artur Czumaj, who chairs the SC for HALG.

[5] http://icetcs.ru.is/luca/files/ICALP2018.pdf

Report on The EATCS General Assembly

ICALP 2016, Rome, 14 July 2016

Luca Aceto
ICE-TCS, School of Computer Science, Reykjavik University

The annual general assembly of the EATCS was held at ICALP 2016 on Thursday, 14 July, from 16:30 till 18:15. The slides I used for the meeting are available[1], for those who are interested.

I started the general assembly by apologizing to the audience for the problems we had in making the official LIPIcs proceedings available by the conference date. (A preliminary version of the proceedings was available in the form of three large files, one per track, but those were very large and hard to download at the conference hotel. The EATCS Secretary prepared a dedicated web page[2] from which the files of the individual papers could be accessed, but this page was available too late.) This ICALP was the first edition of the conference with LIPIcs proceedings[3] and there were some associated teething problems. (ICALP is the largest conference ever to publish its proceedings with LIPIcs, as far as I know.) I am responsible for this problem and promised that it won't happen again.

During the ensuing discussion, Thore Husfeldt mentioned that, based on his experience as editor of a recent LIPIcs proceedings, he realized that we (theoretical computer scientists) are not good at following the given typesetting guidelines and that this makes the work of the proceedings editors and of the LIPIcs staff harder than it needs to be. (In passing, in a comment to the blog post on which this report is based, Marc Herbstritt from LIPIcs pointed out that some ICALP papers contains flaws that LIPIcs is still trying to resolve as part of publishing a high-quality proceedings volume. He also noted that most of the authors did not comply with the typesetting instructions they were given, which results in a huge amount of additional work for LIPIcs, and asked: "How come?")

I thanked Thore and asked the audience to help the proceedings chair and the LIPIcs staff by sticking to the typesetting instructions they are given. With electronic proceedings, one or two pages more don't matter and there is no point in trying to gain them by hacking the style files or using fonts that are forbidden by the publisher.

As a counterpoint, Mikkel Thorup and Yuval Rabani stated that they felt authors should not be bothered by strict typesetting guidelines, and that they should

[1] http://icetcs.ru.is/luca/files/GeneralAssembly2016.pdf

[2] http://www.eatcs.org/icalp2016/index.html

[3] https://www.dagstuhl.de/en/publications/lipics

spend their time doing good science rather than having to worry about typesetting guidelines from the publishers. Mikkel stated that "if it typesets, it should be good enough". He also suggested that a nice web interface to which authors could upload their papers for checking whether they meet the guidelines of the publisher would be very helpful.

I thanked all the contributors to the discussion. The EATCS will take all the suggestions into account and discuss them with LIPIcs. ICALP will also try to cooperate with other conferences and LIPIcs in order to develop some automated support that can help in preparing the proceedings efficiently and professionally.

I then remembered four colleagues who have left us too early: Hartmuth Ehrig, Zoltán Ésik, David Johnson and Helmut Veith. Obituaries for all these colleagues, apart from Zoltán Ésik, may be found in the June issue of the Bulletin of the EATCS[4]. I trust that contributions honouring the memory of Zoltán Ésik will appear in the October issue of the Bulletin. The EATCS Council decided to offer a small donation to the award in memory of Helmut Veith, established by the University of Vienna to support promising students. As usual, I invite the members of the TCS community to honour the memory of the aforementioned colleagues by building on their work and disseminating it amongst our students.

Tiziana Calamoneri delivered the report on ICALP 2016 from the conference organizers. The conference had 239 registered participants, 205 of whom registered by the early registration deadline. In her presentation, Tiziana also analyzed some of the reasons for the lack of workshops at this year's edition of ICALP. (Tiziana's slides are available[5].)

Yuval Rabani, who chaired the PC for Track A of ICALP, delivered the report on the PC chairs. (The slides are at `http://icetcs.ru.is/luca/files/PC-report.pdf`.) Yuval said that chairing the PC of Track A was an unexpectedly pleasant experience and thanked his PC for the splendid work it had done. Apart from reporting on the figures related to accepted and submitted papers, Yuval described the selection process for Track A, building on his blog posts on the topic[6]. Quoting from Yuval's blog,

> The committee identified around 50 borderline papers, and we had to choose among them 5 or 6 papers. (For those familiar with EasyChair, most papers with scores 2, 1, 1 were rejected.) Choosing those 5–6 papers out of 50 or 51 papers took up about half of the discussion time, because it was indeed a difficult choice. We felt that almost all of the borderline papers could have ended up in the program. The final choice was made, in part, by assessing the "added value" to already

[4] `http://bulletin.eatcs.org/index.php/beatcs/issue/view/21`
[5] `http://icetcs.ru.is/luca/files/ICALP2016summary_Short.pdf`
[6] `http://yrabani.tumblr.com/`

> chosen papers. For 2 of the 6 slots we ended up voting between 2–3 alternatives for each slot (papers in the same area that were thought to be of about the same quality). Aside from these few last papers, we devoted almost no attention to balancing subareas of theory. Papers were accepted based on pure merit, as judged by experts. Despite the indifference to areas, I think the program came out rather balanced between algorithms and complexity theory, with a nice presence in specialized niche areas. This is a natural outcome of a diverse committee.

Immediately after Yuval's presentation, I handed out the awards for the best papers and the best student papers at ICALP 2016. The best paper awards went to the following papers:

- Andreas Galanis, Andreas Göbel, Leslie Ann Goldberg, John Lapinskas and David Richerby. Amplifiers for the Moran Process. (Track A)
- Neeraj Kayal, Chandan Saha and Sébastien Tavenas. An almost Cubic Lower Bound for Depth Three Arithmetic Circuits. (Track A)
- Olivier Bournez, Daniel Graça and Amaury Pouly. Polynomial Time corresponds to Solutions of Polynomial Ordinary Differential Equations of Polynomial Length. (Track B)

The following papers received the best student paper awards:

- Samuel Hetterich. Analysing Survey Propagation Guided Decimation on Random Formulas. (Track A)
- Keerti Choudhary. An Optimal Dual Fault Tolerant Reachability Oracle. (Track C)

Congratulations to the authors of the award-receiving papers!

Mikolaj Bojanczyk gave a short report on the organization of ICALP 2017 on behalf the organizing committee. ICALP 2017 will be held in Warsaw, Poland, in the period 10–14 July 2017. The PC chairs will be Piotr Indyk (MIT, USA) for Track A, Anca Muscholl (LaBRI, France) for Track B and Fabian Kuhn (Freiburg, Germany) for Track C. Three invited speakers have already been confirmed. They are Mikolaj Bojanczyk (Warsaw, Poland), Monika Henzinger (Vienna, Austria) and Mikkel Thorup (DIKU, Denmark). A fourth invited speaker will be announced soon.

Mikolaj mentioned that four workshops have already agreed to co-locate with ICALP 2017. If you are interested in organizing a workshop at ICALP 2017, please contact the local organizers.

Jiří Sgall presented a bid to host ICALP 2018, the 45th ICALP, in Prague in the period July 9–13, 2018. The slides for Jiří's presentation are at `http://icetcs.ru.is/luca/files/ICALP2018.pdf`. The bid from Prague was accepted by the General Assembly. Thanks to Jiří and his colleagues for their kind offer to host ICALP in the beautiful city of Prague! ICALP 2017 and 2018 will also allow us to celebrate the excellent contributions of the Polish and Czech research communities to TCS and discrete mathematics.

After the ICALP-related presentations, I asked the audience the following questions:

- Does ICALP cover TCS sufficiently broadly?
- What do you think of the current acceptance rates at ICALP?
- What would you like to see at ICALP that we don't do?
- Do you have any criticisms/kudos/suggestions?

There were interesting suggestions from several colleagues. In particular, there was a lively discussion related to the role of the current incarnation of Track C. Despite the best efforts of the PC chairs of the last few years to "brand" this track as a "theory of networking" track, it is fair to say that, despite the high quality of the contributed papers, Track C is still being seen by many as a less competitive version of Track A. This opinion was, for instance, aired by Mikkel Thorup. In particular, Mikkel asked: "What is the role of the current Track C rather than allowing PC members for Track A to submit to the conference?" I reminded the audience that Track C was meant to cover "emerging areas" and that its scope should therefore be regularly considered. During the ensuing discussion, Paul Spirakis suggested that perhaps Track C could be solely devoted to Algorithmic Game Theory. Summing up, the EATCS Council will examine the future of Track C of ICALP in its coming meetings.

Mikkel Thorup also suggested that the submitted versions of the accepted ICALP papers should be posted on the conference web page as soon as they are accepted. This suggestion led to further interesting discussions. To my mind, it would certainly be beneficial to post the final versions of the accepted papers on the conference web site as soon as they arrive.

Thore Husfeldt suggested that the EATCS establish an SC for the conference, possibly independent of the council, and that the EATCS consider establishing a "fast track" for the publication of journal versions of the best ICALP papers. Regarding the first point, I informed the audience that the EATCS already has an ICALP Liaison Committee, but that it would be a good idea to give more power and responsibilities to it. That committee should also revise the current version of the guidelines for ICALP organizers, which are definitely out of date in the light of the new publication outlet for the proceedings and the new awards sponsored by the EATCS. I also informed the audience that the EATCS Council has been discussing the possible establishment of an open-access journal of the association for some time.

Regarding awards, the audience suggested that the EATCS consider establishing an ICALP Test-of-Time Award. A young researcher even suggested that ICALP should have a best reviewer award.

I thank the attendees for their many suggestions and invite any reader of this post to send theirs to the president of the EATCS or as comments to this post. You are the life and blood of the association. Your input is always most welcome and the EATCS listens to you. We are here to serve.

Next the secretary[7] and the treasurer[8] of the EATCS delivered their annual reports. We also thanked Dirk Janssens who left his post as treasurer of the EATCS after 27 years of sterling service to the association. We welcomed Jean-Francois Raskin as the new treasurer of the EATCS.

The rest of the general assembly was devoted to a report from the outgoing president (viz. me). I refer you to the slides for my presentation and to the EATCS Annual Report[9] for the details. Here I will limit myself to saying that at the general assembly I announced the new leadership of our association for the coming two-year term. The new president of the EATCS will be Paul Spirakis (University of Liverpool and U. Patras). He will be supported by Leslie Ann Goldberg (University of Oxford), Antonin Kucera (Masaryk University) and Giuseppe Persiano (University of Salerno) as vice-presidents.

During the general assembly, Paul gave a short speech describing some if his objectives as president of the EATCS for the coming two years. The EATCS is in very good hands and I look forward to seeing its influence grow under its new leadership.

Let me close this report by asking my readers and the members of the TCS community at large the questions I posed to the colleagues who attended the general assembly:

- What should the EATCS do for the TCS community?
- What activities should the EATCS support (financially or otherwise)?
- How can we make EATCS membership more attractive (especially among the younger generations)?

[7]`http://icetcs.ru.is/luca/files/FinancialReport16.pdf`
[8]`http://icetcs.ru.is/luca/files/2016-secretary-report-GA.pdf`
[9]http://icetcs.ru.is/luca/files/EATCS-report-2016.pdf

Any input you might have will be useful for the new leadership of the EATCS. Make your voice heard, so that the EATCS can serve the TCS community even better than it is already doing.

I thank all of you for the support I have received over the last four years in my role of president of the EATCS. It was a lot of work (to achieve probably very little), but I learned much from many of you. Thank you! You are the life and blood of the EATCS.

Report on ICALP 2016

The 43rd International Colloquium on Automata, Languages and Programming

Luca Aceto
ICE-TCS, School of Computer Science, Reykjavik University

ICALP 2016 took place in Rome from the 12th till the 15th of July 2016. The conference, which brought ICALP to Italy for the fifth time, was well organized by Tiziana Calamoneri, Irene Finocchi, Nicola Galesi and Daniele Gorla, whom I thank for the effort they put into making ICALP 2016 a memorable event.

According to the data presented by Tiziana on behalf of the local organizers during the General Assembly of the EATCS held on Thursday, 14 July, ICALP 2016 had 239 registered participants, 74 of whom were students. The USA was the country contributing the largest share of attendees (50), followed by France, the UK, Germany and Italy. Let me note, in passing, that I would have expected a larger number of participants from Italy, given the size of the Italian TCS community, the number of TCS researchers based in Rome and in neighbouring cities, and the ease with which Rome can be reached from most of the country. (Italy contributed 21 participants to the conference.)

ICALP 2016 featured four invited talks, which were delivered by Devavrat Shah (MIT, USA), Xavier Leroy (INRIA, France), Seffi Naor (Technion, Israel) and Marta Z. Kwiatkowska (Oxford, UK), as well presentations by the recipients of the Presburger Award, the Gödel Prize and the EATCS Award.

Devavrat Shah kicked off the conference on Monday, 12 July, by delivering a talk entitled *Computing Choice*. In his talk, Devavrat discussed algorithmic results relating to ranking, rank aggregation and personalized rankings associating intensity to rankings based on partial information resulting from a sparse set of comparisons. The talk, which was excellently paced and interesting, presented many results and I invite you to check Dev's work for the details. This work addresses computational challenges for decision making without a choice model, and offered a glimpse of the exciting possibilities for inter-disciplinary work across disciplines such as CS, EE, OR and Economics.

Xavier Leroy delivered the second invited talk, entitled *Formally verifying a compiler: What does it mean exactly?*, on Wednesday, 13 July. In his talk, Xavier discussed the context for, and the results of, the CompCert project[1], which investigates the formal verification of realistic compilers usable for critical embedded software. Such verified compilers come with a mathematical, machine-checked

[1] http://compcert.inria.fr/

proof that the generated executable code behaves exactly as prescribed by the semantics of the source program. In this project, Coq is used both as a proof assistant and as a programming language.

In his talk, Xavier said that "Pure functional programming is the shortest path to writing and verifying software." He also asked and addressed two fundamental questions arising from this work and related ones:

- Did we prove it (the compiler) right?
- Did we prove the right thing?

In particular, Xavier discussed the latter question in detail and argued that the social consensus underlying the acceptance of proofs in mathematics also plays a role in accepting proofs of software correctness. He also mentioned the "unreasonable effectiveness of labelled transition systems" in semantics and in supporting such correctness proofs.

Seffi Naor's talk took place on Thursday, 14 July, and was entitled *Maximatization of submodular functions: Recent progress*. Seffi stepped in at the last moment for Subhash Khot, who was unable to make the trip to Rome. On behalf of the EATCS and of the TCS community as a whole, I thank him for delivering an excellent talk at such a short notice.

Research on the topic of Seffi's talk started in the 1950s–1960s and is now thriving. It has applications in the study of social welfare, economics/game theory, combinatorial optimization, machine learning and information theory. In his talk, Seffi first surveyed results on unconstrained maximization of non-monotone functions, with focus on approximation algorithms, and then presented results for the constrained maximization problem. I refer the readers to Seffi's papers and to the Wikipedia page at `https://en.wikipedia.org/wiki/Submodular_set_function` for more information.

The last invited talk at ICALP 2016 was delivered by Marta Z. Kwiatkowska on Friday, 15 July. Marta's talk was entitled *Model Checking and Strategy Synthesis for Stochastic Games: From Theory to Practice* and is accompanied by a paper[2] that appears in the conference proceedings. Marta stated right at the start that, despite the success that model checking and synthesis techniques have had and are having, we have not found yet the right modelling abstractions for autonomous mobile agents such as robots and autonomous vehicles. Software for these vehicles is expected to behave reliably under uncertainty, and its analysis and synthesis require quantitative approaches to specification and verification. As Marta argued cogently in her talk, a game-theoretic point of view is fruitful in the study of such systems. Indeed, games of various kinds have played a fundamental

[2] `http://qav.comlab.ox.ac.uk/bibitem.php?key=Kwi16`

role in the study of the synthesis of correct programs from specifications from the very beginning, and papers on game-theoretic models abound in Volume B conferences. (See the slides for a recent talk by Moshe Vardi[3] for historical remarks and an overview of the game-theoretic approach to synthesis.) Rather than attempting to summarize Marta's talk, I strongly encourage you to read her accompanying paper, which beautifully summarizes her work on this topic and contains pointers to related literature.

The core of the scientific programme consisted of the papers that were selected for presentation by the PC chairs (Michael Mitzenmacher, Yuval Rabani and Davide Sangiorgi) and their PCs. Because of EATCS commitments, I could not attend as many talks as I would have liked, but all those I did manage to listen to were excellent both scientifically and from the point of view of the quality of the presentation. (For one of the talks[4], I even had to wear 3D glasses!) Thanks to the PC chairs and their PCs for doing a truly great job!

The award ceremony was held on Wednesday, 13 July, and saw the presentation of the EATCS Distinguished Dissertation Awards, of the Presburger Award to Young Scientists, of the Gödel Prize and of the EATCS Award. The event was a festive occasion and celebrated some of the outstanding members of the TCS community.

The EATCS Distinguished Dissertation Award Committee, which consisted of Javier Esparza, Michal Feldman, Fedor Fomin, Luke Ong and Giuseppe Persiano (chair), has selected the following three theses for the EATCS Distinguished Dissertation Award for 2015:

- Radu Curticapean, The Simple, Little and Slow Things Count: On Parameterized Counting Complexity. Thesis work carried out at the Department of Computer Science at Saarland University, Saarbrücken, Germany. Supervisor: Markus Bläser.

- Heng Guo. Complexity Classification of Exact and Approximate Counting Problems. Thesis work carried out at the Department: of Computer Sciences in the University of Wisconsin-Madison. Advisor: Jin-Yi Cai,

- Georg Zetzsche. Monoids as storage mechanisms. Thesis work carried out at the Department: of Computer Science at University of Kaiserslautern. Supervisor: Roland Meyer.

The award committee received an impressive set of submissions in terms of quality. The three selected theses are outstanding.

[3]http://www.cs.rice.edu/~vardi/papers/sr15.pdf

[4]https://arxiv.org/abs/1605.07061

The Presburger Award was presented to Mark Braverman (Princeton University, USA). The Gödel Prize went to Stephen Brookes and Peter O'Hearn for their invention of concurrent separation logic, and the EATCS Award was given to Dexter Kozen. The presentation of each of these three awards was accompanied by an excellent talk by the award recipient(s). As I mentioned during the award ceremony, this might very well be the first time that the Gödel Prize is mentioned in a piece in the New Yorker[5].

The award session was extremely well attended, and preceded a short bus tour in Rome and a social dinner in a popular restaurant in Trastevere.

The annual General Assembly of the EATCS took place on Thursday, 14 July. A report on the General Assembly is also published in this issue of the Bulletin of the EATCS. Here I will limit myself to saying that, at the General Assembly, I formally stepped down as president of the EATCS after two terms of service (four years). The new president of the EATCS will be Paul Spirakis (University of Liverpool and U. Patras). He will be supported by Leslie Ann Goldberg (University of Oxford), Antonin Kucera (Masaryk University) and Giuseppe Persiano (University of Salerno) as vice-presidents. At ICALP 2016 in Rome, Dirk Janssens also left his post as treasurer of the EATCS after 27 years of sterling service. Jean-Francois Raskin kindly accepted to serve as the new treasurer of our association. The association is very grateful to the above-mentioned colleagues for their willingness to serve and to Dirk for his outstanding service over such a long time. I know that the EATCS community will support the members of the new leadership in their work, just like they helped me during the last four years.

If you were at ICALP in Rome and you have any comment, suggestion or criticism, please send them to `president@eatcs.org`. We are always working on improving an already very successful conference that does its best to provide a bird's eye view of TCS as a whole.

[5] `http://www.newyorker.com/tech/elements/waiting-for-godel`

Report from the Italian Chapter

T. Calamoneri (Sapienza University of Rome)

ICTCS 2016

The 17th Italian Conference on Theoretical Computer Science (ICTCS 2016) was held at the University of Salento, Lecce - Italy, from the 7th to the 9th of September 2016.

ICTCS is the conference of the Italian Chapter of the European Association for Theoretical Computer Science and, besides being a forum of exchange of ideas, it provides the ideal environment where junior researchers and Ph.D. students meet senior researchers. ICTCS 2016 has been also open to researchers from outside Italy, who have been welcome to submit papers and attend the Conference.

This year we have had 48 participants from the Netherlands, Sweden, France, India and, of course, Italy.

The program has included the talks by the authors of 29 papers (divided in regular papers and short communications) and by 3 invited speakers (Giampaolo Bella, Gianlorenzo D'Angelo and Gianluigi Greco).

During the General Assembly of the Italian Chapter of EATCS, the University of Neaples has been presented as host of the next edition of ICTCS, that will be held in conjunction with CILC 2017: we welcome all of you to ICTCS 2017 in Neaples!

IC-EATCS awards

As usual, during the conference, the President of the Italian Chapter gave up some awards.

Every year the Italian Chapter assigns an award for the best Italian PhD thesis in Theoretical Computer Science. For 2016, the Selection Committee, composed by professors Paola Bonizzoni (Univ. of Milan), Michele Flammini (Univ. of l'Aquila) and Marinella Sciortino (Univ. of Palermo) selected as recipients of the award

Ilario Bonacina (whose thesis is entitled *Space in Weak Proof Systems*)

Furthermore, the award for the best young Italian researcher in Theoretical Computer Science has been established. The Selection Committee, composed by professors Giuseppe Di Battista (3rd Univ. of Rome), Simone Faro (Univ. of Catania) and Dino Mandrioli (Polytechnic of Milan) selected as recipient of this award

Gianlorenzo D'Angelo (Gran Sasso Science Institute – GSSI).

These three young researchers gave a talk to the ICTCS attendees describing their research area and summarizing their own results. In particular, a summary of Gialorenzo's research can be found in this volume of BEATCS.

Moreover, we have a further award for the best Master Thesis in Theoretical Computer Science. This year Massimo Cairo (Univ. of Pisa) has been selected by the Council members of the Italian Chapter of the EATCS as the recipient of this award. The title of his thesis is *New Bounds for Approximating Extremal Distances in Undirected Graphs* and the results contained in it appeared in the Proceedings of SODA 2016.

The Italian Chapter

REPORT ON 1ST GSSI SUMMER MEETING ON ALGORITHMS

A report by M. Flammini and G. Persiano

The first GSSI Summer Meeting on Algorithms was held at the Gran Sasso Science Institute in L'Aquila, Italy, on July, 9th, 2016. The Gran Sasso Science Institute (GSSI) is a new international research center and PhD school. The GSSI has been recently funded with the objective to create a new center of scientific excellence in L'Aquila fostering the skills and highly specialized structures already present in the area, such as the Gran Sasso National Laboratories of the National Institute for Nuclear Physics (INFN) and the University of L'Aquila.

The aim of the meeting was to bring together top researchers in the area of algorithms to have presentations of recent results and informal discussions. The morning started with Paul Spirakis (U. Liverpool) that gave a presentation entitled "The Complexity of Greedy Matchings." The study of greedy matchings is motivated by the fact that in several cases a matching in a graph is stable if and only if it is produced by a greedy algorithm. A greed matching algorithm considers edges by decreasing weight and choices are to be made when edges have equal weight. In wide contrast to the maximum weight matching problem, for which many efficient algorithms are known, the talk showed that GreedyMatching is strongly NP-hard and APX-complete, and thus it does not admit a PTAS unless P=NP, even on simple graphs.

Moti Yung (Columbia U. and Snapchat Inc.) gave a perspective on the difficulties one encounters in the deployment of advance cryptographic protocols in a commercial environment. Specifically, the reasons for the inherent difficulty of developing secure multi-party protocols for achieving actual business goals have been discussed. Secure computation protocols were invented as a basic theoretical notion, capturing specific and then general computational tasks, about 40 years ago and in spite of its theoretical and even more recent commendable experimentation success, the notion has not yet been widely and seriously used in achieving routine relevant business goals in contrast with symmetric key and public key cryptosystems and protocols, which were also proposed a little more than 40 years ago and are used extensively, primarily to implement secure authenticated channels.

Yuval Rabani (Hebrew U.) presented results regarding the convergence of Fisher Markets with constant elasticities of substitution (CES) utilities with respect to a limited rationality dynamics. Specifically, the talk considered a "control theoretic" approach to the dynamics of economic exchange, based on limited lookahead situational analysis of the participating agents. It is motivated by and generalizes the level k model in which a level 0 player adopts a very simple response to current conditions, a level 1 player best-responds to a model in which others take level 0 actions, and so forth. The main result shows the dynamics a linear rate of convergence.

In the afternoon, we started with a talk by Pierre Fraigniaud (U. Paris-Diderot) that discussed property testing in the context of distributed computing. It is known that testing whether a graph is triangle-free can be done in a constant number of rounds, where the constant depends on how far the input graph is from being triangle-free. This result is extended to H-freeness, for every connected 4-node graph H. Quite surprisingly, testing K_k-freeness and C_k-freeness for $k \geq 5$ appears to be more difficult as natural algorithms (the DFS and the BSF testers) require more than a constant number fo rounds.

The last talk of the seminar was given by Seffi Naor (The Technion) oon the metric multi-labeling problem, that is motivated by applications in multi-label learning. The metric multilabeling problem is NP-hard, and the talk tackles it by formulating an integer program capturing the deviation from a benchmark representing an "ideal" labeling. This approach leads a tight 2-approximation algorithm for metric multi-labeling by using a counterintuitive approach that distorts the optimal likelihood values computed by the linear programming relaxation.

Report on UCNC 2016
the 15th International Conference on Unconventional Computation and Natural Computation

Susan Stepney

The 15th International Conference on Unconventional Computation and Natural Computation (UC 2016) took place at the Manchester Metropolitan University, UK, 11–15 July 2016. Manchester is birthplace of the industrial revolution, home to Alan Turing and the first ever stored-program computer, and the driving force behind graphene. The conference was organised by the interdisciplinary Informatics Research Centre, and was held at the University's Business School. The conference received support from Manchester Metropolitan University, and Springer.

As always, the fully international complement of authors (submitted and accepted papers), and delegates came from across the globe, this year from: Austria, Brazil, Canada, China, France, Germany, India, Japan, Republic of Korea, Kuwait, Lebanon, Mexico, Republic of Moldova, Norway, Paraguay, Poland, Singapore, UK, and USA (figure 1).

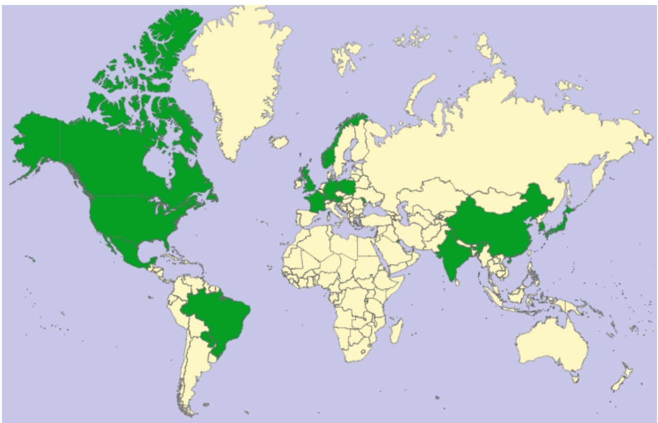


Figure 1: International participation in UCNC2016. (Map produced using www.amcharts.com/visited_countries/)

The three invited keynote speakers and their talks were

- Friedrich Simmel (Professor of Experimental Physics, Technische Universität München, Germany) "Chemical Communication Between Cell-Sized Reaction Compartments".

This was a fascinating account about a series of experiments sending signals between cells, droplets, and "genelets" (droplets containing cellular "naked" genetic machinery), based on the ideas of quorum sensing: when a high enough chemical signal concentrations is produced, because there are enough producers around, it invokes a response. We saw droplets signalling the chemicals, inducing bacteria to react, and that signal propagating through multiple droplets. There is a "bacterial Turing test": can you make a droplet that a bacterium will interact with (through chemical signals) just as if it were another bacterium? These systems pass it. Through a clever use of microfluidics, we saw videos of sheets of bacteria interacting, via fluorescent protein production. The fluorescence increases both due to the being switched on by the signalling, and due to the bacteria reproducing, two process with similar timescales. The possibilities of this approach include forming spatial and temporal patterns through reaction-diffusion systems of interacting genetically programmed droplets. Simmel then finished his talk with a description of using electron lithography to etch chips, deposit gene-length strands of DNA in a controlled manner, which could then be manipulated to stick together (condense) into linear bundles. It's early days yet; next on the agenda is using gene expression to control the condensation.

- Bob Coecke (Professor of Quantum Foundations, Logics and Structures, Department of Computer Science, University of Oxford, UK) "In Pictures: From Quantum Foundations to Natural Language Processing".

Coecke introduced us to a beautiful, formal, diagrammatic notation for quantum systems, and how the power of this notation makes many complicated quantum puzzles and proofs essentially vanish. There will soon be a book, *Picturing Quantum Processes*, from Cambridge University Press, covering this. It is 922 pages long, because pictures take a lot of space. After all this the quantum mechanics, Coecke went off in an unexpected direction, by showing how the very same notation could be used to calculate the meaning of sentences from their underlying grammar and the meaning of the individual words. Some modern meaning systems use high dimensional vectors to encapsulate word meanings. Adding the grammar via the diagrams improves the calculated meaning enormously. Then thinking about the mathematical structures needed leads to the suggestion of using density matrices rather than vectors, to cope with ambiguous meanings. This is a nice example of a deep piece of work in one domain that is not only applicable in a seemingly unrelated domain, but that suggests advances there, too.

- Steve Furber (ICL Professor of Computer Engineering, School of Computer Science, University of Manchester, UK) "The SpiNNaker Project".

After some interesting historical context, Furber told us of the SpiNNaker machine: one million processors in an asynchronous spiking architecture (SpiNNaker stands for "Spiking Neural Network Architecture"). The preliminary machine, with 500,000 cores, was launched 30 March 2016, and more cores have been added since. It can be programmed in the Python PyNN language. For example, 165 lines of Python are needed for a Sudoku solver, where the neuronal groups inhibit other groups with the same integer value in the the same row, column, or 3×3 cell. Once a solution has been found, the inhibitory links decrease, and the spiking rate goes up, solving a "diabolical" puzzle in about 10 seconds. This isn't just a toy: it is representative of complex constraint problems. So far people have only been running small programs, as they think how to scale up their ideas. Although each core is a standard processor, exploiting the asynchronous spiking communication requires a different way of thinking.

There were also three invited tutorials:

- "Many Hands Make Light Work: A Case Study in Swarm Robotics", by Jon Timmis (Professor of Intelligent and Adaptive Systems, Department of Electronics, University of York, UK), on XXX.

This subject has multiple simple automomous robots working together with no global control, to produce an emergent behaviour and capability that none has individually. The tutorial covered the history of the subject, showing how some of the original constraints have become irrelevant: today's "simple" robots are actually quite sophisticated compared to those at the discipline's inception; and the original "nature inspiration" is no longer so prominent: use it if it helps, ignore it if it doesn't. There are a couple of issues that make the subject difficult. The first is, how to design the local, individual robot rules that produce the desired emergent behaviour (and doesn't produce undesired behaviours also)? This often reduces to an iterative design: suggest, test, refine, which can be automated in a search algorithm, such as an evolutionary search. This leads to the second issue: this search is most efficiently done in simulation, but there is a "reality gap" in simulation: the simulated physics is often too simplistic, leading to "overfitting" to the simulation and the solution then not working on the embodied physical robots. There are lots of fascinating results addressing these issues: the next challenge is moving this research out of the lab into the real world.

- "Gellular Automata", Masami Hagiya (Professor, Department of Computer Science, University of Tokyo, Japan).

Gellular Automata are a form of cellular automata implemented using gels and chemical reactions. The walls between cells can be "decomposed" or "composed" using chemical reactions, or instead can "swell" or "unswell", forming a valve. This allows chemicals to move between cells. There are theoretical results demonstrating these systems can in principle implement certain kinds of CAs. The tutorial moved on to talking about implementations. Most of the manipulations involve a form of DNA chemical computing: using complementary strands to form networks of polymers, or to control diffusion by attaching anchors. These processes can be controlled by the DNA technique of "strand displacement" that breaks the bonds between the complementary strands. There are some initial prototype implementations. These are still rather complicated, needing multiple chemical species to implement relatively simple state transitions. However, it is early days yet, and more efficient approaches may well be discovered.

- "Self-Assembling Adaptive Structures with DNA", Rebecca Schulman (Assistant Professor of Chemical and Biomolecular Engineering and Computer Science, Johns Hopkins University, USA).

Schulman's philosophy is, rather than trying to assemble arbitrary structures, let's just look at what can be done with 1D systems: filaments of DNA nanotubes than can controllably be built into strings, trees, and network structures. She pointed out that it doesn't make sense to build every structure from weaving pure DNA: a human-size object would need about 3 light years of it. But smaller things can sensibly be built this way. This approach doesn't include only static structures: movement can be achieved by growing at the front and dissolving at the back. This is the way the cytoskeleton in cells works to move them around. DNA nanotube growth can be controlled by a variety of chemical processes, but it's hard to design different systems: there's no good enough model or simulation of how it all works. Currently things are a mixture of approximate yet expensive simulations, and lab experiments. But this is clearly a very powerful and rich area.

The full conference comprised these keynotes and tutorials, together with the scientific programme of technical presentations of the published papers, and a poster session.

Proceedings of UCNC 2016 are published in the Springer series as LNCS volume 9726 (ISBN 978-3-319-41312-9). The volume contains abstracts from the six keynote speakers and tutorial presenters, and 15 refereed contributed papers.

There were also two workshops on related unconventional topics run in association with the main technical conference:

- Workshop on Membrane Computing (WMC 2016)
- The 7th International Workshop on Physics and Computation; electronic proceedings available at arxiv.org/html/1606.06513v1

There were some particular highlights of the conference for me, in addition to the excellent keynotes and tutorials. Ella Gale talked on "Analysis of Boolean Logic Gates Logical Complexity for use with Spiking Memristor Gates", demonstrating that analysing the gates natural to memristor systems leads to a ternary logic formulation. Raul Rojas talked on "Babbage meets Zuse: a Minimal Mechanical Computer", starting with a description of Zuse's mechanical computer, and gradually paring down the system to produce a universal computer comprising three cog wheels and one gate. Gilles Dowek, an invited speaker at the Physics and Computation workshop, formalised a simple concept across Newtonian mechanics, Special Relativity, and General Relativity, in a cellular automaton. On the way he introduced a particular set of units familiar to astrophysicists, setting $c = G = 1$; in these units Planck's constant has dimensions of area, with a value closely related to the area of one bit in the Bekenstein bound.

These examples just help illustrate the broad interdisciplinary diversity of material covered by Unconventional Computation and Natural Computation. Further information can be found at the conference website: www.ucnc2016.org

Many thanks for another very well organised event go to: Martyn Amos and Anne Condon (Co-chairs), James Charnock, Matthew Crossley, René Doursat, and Emma Norling (organising committee).

Next year's UCNC moves west: 5–9 June 2017 in Fayetteville, Arkansas, USA. UCNC 2018 will be in Fontainebleu, France.

Miscellaneous

Foreword

Luca Aceto
ICE-TCS, School of Computer Science
Reykjavik University

It is fair to say that not many computer scientists try to present innovative research findings in a way that is accessible to an interested, but rather unspecialized, public. Even fewer succeed and the rewards for those who do are relatively minor. As a consequence, the number of essays and books about computer science that have a wide readership is substantially smaller than those about astronomy and physics, say. In my humble opinion, this is a pity, since many of intellectual achievements of computer science research deserve to be known by any intellectually curious layperson.

I was therefore happy to learn about the Klaus Tschira Award for Achievements in Public Understanding of Science. Since 2006, the Klaus Tschira Stiftung has looked for young scientists who can write a generally understandable article (8,000 to 9,000 words) in German about their research and the content of their PhD thesis. The prize is awarded in each of biology, chemistry, information technology, mathematics, neurosciences and physics as well as in closely related fields. The contributions are judged by a panel of experts on science and communication, which selects the winners based on scientific quality and on how well the scientific contribution is presented in a way that is amenable to public understanding. Yearly, up to six winners receive the award, which is endowed with prize money of $5,000$ Euros. The prize-winning contributions are published in a supplementary issue of the popular science magazine *bild der wissenschaft* (German). Moreover, all competitors are offered a participation in a two-day workshop for science communication.

The piece by Ágnes Cseh[1] you are about to read is the English translation of the German original that was selected as one of the prize-winning contributions for 2016. It is based on Ágnes' PhD thesis *Complexity and algorithms in matching problems under preferences*[2], which she defended in 2015 under the supervision of Martin Skutella at TU Berlin. I am sure that you will enjoy reading it as much as I did, regardless of whether you believe that algorithms can help us find stable marriages in real life.

[1] `https://sites.google.com/site/csehagnes88/`
[2] `http://dx.doi.org/10.14279/depositonce-5076`

Marriages are made in calculation

Ágnes Cseh
ICE-TCS, School of Computer Science
Reykjavik University

Abstract

Butterflies in the stomach, racing heart and sweaty hands... is he really the one? The mathematician simply sits down to the computer and calculates it quickly. On the side she makes important discoveries that come handy in various other fields of life, such as job applications.

In our everyday life we are surrounded by mathematics. It plays a role in planning logistics, scheduling buses, organizing soccer tournaments, establishing evacuation routes, only to name a few. But is it possible to model 'partner finding' as well with the help of numbers and forms? In my PhD thesis, I studied the 'stable marriage problem', which has been in the spotlight of research for over 50 years. In 2012, Alvin Roth and Lloyd Shapley were awarded the Nobel Memorial Prize in Economic Sciences for their achievements in the field. What is this mathematical model that is so often used in the industry and can be illustrated through romantic bonds?

Imagine a group of men and a group of women, all of whom are assumed to be heterosexual. Every person draws up a list. The first on this list is his or her true love, the second is their second best match and so on. Our goal now is to match these people so that their marriages stand the test of time. Instability is induced by a man and a woman who are not married to each other, yet they prefer each other to their respective partners or to being single. In a stable matching no such pair occurs.

David Gale and Lloyd Shapley proved that such a stable matching always exists, moreover, it can be found via a simple procedure. This procedure –also often referred to as 'algorithm' by us mathematicians– works through a series of old-fashioned proposals. Each man asks out the woman of his first-choice. Every woman who received at least one request takes the best one very cautiously only temporarily and she rejects the rest of them. Thus we have some couples now, but they are not married yet. In the second round, each single man asks out the best woman on his list who has not rejected him yet. Then the women contemplate whether they stay with their current partner or accept one of the new offers. If a woman decides to take a new offer, then her previous partner becomes single again. The algorithm runs in such a manner until every man is either in a relationship or he has been rejected by every woman on his list. The matching derived by this procedure is not only stable, but it is also the best stable matching for all men — and at the same time, it is the worst stable matching for all women. The reason for this is that while men were expressing their preferences, the women were forced to sit quietly and wait for offers. The takeaway message is thus: the more active you are during dating, the better partner you receive. And this advice is not coming from a motivational self-help book, but it is proved with mathematical rigor.

We all know that feelings cannot be modeled as ordered lists, moreover, probably nobody would like to automatize dating. Thus, stable matchings serve rather as a metaphor for the main concept of stability, and the statements in the topic marriage are not to be taken literally. However, the concept has been used for over 60 years in assigning medical residents to hospitals. In this application, the job openings play the role of women, while the prospective residents are the men. The goal is to find an assignment in which every resident is matched to the best hospital that could not fill its positions with better applicants. The wide range of possible applications contains admission decision at colleges and in schools, schedules at sports events, living donor kidney exchange programs, online auctions and distribution of dormitory places.

It is easy to see on these applications that the efficiency of the matching algorithm is absolutely necessary: it must run very fast even if there are a huge number of participants. For example, the residency matching program in the U.S. assigns over 40000 residents to hospitals each year. Besides, the algorithm must be adjusted to various extensions of the basic problem.

Many of these extensions are discussed in my thesis with one of them being the case of changing preferences. The above described algorithm is based on the assumption that no new person arrives to the scheme and that every participant sticks to their original preference list. Reality proves to be different though, since there will always be a resident who changes their preferences in the last minute. The smallest change in the allocation can induce a huge chaos in the system, in which one resident reassignment is followed by another one. When and how does this avalanche come to a halt? I have shown for complex and realistic systems –for example when hospitals advertise several open positions– that after altering their preference lists the participants can soon find a new stable matching on their own. The main point here is that the participants stabilize the matching without the interaction of any outside authority. This principle can be carried over to other problems as well, such as player transfer in soccer championships or allocating roommates in dormitories.

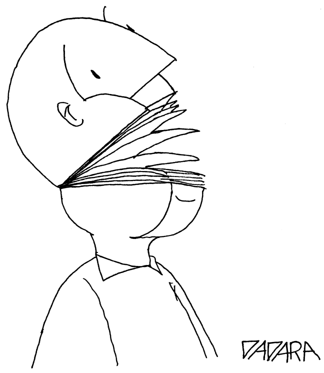
During a longer research stay in India I noticed that the Western view on arranged marriages is quite narrow. Several people who underwent the process, testified that arranged marriages are often success stories. As a researcher of stable matchings, I had to take a closer look at the problem, of course, from a scientific point of view. The original model can easily be extended to capture the wish of parents as well. The young men and women form the previously described system. Naturally, they all have their own preferences, but their parents' wishes might differ from this as day and night. Strict parents prescribe a forced partner, while the lenient ones only forbid some potential relationships. The goal is to find a matching that is stable with respect to the youngsters preferences, moreover it contains all forced marriages while it avoids all forbidden ones. In this way all parents and their children will be made content and on the top of this, there is no inclination for an affair among the youngsters. A forced marriage translates into the situation in the residency placement application where a hospital director is determined to hire a resident even if he or she belongs to the weaker candidates.

It is easy to see that the wishes of the parents can undermine the stability of the marriages. Any marriage scheme will be destabilized by a forbidden relationship in which the man and the woman both are each other's dream partners. Thus the next question rises naturally: How can one find a matching that is approved of by all parents and it has the fewest possible number of potential affairs? It turns out that computing such a matching is at least as hard as the solving a so-called NP-complete problem. These problems have been well studied by mathematicians and computer scientists for decades, and despite of a considerable amount of effort on their side, nobody has managed to come up with a fast algorithm for any NP-complete problem. A one-million-dollar prize is offered as a reward for the person who fist cracks the nut and answers the question whether such a fast algorithm can exist at all. It is therefore very probable that a good set of marriages under forced and forbidden pairs cannot be computed in reasonable time, even with the fastest computers of the world. This of course also holds in all applications: manipulation in the form of forced and forbidden pairs make the problem significantly hard to solve. What could frighten away backward-thinking parents from matchmaking if not this result?

The existing allocation models are not sufficient for several highly complex problems. For example, supply chains of large firms have a flow of goods through numerous vendors. Each of these vendors can specify their own preferences over their possible deals. I extended the above described proposal-acceptance algorithm for such networks. Moreover, I studied the case of market manipulation in the form of forced and forbidden deals. These results opened doors to novel areas of applications where stability has not been used yet.

The models developed by me can efficiently handle preference changes and they capture complex networks. Furthermore, I showed that manipulation in the form of forced and forbidden pairs is next to impossible to implement. The main strength of these results is the fact that they can be universally used in all fields of applications. Additionally, I also proved that a breakup is not the end of the world and that arranged marriage is rather a bad idea. In the end, one receives the same advice from mathematics: The right thing to do is to listen to the heart. Is there a soft kernel hiding in the ice cold mathematics, if one takes a closer look?

Abstracts of PhD thesis



Abstract of an Award-winning PhD Thesis

Luca Aceto
ICE-TCS, School of Computer Science
Reykjavik University

Every year, the Italian Chapter of the EATCS gives an award for the Best Italian PhD Thesis in Theoretical Computer Science. The award is presented at the annual Italian Conference on Theoretical Computer Science (ICTCS), where the award recipient delivers a presentation on her/his work.

This year's award went to Ilario Bonacina for his thesis *Space in weak propositional proof systems*, which was supervised by Nicola Galesi at the University of Rome "La Sapienza". Ilario's thesis contributes to a classic and deep topic in theoretical computer science, and settles natural questions on the space complexity of proofs using Resolution and the Polynomial Calculus that had been open for about 15 years.

Ilario kindly agreed to contribute a summary of the work presented in his thesis to this issue of the Bulletin of the EATCS. I trust that his survey will be of interest to readers of the Bulletin, regardless of their main research interests. Enjoy it!

Abstract of PhD Thesis

Author:	Ilario Bonacina
Title:	Space in weak propositional proof systems
Language:	English
Supervisor:	Nicola Galesi
Institute:	Sapienza University of Rome, Italy
Date:	14 December 2015

Abstract

This thesis was defended on December 14, 2015 at the Sapienza University of Rome for a Ph.D. title in Computer Science under the supervision of Prof. Nicola Galesi. It was awarded "*Best Italian PhD Thesis in Theoretical Computer Science, 2016*". The results presented in this thesis build on top of the following publications [9, 12–16].

1 Preliminaries

Propositional proof complexity, that is the complexity of propositional proofs, plays a role in the context of feasible proofs as important as the role of Boolean circuits in the context of efficient computations. Although the original motivations to study the complexity of propositional proofs came from proof-theoretical questions about first-order theories, it turns out that, essentially, the complexity of propositional proofs deals with the following question: *what can be proved by a prover with bounded computational abilities?* For instance if its computational abilities are limited to small circuits from some circuit class. Hence, propositional proof complexity mirrors to non-uniform computational complexity and indeed there is a very productive cross-fertilization of techniques between the two fields. Our understanding of propositional proof systems is similar to the general situation in complexity theory, in the sense that in both fields we can prove lower bounds in very special cases and indeed there are many very basic and important open problems, such as the very famous P *vs* NP. In propositional proof complexity the situation is similar in the sense that we can prove super-polynomial lower bounds on the length of proofs only for restricted proof systems. Indeed proving super-polynomial lower bounds on the length of proofs in *every* propositional proof system is equivalent to showing that $\mathsf{NP} \neq \mathsf{coNP}$ [21], which in turn is one of the open and very important problems in computational complexity.

In this thesis we investigate space complexity in propositional proof systems, so what is the *space*[1] of a proof? Intuitively, the space required by a refutation is the amount of information we need to keep simultaneously in memory as we work through the proof and convince ourselves that the original propositional formula is unsatisfiable. This model is inspired by the definition of space complexity for Turing machines, where a machine is given a read-only input tape from which it can download parts of the input to the working memory as needed. This model is sometimes called in the literature *blackboard model* and the name comes from the image of a teacher in front of a class of students. The goal of the teacher is to show that a propositional formula is contradictory[2] writing down clauses and performing inferences on a blackboard. In this analogy students understand inferences based on the rules of some particular proof system, for example (among others) Frege; or *Resolution*, a well studied proof system that is at the core of state-of-the-art algorithms to solve SAT instances (Res); or *Polynomial Calculus* (PC), a proof system that uses polynomials to refute contradictions. As for length of proofs, the study of space complexity for proof systems represents a great theoretical challenge and may also have practical consequences on techniques for SAT solving and their implementation.

We completely answer questions on the space complexity for Resolution and Polynomial Calculus raised for the first time in [2, 6] and since then reported many times in the literature. The results we show can be summarized as follows.

Monomial space in Polynomial Calculus We introduce a combinatorial framework to prove monomial space lower bounds. This framework belong to the class of game theoretic methods and combinatorial characterizations that are widely used in proof complexity to study complexity measures[3]. As an application we then have asymptotically optimal lower bounds on the monomial space needed to refute random k-CNF formulas (and the graph pigeonhole principle) or *Tseitin formulas* in Polynomial Calculus. Those results were conjectured to be true and posed as open problems in many works, [2, 6, 24] among others. The framework is described on a very high level in Section 2.1 of this abstract, the results about random k-CNFs in Section 3 and the ones about Tseitin formulas in Section 4.

Total space in Resolution We give another combinatorial framework to prove total space lower bounds which results in a tight connection between the total space measure and the width. Then, as corollaries, we have asymptotically optimal total

[1]The problem of the *space* taken by propositional proofs was posed for the first time by Armin Haken during the workshop "*Complexity Lower Bounds*" held at Fields Institute in Toronto 1998.

[2]In this abstract and the thesis *proofs* will be always *refutations* of contradictions. So we use the two terms interchangeably.

[3]Some examples are the Pudlák games characterizing the *size* of Resolution proofs [35] or the families of assignments characterizing Resolution *width* [3], where the width of a proof is the number of literals in the largest clause appearing in it.

lower bounds in Resolution for *Tseitin formulas* over d-regular expander graphs, completely answering open problem from [2, Open question 2] for Resolution and we prove asymptotically optimal total space lower bound in Resolution for random k-CNF formulas, completely answering an open problems from [2, 6, 25] among others. Moreover it follows an optimal separation of Resolution and *semantic* Resolution from the point of view of the total space measure, completely answering [2, Open question 4] for Resolution. The framework is described more in details in Section 2.2 of this abstract, the results for random k-CNFs in Section 3 and the ones for Tseitin formulas in Section 4.

Size and width in Resolution Together with the main results about space this thesis contains also a detour on size, cf. Section 5 of this abstract. Indeed, using the game theoretic characterization of width and size in Resolution, we are able to prove that the Strong Exponential Time Hypothesis (SETH) is consistent with a sub-system of Resolution, that is no algorithm with track formalizable in such system is able to refute SETH.

In this abstract, the numbering of theorems, corollaries, lemmas and propositions refer to their numbering in the thesis.

1.1 Resolution

Resolution (Res) [11, 38] is a sound and complete propositional proof system manipulating unsatisfiable CNF formulas. A formula is *Conjunctive Normal Form* (CNF) is a conjunction ($\wedge$) of clauses, where each clause is a disjunction ($\vee$) of *literals* and each literal is either a variable x_j or a negation of a variable $\neg x_i$. If each clause has at most k literals then it is a k-CNF formula. A Res refutation of a CNF formula φ is a sequence of clauses ending with the empty clause $\perp$ and such that each clause is either a clause from φ or can be inferred from previous clauses by the following inference rule:

$$\frac{C \vee x \qquad D \vee \neg x}{C \vee D} \text{ (Res rule)},$$

where C, D denote clauses and x is a variable that we say is *resolved*. A CNF formula φ is unsatisfiable if and only if the empty clause, $\perp$, can be inferred from φ using the Res rule.

To understand the complexity of Resolution proofs various hardness measures were defined and investigated. Historically, the first and most studied is the *size*: the number of clauses in a Resolution refutation π is its *size*, $\mathsf{size}(\pi)$. The *width* of a Resolution proof π, $\mathsf{width}(\pi)$, is the number of literals in the biggest clause appearing in π.

Given any unsatisfiable k-CNF formula φ in n variables, if there exists a Resolution refutation π of φ such that $\mathsf{size}(\pi) \leqslant S$ then there exists a Resolution proof

π' of φ such that

$$\mathsf{width}(\pi') \leqslant \sqrt{n \cdot O(\log S)} + k, \tag{1}$$

so if for every Resolution proof π of φ, $\mathsf{width}(\pi) \geqslant \omega(\sqrt{n \log n})$ and φ has $n^{O(1)}$ clauses then immediately we have that φ must require Resolution refutations of super-polynomial size. This is known as the "*size-width tradeoff*" [8] and it is optimal up to logarithmic factors [17]. Equation (1) is the standard tool to prove exponential size lower bounds, but in some cases it is not enough. In this thesis we prove some results on Resolution size stronger than the size lower bound we could get by the technique presented above.

Nowadays Resolution is mostly studied due to its importance in applied contexts due to a connection to the *CDCL solvers*, which are at the core of modern SAT-solvers [33]. In particular, lower bounds on Resolution size and Resolution *space* (cf. Section 2) imply lower bounds on the running time and the memory consumption of CDCL solvers.

1.2 Polynomial Calculus

In *Polynomial Calculus*, PC, [2, 20] an unsatisfiable CNF formula φ in the variables $x_1, \dots, x_n$ is shown to be unsatisfiable first translating it into a set of multilinear monomials $tr(\varphi)$ such that φ is unsatisfiable if and only if 1 is in the ideal generated by $tr(\varphi)$ ($1 \in \mathrm{ideal}(tr(\varphi))$) in the ring of polynomials $\mathbb{F}[x_1, \dots, x_n, \bar{x}_1, \dots \bar{x}_n]$ where the $\bar{x}_i$ variables are new variables and $\mathbb{F}$ is a field[4]. Then, to show that $1 \in \mathrm{ideal}(tr(\varphi))$ we use the following inference rules starting from the monomials in $tr(\varphi)$

$$\frac{p \quad q}{\alpha p + \beta q}\ \alpha, \beta \in \mathbb{F}, \qquad \frac{p}{qp}\ q \in \mathbb{F}[x_1, \dots, x_n, \bar{x}_1, \dots \bar{x}_n], \qquad \frac{}{x_i^2 - x_i}, \qquad \frac{}{x_i + \bar{x}_i - 1}.$$

These rules model the fact that ideals are closed under linear combinations and multiplications of generic polynomials. Moreover, they force the semantic meaning of the variables to be just Boolean variables and such that $\bar{x}_i = 1 - x_i$. In PC the polynomials are expressed in their expanded form as a sum of monomials, and the size of a PC proof π, $\mathsf{size}(\pi)$, is measured as the total number of monomials appearing in it[5]. As in Resolution, there are unsatisfiable formulas requiring exponentially long PC proofs and there exists a "*size-degree tradeoff*" [20], where the *degree* of a PC proof π, $\mathsf{degree}(\pi)$, is the maximum degree of a polynomial appearing in π. Given a k-CNF formula φ, if there exists a PC proof of φ such that

[4]For sake of clarity we avoid here the details of the translation $tr(\varphi)$.

[5]There are also algebraic proof systems that allow manipulations on polynomials in implicit forms and this results in stronger, not so well understood, proof systems [18, 19, 28–30, 34, 37].

$\mathsf{size}(\pi) \leqslant S$ then there exists a PC proof π' of φ such that

$$\mathsf{degree}(\pi') \leqslant \sqrt{n \cdot O(\log S)} + k. \tag{2}$$

Hence, if for every PC proof π of φ we have that $\mathsf{degree}(\pi) \geqslant \omega(\sqrt{n \log n})$ and φ has $n^{O(1)}$ clauses then φ cannot have polynomial size PC proofs. Moreover, if $\mathrm{char}\,\mathbb{F} \neq 2$ then some Fourier-like transformation can be used to reduce degree lower bounds to Resolution [7]. More general techniques to prove degree lower bounds, working also if $\mathrm{char}\,\mathbb{F} = 2$, were introduced in [1] and generalized in [26, 32]. It is interesting to notice the similarity between equation (2) and equation (1). Indeed, lot of results on the complexity of Resolution proofs are qualitatively similar to results on the complexity of PC proofs. As for Resolution, the size-degree relationship is essentially optimal [27] and most of the super-polynomial or exponential size lower bounds for PC proofs are obtained through degree lower bounds.

Our motivation to study algebraic proof systems is that they are not at all as well understood as Resolution and this lack of knowledge from the theoretical point of view might be one of the reasons for not having efficient SAT solvers properly exploiting the potential of algebraic manipulations. Moreover, the study of algebraic proof systems could shed light on major open problems in propositional proof complexity such as proving super-polynomial size lower bounds for $AC_0[p]$-Frege a Frege system where only bounded-depth formulas over the Boolean connectives and a MOD_p connective are allowed [19, 20].

2 Space

The formal definition goes as follows [2, 23]: A Resolution refutation π of a CNF formula φ is a sequence of memory configurations $\pi = (\mathfrak{M}_0, \ldots, \mathfrak{M}_\ell)$ where each $\mathfrak{M}_i$ is a set of clauses, $\mathfrak{M}_0 = \emptyset$, $\bot \in \mathfrak{M}_\ell$ and for each $i \geqslant 1$, $\mathfrak{M}_i$ is obtained from $\mathfrak{M}_{i-1}$ applying one of the following rules

(Axiom Download) $\mathfrak{M}_i = \mathfrak{M}_{i-1} \cup \{C\}$, where C is a clause in φ;

(Erasure) $\mathfrak{M}_i \subseteq \mathfrak{M}_{i-1}$;

(Inference) $\mathfrak{M}_i = \mathfrak{M}_{i-1} \cup \{C\}$ where C is the result of the Resolution inference rule applied with premises in $\mathfrak{M}_{i-1}$.

Clearly this definition can be adapted to other proof systems, for instance for PC we will just have as memory configurations sets of polynomials and as inference rules the ones from PC.

As Alekhnovich et al. [2] pointed out, the very first question, when starting the investigation of space, is how to measure the memory content/blackboard size at any given moment in time for a specified propositional proof system. Recalling Krajíček [31], the most customary measures for the size complexity of propositional proofs are the bit size and the number of lines. Among the two the bit size is the most important and can be defined analogously also for space complexity. In the case of space we measure the total number of literals in memory, the *total space*, a measure logarithmically related to the bit-size of the memory. Given a Resolution proof π we denote with $\mathsf{TSpace}(\pi)$ the maximum number of literals appearing in a memory configuration in π.

The line complexity is not an adequate space measure as long as the language of the proof system is strong enough to handle unbounded fan-in $\wedge$ gates: in this case just $O(1)$ memory cells are sufficient as one of them can contain a big-$\wedge$ of all the formulas derived in previous steps. For Resolution, that is not closed under $\wedge$, the lines are just clauses and the clause space makes prefect sense. Indeed Esteban and Torán [23] proposed the study of such measure: given a Resolution proof π, the *clause space*[6], $\mathsf{CSpace}(\pi)$, is the maximum number of clauses appearing in a memory configuration in π. For every contradictory CNF formula in n variables φ there exists a Resolution refutation π of φ such that $\mathsf{CSpace}(\pi) \leqslant n+1$ and hence, clearly, also $\mathsf{TSpace}(\pi) \leqslant n(n+1)$ [23].

An analogue of clause space makes sense also for stronger proof systems, such as Polynomial Calculus, where we consider the number of distinct *monomials* appearing in memory configuration, and analogously as before we define the *monomial space* of a PC refutation π, $\mathsf{MSpace}(\pi)$. Since the Resolution inference rule can be simulated efficiently in PC, from the point of view of space, for every unsatisfiable CNF formula φ in n variables, there exists a PC refutation π of φ such that $\mathsf{MSpace}(\pi) \leqslant O(n)$ and $\mathsf{TSpace}(\pi) \leqslant O(n^2)$. Total space in PC is not yet well understood and the only total space lower bound for PC are the ones by Alekhnovich et al. [2] where this measure was originally introduced.

The second interesting property of space is that this measure is actually nontrivial for not too strong proof systems, indeed Alekhnovich et al. [2, Theorem 6.3] showed that any tautology in n variables has a proof in Frege with "*formula space*" $O(1)$ and total space linear in the number of variables. This fact justifies the study of space for "*weak*" proof systems where actually super-linear lower bounds on space could be achieved, although total space in Frege is still a meaningful complexity measure.

[6] As already noticed by [23], the clause space in Resolution is connected also to the pebbling game on the DAGs associated to Resolution derivations but we do not exploit this analogy.

2.1 Monomial space

We consider families of assignments, r-BG *families*, consisting of *many* partial truth assignments with a combinatorial structure we called *flippable products*. For such families we can define a notion of *rank* that turns out to be roughly the logarithm of the number of assignments in the family. The formal definition of r-BG families, too technical to be presented here, is one of the main innovations of this thesis, since it reduces space lower bounds in algebraic proof systems to a combinatorial property on families of Boolean assignments. The r-BG families resembly other combinatorial definitions used to prove lower bounds in Resolution: the definition of *k-dynamical satisfiability*[22]; the winning strategies characterizing width in Resolution[3]. An r-BG family of assignments for $tr(\varphi)$ is a family of collections of partial assignments such that each collection has rank at most r, none of the collections falsify the polynomials in $tr(\varphi)$ and they satisfy some additional combinatorial properties.

Theorem 3.6 (informal[7]). *Given an unsatisfiable CNF formula φ. If there exists a non-empty r-BG family of partial assignments for $tr(\varphi)$ then for every PC refutation π of φ, $\mathsf{MSpace}(\pi) \geqslant \frac{r}{4}$.*

All the monomial space lower bounds obtained using this theorem are not dependent on the characteristic of the ground field $\mathbb{F}$ used in PC. This result generalizes the techniques used in [2, 24] and indeed the main technical difficulty to prove Theorem 3.6 is to prove a generalization of [2, Lemma 4.14], the *Locality Lemma*. As corollaries of Theorem 3.6, we are able to re-obtain all the lower bounds on monomial space known from [2, 24] and to prove the first monomial space lower bound for random k-CNF formulas, for $k \geqslant 3$, cf. Section 3. Moreover, Filmus et al. [25] applied (a preliminary version of) Theorem 3.6 to *Tseitin formulas* over random 4-regular graphs, cf. Section 4.

2.2 Total space in Resolution

The main result here is a general technique to prove *total space* lower bounds in Resolution, cf. Theorem 2.5, and, as an application, the fact that in Resolution '*total space is lower bounded by the square of width*', cf. Corollary 2.11. Then, as corollaries, we immediately have total space lower bounds for various families of CNF formulas of interest. We postpone the discussion of the results on random k-CNF formulas to Section 3 and the results on Tseitin formulas to Section 4.

Our main theorem for total space in Resolution, Theorem 2.5, and Theorem 3.6 on monomial space have similar statements. Here, to get total space lower

[7]The result proven is actually stronger since it holds for *semantic* PC, but for simplicity we state it here just for PC.

bounds we use r-BK families of partial truth assignments introduced in [10] to characterize the *asymmetric width* in Resolution, a complexity measure similar to the width[8]. An r-BK family for φ is a collection of partial assignments not falsifying φ and such that some combinatorial extension property holds for assignments of domain bounded by r.

Theorem 2.5 (informal)**.** *Given an unsatisfiable CNF formula φ, if there exists a non-empty r-*BK *family of assignments for φ then any* Res *refutation of φ must pass through a memory configuration of at least $r/2$ clauses each at least of $r/2$ many literals. Hence, in particular any* Res *refutation of φ require total space $r^2/4$.*

Corollary 2.11 (informal)**.** *Let φ be an unsatisfiable k-CNF formula, if there exists a* Res *refutation π of φ such that $\mathsf{TSpace}(\pi) \leqslant T$ then there exists a* Res *refutation π' of φ such that $\mathsf{width}(\pi') \leqslant O(\sqrt{T}) + k$.*

There are many of such CNF formulas φ with the properties above, for example random k-CNFs (see next Section). Moreover it follows an optimal separation between Resolution and a *semantic* version of it from the point of view of the total space measure. In the thesis there are also some lower bounds on total space for semantic Resolution and for a bounded version of it.

3 Random k-CNFs

Let k a positive integer and Δ a positive real number, an (n, k, Δ)-*random CNF formula* φ is a k-CNF formula with n variables and Δn clauses picked uniformly at random from the set of all CNF formulas in the variables $\{x_1, \dots, x_n\}$ which consist of exactly Δn clauses, each clause containing exactly k literals and no variable appears twice in a clause. For large enough Δ (depending on k), with high probability, an (n, k, Δ)-random CNF formula is unsatisfiable and there exists a constant $\gamma > 0$ such that for each Res refutation π of φ, $\mathsf{width}(\pi) \geqslant \gamma n$ [8].

Theorem 4.36 (informal)**.** *Let $k \geqslant 3$ and $\Delta > 1$. If φ is a (n, k, Δ)-random CNF, then for large n, with high probability, (1) for every* Res *refutation π of φ, $\mathsf{TSpace}(\pi) \geqslant \Omega(n^2)$; and (2) for every* PC *refutation π of φ, $\mathsf{MSpace}(\pi) \geqslant \Omega(n)$.*

The total space lower bound completely answers an open problem on the total space complexity in Resolution of random k-CNF formulas from [2, 6, 25] among others. It follows immediately by Corollary 2.11 and it also shows an optimal separation between semantic Resolution and Resolution from the point of view of total space and thus completely answers [2, Open question 4] for Resolution.

[8] This characterization is similar to the characterization of width in [3].

The lower bound on monomial space was conjectured to be true and posed as an open problem in many works, for instance [2, 6, 24]. The proof of this result use Theorem 3.6 and essentially consists in constructing an $\Omega(n)$-BG family of partial assignments for φ[9]. This technical construction relies on some combinatorial games over bipartite graphs, the *Cover Games*, and to some variations of Hall's theorem to objects similar to matchings, V-matchings and VW-matchings.

4 Tseitin formulas

Tseitin formulas, $\mathsf{Tseitin}(G, \sigma)$, are essentially Boolean encodings of the fact that the total degree of any graph is an even number[10]. Tseitin formulas are one of the standard tools used in proof complexity to prove lower bounds and trade-offs, for example they have used to prove the very first super-polynomial lower bound for Resolution Tseitin [39], result improved then to an exponential lower bound in [40]; they have been investigated regarding the width [8], clause space [23] and regarding size-space trade-offs in both Res and PC [4]. Notice that Tseitin formulas have polynomial size refutations in PC over $\mathbb{F}_2$, essentially mimicking Gaussian elimination. In [8] it is proved that for every Resolution refutation π of $\mathsf{Tseitin}(G, \sigma)$,

$$\mathsf{width}(\pi \vdash \bot) \geqslant e(G), \tag{3}$$

where $e(G)$ is the *connectivity expansion* of $G = (V, E)$: for any set E' of at most $e(G)$ edges it holds that $G' = (V, E \setminus E')$ has a (unique) connected component of size strictly larger than $|V|/2$. If $e(G) = \Omega(|V|)$, which happens for example for random d-regular graphs (w.h.p.), then from equation (3) and (1) it follows an exponential lower bound on the size of Resolution refutations of Tseitin formulas. Then as an application of Corollary 2.11 we answer the open problem from [2, Open question 2] concerning total space lower bounds for Tseitin formulas in Resolution.

Theorem 4.7 (informal)**.** *Let $G = (V, E)$ be a connected d-regular graph and σ an odd-weight function over V, then for every Resolution proof π of* $\mathsf{Tseitin}(G, \sigma)$

$$\mathsf{TSpace}(\pi) \geqslant \Omega((e(G) - d)^2).$$

[9] An analogue result holds also for the *matching principle over a graph G*, G-PHP, where G is an expander bipartite graph with left degree at least 3, cf. Theorem 4.38.

[10] Formally the Tseitin formulas are defined as follows. Let $G = (V, E)$ be a finite connected graph of degree at most d over n vertices and $\sigma : V \to \{0, 1\}$ be such that $\sum_{v \in V} \sigma(v) \equiv 1 \pmod 2$. Consider now the set of Boolean variables $X = \{x_e : e \in E\}$ and for each $v \in V$ let $\mathsf{PARITY}_{v,\sigma}$ be the CNF formula expressing the following parity: $\sum_{e \ni v} x_e \equiv \sigma(v) \pmod 2$. The *Tseitin formula*, $\mathsf{Tseitin}(G, \sigma)$, is then $\bigwedge_{v \in V} \mathsf{PARITY}_{v,\sigma}$.

In particular if G is a 3-regular expander graph over n vertices then every Resolution refutation π is such that $\mathsf{TSpace}(\pi) = \Theta(n^2)$.

Regarding the monomial space in Polynomial Calculus the picture is more complex. We do not know non-trivial monomial space lower bound for Tseitin formulas over 3-regular expander graphs. Yet we have some monomial space lower bounds for some Tseitin formulas. In particular the following results showed by Filmus et al. [25] relying on a preliminary version of Theorem 3.6:

• If $G = (V, E)$ is a d-regular graph with edges with multiplicity 2, then for every PC refutation π of $\mathsf{Tseitin}(G, \sigma)$, $\mathsf{MSpace}(\pi) \geqslant \Omega(e(G) - d)$.

• If $G = (V, E)$ is a random d-regular graph on n vertices, where $d \geqslant 4$, then w.h.p. for each PC refutation of $\mathsf{Tseitin}(G, \sigma)$, $\mathsf{MSpace}(\pi) \geqslant \Omega(\sqrt{n})$.

5 Strong size lower bounds

Given a k-CNF formula in n variables φ, we call a Resolution size lower bound *strong*[11] if for every Resolution refutation π of φ,

$$\mathsf{size}(\pi) \geqslant 2^{(1-\epsilon_k)n},$$

where $\lim_{k\to\infty} \epsilon_k = 0$. Similarly a width lower bound is *strong*[12] if for every Resolution refutation π of φ $\mathsf{width}(\pi) \geqslant (1 - \epsilon_k)n$, where $\lim_{k\to\infty} \epsilon_k = 0$.

We show a strong size lower bound for a sub-system of Resolution where at most a fraction of δ variables can be resolved multiple times along any path in a refutation DAG of an unsatisfiable CNF formula. We called *δ-regular* Resolution such system in between unconstrained Resolution and *regular* Resolution, a variation of Resolution where are allowed as valid only the Resolution refutations that have a DAG structure where along any path no variable is resolved twice. Similarly we can define *tree-like* Resolution, a variation of Resolution where are allowed as valid only the Resolution refutation that have a tree-like structure. Before our result strong size lower bounds were known for tree-like Resolution [36] and for regular Resolution [5]. Our results both improve and simplify the strong size lower bound from Beck and Impagliazzo [5] and improve the asymptotic of the ϵ_k for tree-like and regular Resolution. More precisely we prove the following.

[11] Proving a strong exponential size lower bound for Resolution will mean that no SAT-solver purely based on Conflict Driven Clause Learning will be able to refute the *Strong Exponential Time Hypothesis*, due to the fact that such solvers are polynomially simulated by Resolution.

[12] It is always the case that strong width lower bounds in Resolution imply strong size lower bound in *tree-like* Resolution, due to the size-width tradeoff for tree-like Resolution [8]. This is not the case for general Resolution, since the best known general tradeoff between width and size, equation (1), has some constant loss.

Corollary 5.8 (informal)**.** *For any large enough n and $k \in \mathbb{N}$ there exists an unsatisfiable k-CNF formula ψ in n variables such that for every δ-regular Resolution refutation π of ψ* $\mathsf{size}(\pi) \geqslant 2^{(1-\epsilon_k)n}$, *where both ϵ_k and δ are $\widetilde{O}(k^{-1/4})$.*

The first ingredient to prove this result is a strong width lower bound.

Theorem 5.6 (informal)**.** *For any large n and k, there exist an unsatisfiable k-CNF formula φ on n variables such that for every Resolution refutation π of φ* $\mathsf{width}(\pi) \geqslant (1 - \zeta_k)n$, *where $\zeta_k = \widetilde{O}(k^{-1/3})$.*

Notice that the best possible would be $\zeta_k = O(k^{-1})$ since for every unsatisfiable k-CNF formula on n variables there exists a tree-like Resolution of size at most $2^{\left(1-\Omega(k^{-1})\right)n}$, cf. Theorem 5.2.

The second ingredient to prove Corollary 5.8 is an hardness amplification result, Theorem 5.5, proved using characterizations of Resolution size [35] and width [3] as games. Given a CNF formula φ in n variables, the *ℓ-xorification* of φ, $\varphi[\oplus^\ell]$, is a formula over ℓn new Boolean variables obtained by replacing each occurrence of x_i in φ with $y_i^1 \oplus \cdots \oplus y_i^\ell$ where y_i^j are fresh new variables.

Theorem 5.5 (informal)**.** *Let φ an unsatisfiable CNF formula in n variables and let W, δ and ℓ be parameters. If for every Resolution refutation π of φ,* $\mathsf{width}(\pi) \geqslant$ *W, then for every δ-regular Resolution refutation π' of $\varphi[\oplus^\ell]$,*

$$\mathsf{size}(\pi') \geqslant 2^{(1-\epsilon)W\ell},$$

where $\epsilon = \frac{1}{\ell}\log\left(\frac{e^3\ell n}{W}\right) + \frac{\delta n}{W}\log\left(\frac{e^3\ell}{\delta}\right)$.

References

[1] Michael Alekhnovich and Alexander A. Razborov. "Lower Bounds for Polynomial Calculus: Non-Binomial Case". In: *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*. IEEE Computer Society, 2001, pp. 190–199.

[2] Michael Alekhnovich et al. "Space Complexity in Propositional Calculus". In: *SIAM J. Comput.* 31.4 (2002), pp. 1184–1211.

[3] Albert Atserias and Víctor Dalmau. "A combinatorial characterization of resolution width". In: *J. Comput. Syst. Sci.* 74.3 (2008), pp. 323–334.

[4] Chris Beck, Jakob Nordström, and Bangsheng Tang. "Some trade-off results for polynomial calculus: extended abstract". In: *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*. Ed. by Dan Boneh, Tim Roughgarden, and Joan Feigenbaum. ACM, 2013, pp. 813–822.

[5] Christopher Beck and Russell Impagliazzo. "Strong ETH holds for regular resolution". In: *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*. Ed. by Dan Boneh, Tim Roughgarden, and Joan Feigenbaum. ACM, 2013, pp. 487–494.

[6] Eli Ben-Sasson. "Expansion in Proof Complexity". Hebrew University. PhD thesis. Hebrew University, 2001.

[7] Eli Ben-Sasson and Russell Impagliazzo. "Random CNF's are Hard for the Polynomial Calculus". In: *Computational Complexity* 19.4 (2010), pp. 501–519.

[8] Eli Ben-Sasson and Avi Wigderson. "Short proofs are narrow - resolution made simple". In: *J. ACM* 48.2 (2001), pp. 149–169.

[9] Patrick Bennett et al. "Space proof complexity for random 3-CNFs". In: *CoRR* abs/1503.01613 (2015).

[10] Olaf Beyersdorff and Oliver Kullmann. "Unified Characterisations of Resolution Hardness Measures". In: *Theory and Applications of Satisfiability Testing - SAT 2014 - 17th International Conference, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 14-17, 2014. Proceedings*. Ed. by Carsten Sinz and Uwe Egly. Vol. 8561. Lecture Notes in Computer Science. Springer, 2014, pp. 170–187.

[11] Archie Blake. "Canonical Expressions in Boolean Algebra". University of Chicago. PhD thesis. University of Chicago, 1937.

[12] Ilario Bonacina and Nicola Galesi. "A Framework for Space Complexity in Algebraic Proof Systems". In: *J. ACM* 62.3 (June 2015), 23:1–23:20.

[13] Ilario Bonacina and Nicola Galesi. "Pseudo-partitions, transversality and locality: a combinatorial characterization for the space measure in algebraic proof systems". In: *Innovations in Theoretical Computer Science, ITCS '13, Berkeley, CA, USA, January 9-12, 2013*. Ed. by Robert D. Kleinberg. ACM, 2013, pp. 455–472.

[14] Ilario Bonacina, Nicola Galesi, and Neil Thapen. "Total Space in Resolution". In: *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*. IEEE Computer Society, 2014, pp. 641–650.

[15] Ilario Bonacina and Navid Talebanfard. "Improving resolution width lower bounds for k-CNFs with applications to the Strong Exponential Time Hypothesis". In: *Information Processing Letters* 116.2 (2015), pp. 120 –124.

[16] Ilario Bonacina and Navid Talebanfard. "Strong ETH and Resolution via Games and the Multiplicity of Strategies". In: *10th International Symposium on Parameterized and Exact Computation (IPEC 2015)*. Ed. by Thore Husfeldt and Iyad Kanj. Vol. 43. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015, pp. 248–257.

[17] Maria Luisa Bonet and Nicola Galesi. "Optimality of size-width tradeoffs for resolution". In: *Computational Complexity* 10.4 (2001), pp. 261–276.

[18] Samuel R. Buss et al. "Linear Gaps Between Degrees for the Polynomial Calculus Modulo Distinct Primes". In: *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, May 1-4, 1999, Atlanta, Georgia, USA*. Ed. by Jeffrey Scott Vitter, Lawrence L. Larmore, and Frank Thomson Leighton. ACM, 1999, pp. 547–556.

[19] Samuel R. Buss et al. "Proof Complexity in Algebraic Systems and Bounded Depth Frege Systems with Modular Counting". In: *Computational Complexity* 6.3 (1997), pp. 256–298.

[20] Matthew Clegg, Jeff Edmonds, and Russell Impagliazzo. "Using the Groebner Basis Algorithm to Find Proofs of Unsatisfiability". In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*. Ed. by Gary L. Miller. ACM, 1996, pp. 174–183.

[21] Stephen A. Cook and Robert A. Reckhow. "The Relative Efficiency of Propositional Proof Systems". In: *J. Symb. Log.* 44.1 (1979), pp. 36–50.

[22] Juan Luis Esteban, Nicola Galesi, and Jochen Messner. "On the complexity of resolution with bounded conjunctions". In: *Theor. Comput. Sci.* 321.2-3 (2004), pp. 347–370.

[23] Juan Luis Esteban and Jacobo Torán. "Space Bounds for Resolution". In: *Inf. Comput.* 171.1 (2001), pp. 84–97.

[24] Yuval Filmus et al. "Space Complexity in Polynomial Calculus". In: *SIAM J. Comput.* 44.4 (2015), pp. 1119–1153.

[25] Yuval Filmus et al. "Towards an Understanding of Polynomial Calculus: New Separations and Lower Bounds - (Extended Abstract)". In: *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part I*. Ed. by Fedor V. Fomin et al. Vol. 7965. Lecture Notes in Computer Science. Springer, 2013, pp. 437–448.

[26] Nicola Galesi and Massimo Lauria. "On the Automatizability of Polynomial Calculus". In: *Theory Comput. Syst.* 47.2 (2010), pp. 491–506.

[27] Nicola Galesi and Massimo Lauria. "Optimality of size-degree tradeoffs for polynomial calculus". In: *ACM Trans. Comput. Log.* 12.1 (2010), p. 4.

[28] Dima Grigoriev and Edward A. Hirsch. "Algebraic proof systems over formulas". In: *Electronic Colloquium on Computational Complexity (ECCC)* 8.11 (2001).

[29] Dima Grigoriev, Edward A. Hirsch, and Dmitrii V. Pasechnik. "Complexity of semi-algebraic proofs". In: *Electronic Colloquium on Computational Complexity (ECCC)* 103 (2001).

[30] Joshua A. Grochow and Toniann Pitassi. "Circuit Complexity, Proof Complexity, and Polynomial Identity Testing". In: *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*. IEEE Computer Society, 2014, pp. 110–119.

[31] Jan Krajíček. *Bounded Arithmetic, Propositional Logic and Complexity Theory*. Cambridge University Press, 1995.

[32] Mladen Mikša and Jakob Nordström. "A Generalized Method for Proving Polynomial Calculus Degree Lower Bounds". In: *30th Conference on Computational Complexity, CCC 2015, June 17-19, 2015, Portland, Oregon, USA*. Ed. by David Zuckerman. Vol. 33. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015, pp. 467–487.

[33] Jakob Nordström. "On the Interplay Between Proof Complexity and SAT Solving". In: *ACM SIGLOG News* 2.3 (Aug. 2015), pp. 19–44.

[34] Toniann Pitassi. "Algebraic Propositional Proof Systems". In: *Descriptive Complexity and Finite Models, Proceedings of a DIMACS Workshop, January 14-17, 1996, Princeton University*. Ed. by Neil Immerman and Phokion G. Kolaitis. Vol. 31. DIMACS Series in Discrete Mathematics and Theoretical Computer Science. American Mathematical Society, 1996, pp. 215–244.

[35] Pavel Pudlák. "Proofs as Games". In: *The American Mathematical Monthly* 107.6 (2000), pp. 541–550.

[36] Pavel Pudlák and Russell Impagliazzo. "A lower bound for DLL algorithms for k-SAT (preliminary version)". In: *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms, January 9-11, 2000, San Francisco, CA, USA*. Ed. by David B. Shmoys. ACM/SIAM, 2000, pp. 128–136.

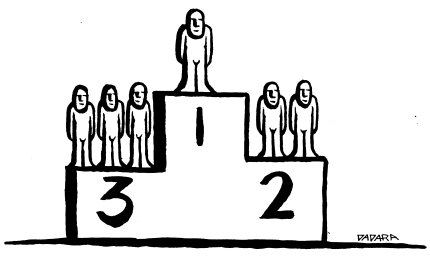
[37] Ran Raz and Iddo Tzameret. "The Strength of Multilinear Proofs". In: *Computational Complexity* 17.3 (2008), pp. 407–457.

[38] John Alan Robinson. "A Machine-Oriented Logic Based on the Resolution Principle". In: *J. ACM* 12.1 (1965), pp. 23–41.

[39] G.S. Tseitin. "On the Complexity of Derivation in Propositional Calculus". English. In: *Automation of Reasoning*. Ed. by Jörg H. Siekmann and Graham Wrightson. Symbolic Computation. Springer Berlin Heidelberg, 1983, pp. 466–483.

[40] Alasdair Urquhart. "Hard examples for resolution". In: *J. ACM* 34.1 (1987), pp. 209–219.

Contributions by EATCS Award Recipients



Abstracts of the EATCS Distinguished Dissertations 2015

Luca Aceto
ICE-TCS, School of Computer Science
Reykjavik University

In 2015, the EATCS established the Distinguished Dissertation Award to promote and recognize outstanding dissertations in the field of Theoretical Computer Science. Each of the selected dissertations receives a prize of 1000 Euro and the award receiving dissertations are published on the EATCS web site, where all the EATCS Distinguished Dissertations are collected at `http://eatcs.org/index.php/dissertation-award`.

The 2015 EATCS Distinguished Dissertation Award committee consisted of Javier Esparza, Michal Feldman, Fedor Fomin, Luke Ong and Giuseppe Persiano (chair). The committee selected the following three dissertations for the award from a collection of outstanding theses:

- Radu Curticapean. *The Simple, Little and Slow Things Count: On Parameterized Counting Complexity*. Department of Computer Science at Saarland University, Saarbrücken, Germany. Supervisor: Markus Bläser.
- Heng Guo. *Complexity Classification of Exact and Approximate Counting Problems*. Department of Computer Sciences, University of Wisconsin-Madison. Advisor: Jin-Yi Cai.
- Georg Zetzsche. *Monoids as storage mechanisms*. Department of Computer Science at the University of Kaiserslautern. Supervisor: Roland Meyer.

The committee's laudatio for Radu Curticapean's thesis reads as follows:

> The thesis is in the field of parameterized complexity for counting problems. It concentrates on the classical problem of counting perfect matchings and gives a polynomial time algorithm for the problem on all graphs that exclude a minor that can be drawn in the plane with at most one crossing. The thesis also considers the problem of counting the number of subgraphs of a graph G that are isomorphic to a given graph H. Here the size of H is considered a parameter. The main contribution is an almost tight hardness proof. Finally, the thesis presents also conditional lower bounds for counting problems under the exponential time hypothesis and its counting version.
>
> The results presented in the dissertation significantly advance our understanding of various aspects of counting problems and solve open questions asked earlier by other researchers.

The citation for Heng Guo's dissertation states:

> The thesis considers the Holant problems and proves that each Holant problem is either P-time solvable or P-time solvable on planar graphs and #P-hard on general graphs or #P-hard for planar graphs. Moreover, the thesis presents an FPTAS for the anti-ferromagnetic 2-spin system up to the tree uniqueness threshold. This is tight as beyond this threshold the existence of such an algorithm would imply NP=RP. These results comprise a fraction of the thesis but are noteworthy because they stand by themselves as results that are fundamental, and which will be remembered.

Last, but by no means least, here is what the committee wrote about Georg Zetzsche's thesis work:

> This thesis is a systematic study of the languages of finite automata extended with storage mechanisms, using the framework of valence automata. The stores, which are modeled by graph monoids, can capture mechanisms ranging from counters and stacks to blind and partially blind counters. Consequently the work deals in a unified way with such diverse models as pushdown automata and Petri nets. Impressive in breadth and depth, the thesis contains answers to a number of notable problems. Among many other results, Zetzsche's work characterizes the monoids that yield automata whose languages properly contain the regular languages; it also characterizes those whose languages have a semi-linear Parikh image. It provides, for several classes of automata, the first algorithms to compute the downward closure of their languages. These results have clear and compelling applications to the theory of formal verification.

The recipients of the 2015 EATCS Distinguished Dissertation Award have contributed a summary of the work presented in their theses to this issue of the Bulletin of the EATCS. I trust that their surveys will be of interest to readers of the Bulletin, regardless of their main research interests. Enjoy them!

THE SIMPLE, LITTLE AND SLOW THINGS COUNT: ON PARAMETERIZED COUNTING COMPLEXITY

Radu Curticapean*

Abstract

This contribution is intended to be a self-contained and minimally technical exposition of the material in my 2015 dissertation, which was supervised by Markus Bläser. As its title suggests, the thesis investigates the complexity of combinatorial counting problems in the frameworks of parameterized (and exponential-time) complexity. More precisely, the following specific settings are explored:

- Counting perfect matchings in structurally "simple" graphs, for instance, in graphs that exclude specific fixed minors
- Counting small subgraph patterns in large host graphs
- Exponential lower bounds on the running time needed to solve counting problems, assuming popular conjectures such as the exponential-time hypothesis

1 Introduction

Many problems in theoretical computer science ask about the *existence* of solutions to a given instance of a problem. This includes, most prominently, the problems in NP, such as the NP-complete Boolean satisfiability problem SAT. However, in both practical and theoretical applications, it may be equally important to *find* a solution, to *list* all solutions, or to *count* the solutions for a given input–it is this last problem that we study in the dissertation. For instance, we can extend SAT to a counting problem #SAT, which asks, given as input a Boolean formula φ, to determine the number of assignments satisfying φ. This is obviously more difficult than merely deciding satisfiability of φ, and by Toda's theorem [41],

*Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI). Supported by ERC Starting Grant PARAMTIGHT (No. 280152). The work on this thesis was done under the supervision of Markus Bläser while the author was a student at Saarland University.

an oracle for #SAT even gives us polynomial-time algorithms for problems we assume to lie outside of NP, namely, the entire polynomial-time hierarchy.

However, counting problems also occur in scientific disciplines other than computational complexity theory: For instance, in statistical physics, various thermodynamic properties of systems can be expressed in terms of their *partition functions*, which are essentially weighted sums over the system's state space [32]. Since such spaces are typically of exponential size, brute-force summations over all states are prohibited, so more efficient algorithms are required (and can sometimes even be obtained [40, 31]). Furthermore, counting also occurs in probabilistic inference [8], since asking for the number of satisfying assigments to a formula φ is equivalent to asking for the probability that a random assignment satisfies φ. In the areas of bioinformatics and network analysis, counting problems occur when one wants to prove that a specific pattern occurs with significant frequency in a given network [27, 42].

The complexity of counting problems

For the vast majority of interesting counting problems, efficient algorithms are not known, creating a need for a complexity theory of counting problems. This was provided by Valiant in his classical paper [44], where he introduced the complexity class #P that captures the counting versions of problems in NP. For instance, #SAT is complete for #P under polynomial-time many-one reductions. More interestingly however, Valiant identified natural #P-complete counting problems whose corresponding decision version can be solved in polynomial time. For instance, his paper contains a seminal proof that counting perfect matchings in a bipartite graph is #P-complete (under polynomial-time Turing reductions), even though the existence of a perfect matching in a given graph can be tested in polynomial time [23].

Since this initial result, the complexity-theoretic study of counting problems advanced to a classical sub-area of complexity theory, and the specific problem of counting perfect matchings played an important role throughout this development. We will abbreviate this problem by #PerfMatch in the following, and we also consider a generalization to edge-weighted graphs: Here, the task is, given a graph with edge-weights $w : E(G) \to \mathbb{Q}$, to compute the value

$$\#\mathrm{PerfMatch}(G) = \sum_{M} \prod_{e \in M} w(e), \tag{1}$$

where M ranges over all perfect matchings of G [45]. This clearly generalizes the problem #PerfMatch by setting all edge-weights to 1.

For instance, it was actually already shown before Valiant's hardness result that #PerfMatch (even with edge-weights) can be solved in polynomial time on planar input graphs [40, 31, 32]. Building upon this, Valiant recently introduced the notion of holographic algorithms [45], which allow us to reduce a variety of other counting problems to this specific positive case. Furthermore, in combinatorics and algebraic complexity theory, the number of perfect matchings in a bipartite graph with $n + n$ vertices and bi-adjacency matrix A is known as the *permanent* of A, which is defined as

$$\mathrm{perm}(A) = \sum_{\pi} \prod_{i=1}^{n} A(i, \pi(i)),$$

where π ranges over all permutations of $1, \ldots, n$. The permanent is central to algebraic complexity theory, where an algebraic variant of the "P vs. NP" question asks to distinguish the complexity of the permanent from that of the deceivingly similar looking determinant [1].

Classical strategies for coping with hardness

As it turned out that #PerfMatch and many other interesting counting problems are #P-complete, relaxed versions were introduced to cope with their computational hardness. Classical examples for such relaxations include:

- Restricted input classes: As mentioned, #PerfMatch is polynomial-time solvable on planar graphs [40, 31, 32]. However, it remains #P-complete on 3-regular graphs [20]. The related problem of counting all (not necessarily perfect) matchings is #P-complete on planar 3-regular graphs [43].

- Approximate counting: On bipartite graphs, the problem #PerfMatch admits a fully polynomial randomized approximation scheme [30]. That is, given an n-vertex bipartite graph and numbers ϵ, δ as input, we can output a multiplicative $(1 \pm \epsilon)$ approximation to #PerfMatch(G) with probability at least $1 - \delta$ in time polynomial in $n, \epsilon^{-1}, \delta^{-1}$.

- Counting modulo fixed numbers: The parity of #PerfMatch(G) is easily computed if G is a bipartite graph with bi-adjacency matrix A. Namely, since the permanent and the determinant of any matrix coincide modulo 2, the parity of #PerfMatch(G) is that of $\det(A)$. This argument can be generalized to counting modulo 2^t for fixed $t \in \mathbb{N}$ [44]. On the other hand, #PerfMatch(G) modulo q is NP-hard to compute (under randomized reductions) when q is not a power of two [44].

In my dissertation, we study two of the most recent relaxations of counting problems, namely, their parameterized complexity (introduced by Flum and Grohe

[25]) and their exponential-time complexity (introduced by Dell et al. [21]).[1]

1.1 Parameterized counting complexity

Parameterized counting complexity is dedicated to the study of *parameterized counting problems*. These are pairs $(\#\mathsf{A}, \kappa)$, where $\#\mathsf{A}$ is a counting problem with inputs $\{0,1\}^*$, together with a *parameterization* $\kappa : \{0,1\}^* \to \mathbb{N}$ such that, for "typical" instances x, the parameter value $\kappa(x)$ is much smaller than the input length $|x|$. The concrete choice of a parameterization depends on the application, but for graph problems, parameterizations can broadly be classified as follows:

- *Structural parameters* are intended to measure some notion of complexity in the input graph, and a small parameter value means that the input enjoys a simple structure that could be used algorithmically. Exemplary parameters for a graph G include its maximum degree $\Delta(G)$, its crossing number $\mathrm{cr}(G)$, or its genus $\gamma(G)$.[2]
- For some counting problems, the input itself already contains a number $k \in \mathbb{N}$ such that structures of size k are to be counted; in such cases, it can make sense to *parameterize by the solution size k*. For instance, given a graph G and $k \in \mathbb{N}$, we can ask for the number of matchings in G with exactly k edges, or the number of vertex covers with precisely k vertices. When we parameterize these problems by k, this means intuitively that we are interested in solutions that are much smaller than G. Such a perspective makes sense, e.g., when small patterns are counted in huge databases.

If a suitable parameter κ was identified for a given $\#\mathsf{P}$-hard problem $\#\mathsf{A}$, we now ask whether it can be used algorithmically. The answer to this question may take one of the following forms [24]:

1. In the worst case, $\#\mathsf{A}$ is already $\#\mathsf{P}$-complete for constant values of $\kappa(x)$. For instance, we mentioned earlier that #PerfMatch is $\#\mathsf{P}$-complete, even for input graphs G of maximum degree $\Delta(G) \le 3$.
2. The situation is more favorable if, for every $k \in \mathbb{N}$, the problem $\#\mathsf{A}$ can be solved in polynomial time on inputs x with $\kappa(x) \le k$, or even better, if we can find an algorithm for $\#\mathsf{A}$ that runs in time $O(|x|^{f(\kappa(x))})$, where f is

[1] For *decision* problems, parameterized complexity was already introduced by Downey and Fellows [22], and exponential-time complexity was introduced by Impagliazzo et al. [29, 28].

[2] Here, the crossing number of G is the minimum number of edge-crossings over all drawings of G in the plane. The genus of G is the minimum genus of a surface on which G can be drawn without crossings.

some computable function. In this case, we speak of an **XP algorithm**. For example, such algorithms exist for counting matchings with k edges in graphs with n vertices, since brute-force solves this problem in time $n^{O(k)}$.

3. In the ideal setting, we even obtain an algorithm for #A that admits a constant $c \in \mathbb{N}$ such that for every fixed value of $\kappa(x)$, the running time of the algorithm is bounded by $O(|x|^c)$. Compared to the previous case, we can hence even remove $\kappa(x)$ from the exponent of $|x|$. This leads to the notion of **fixed-parameter tractability (FPT)**: A problem with parameterization κ is FPT if it can be solved in time $O(f(\kappa(x)) \cdot |x|^c)$, where c is a fixed constant and f is any computable function. For instance, vertex-covers of size k can be counted in time $O(2^k \cdot n)$ on n-vertex graphs [25], and #PerfMatch can be solved[3] in time $O(4^g \cdot n^3)$ on n-vertex graphs of genus g [26].

These definitions allow us to state some of the main goals of parameterized algorithms and complexity theory: Given a parameterized problem $(\#\mathsf{A}, \kappa)$, can we find an FPT algorithm (or at least an XP algorithm)? If not, can we explain our lack of progress as a consequence of widely-believed assumptions in complexity theory? For instance, to rule out XP algorithms for parameterized counting problems, it suffices to prove #P-hardness of the problem for a fixed parameter value. This approach however fails to rule out FPT algorithms for problems that already admit XP algorithms, since these are polynomial-time solvable for every fixed parameter value.

To explain the absence of FPT algorithms, a complexity class #W[1] of parameterized counting problems (analogously to #P) was introduced, along with a suitable hardness notion [25]. This class #W[1] can be defined as the set of all parameterized problems that reduce, by means of **parameterized reductions**, to counting k-cliques in a graph, parameterized by k. Here, a parameterized (Turing) reduction from a parameterized problem $(\#\mathsf{A}, \kappa)$ to another $(\#\mathsf{B}, \tau)$ is an algorithm that solves #A on inputs x in time $f(\kappa(x)) \cdot |x|^{O(1)}$ when given oracle access to #B. However, all oracle queries y to #B must satisfy $\tau(y) \leq g(\kappa(x))$. In the above statement, both f and g are arbitrary computable functions. We observe that the class of FPT problems is closed under parameterized reductions.

Using parameterized reductions from counting k-cliques, Flum and Grohe [25] showed that the problems of counting paths (or cycles) of length k are each #W[1]-complete. Hence, under the reasonable and widely-believed assumption that counting k-cliques admits no FPT algorithm, these problems do not admit FPT algorithms. This is particularly interesting, since we can find a path (or cycle) of length k in time $2^{O(k)} \cdot n^{O(1)}$ [2].

[3] We assume here that an embedding into the surface is given along with the input.

1.2 Exponential-time complexity for counting problems

Once a parameterized problem was classified as FPT or #W[1]-hard, we know whether to expect running times of type $f(k) \cdot n^{O(1)}$ or $n^{f(k)}$. However, an even more fine-grained analysis is often possible: Using the theory of exponential-time complexity [29, 21], one can often pinpoint the optimal asymptotic growth of f under reasonable complexity-theoretic assumptions.

In our applications, we use the exponential-time hypothesis ETH [29] and its counting version #ETH [21], which postulate, roughly speaking, that SAT (and #SAT) on formulas in 3-CNF with n variables cannot be solved in time $2^{o(n)}$. Our current understanding does not allow us to prove these hypotheses, as they clearly imply the separation of P and NP (or #P). Nevertheless, a falsification of ETH or #ETH would imply an unexpected breakthrough in the theory of satisfiability algorithms. And, regardless of our belief in ETH or #ETH, it cannot be denied that, for surprisingly many problems, these hypotheses imply lower bounds that match the best known algorithms [34].

For instance, consider a counting problem #A with parameterization κ, and assume there was a polynomial-time reduction from #SAT to #A that maps formulas φ with n variables to instances x of #A with parameter $\kappa(x) = O(n)$. Then #ETH rules out algorithms with running time $2^{o(\kappa(x))} \cdot |x|^{O(1)}$ for the problem. This allows us, e.g., to rule out algorithms with running time $2^{o(g)} \cdot n^{O(1)}$ for #PerfMatch on graphs of genus g [18], thus complementing the previously mentioned upper bound of $O(4^g \cdot n^3)$. We can also use #ETH to derive lower bounds for #W[1]-hard problems: For instance, it is known that #ETH rules out algorithms with running time $f(k) \cdot n^{o(k)}$ for the problem of counting k-cliques [9]. In particular, #ETH thus implies that FPT and #W[1] do not coincide.

1.3 Organization of the dissertation

The remainder of this contribution follows the outline of my dissertation [14], which is structured into one introductory part and three main parts, each of which corresponds to an adjective in the title of the thesis.

In the introductory part of my dissertation, we first collect some basics from complexity theory, graph theory, algebra and combinatorics. We also include an introduction to the Holant framework, which will provide a clean language for many of the subsequent arguments. Building upon this, the first main part analyzes the problem #PerfMatch under structural parameters. In the second main part, we consider the problem of counting small subgraph patterns in large host graphs. In the last part, we study conditional exponential lower bounds for counting problems.

2 The Holant framework

One of the goals of the dissertation was to develop general tools for studying the complexity of counting problems. To this end, we first extend the theory of so-called *Holant problems* [45, 5, 7, 6]. These problems are defined on graphs $G = (V, E)$ whose vertices $v \in V$ are labelled with *signatures* f_v: If $I(v)$ denotes the set of edges incident with v, then f_v is a function that maps assignments $\{0, 1\}^{I(v)}$ to complex numbers. Given such a graph, the problem then lies in evaluting a particular quantity Holant(G), which is a sum over all Boolean assignments $x \in \{0, 1\}^E$ to the edges of G. In this sum, each assignment x is weighted by the product of evaluations $f_v(x|_{I(v)})$ over all $v \in V$. Here, $x|_{I(v)}$ denotes the restriction of the assignment x to edges in $I(v)$. Thus, we obtain

$$\mathrm{Holant}(G) = \sum_{x \in \{0,1\}^E} \prod_{v \in V} f_v(x|_{I(v)}).$$

As an example, we can express #PerfMatch on unweighted graphs as a Holant problem: Given a graph G, assign a signature $f_v : I(v) \to \{0, 1\}$ to each vertex $v \in V$ that checks whether the assignment to $I(v)$ has exactly one edge of value 1. In this case, f_v is supposed to output 1, otherwise 0. We can easily see that the non-zero terms in Holant(G) then correspond precisely to the perfect matchings in G. Using more complex signatures, we can also express other problems as Holant problems, including the problem #SAT.

In the thesis, we introduce two tools for Holant problems, which are used throughout the main parts and also found applications in other projects [19, 18]. Firstly, we construct a uniform reduction from Holant problems with arbitrary signatures to #PerfMatch. To this end, we replace the vertices of the graph G in a Holant problem by gadgets that simulate their signatures. In particular, we can show that such gadgets exist for all signatures, unless they are ruled out for trivial reasons. The previous literature focused on the simulation of signatures by *planar* gadgets [45, 4]; for many signatures however, such gadgets do not exist. In joint work with Dániel [18], we later used this technique to show that #PerfMatch cannot be solved in time $2^{o(k)} \cdot n^{O(1)}$ on graphs of crossing number k or treewidth k, unless #ETH fails.

Secondly, in joint work with Mingji Xia [19], we introduce linear combinations of signatures as a technique for parameterized reductions between Holant problems: If a graph G in a Holant problem contains a small number k of "difficult" signatures that can be expressed as linear combinations of $c \in \mathbb{N}$ "simple" signatures, then we can express Holant(G) as a linear combination of c^k values Holant(G'), where each instance G' contains only simple signatures. The resulting running time of $c^k \cdot n$ is compatible with parameterized reductions. Apart from the applications in the thesis, we used this technique to show that #PerfMatch modulo

2^k is $W[1]$-hard to compute (under randomized reductions) [19], complementing the known $n^{O(k)}$ time algorithm [44].

3 Counting perfect matchings in simple graphs

In the first main part of the dissertation, we study the problem of computing #PerfMatch(G) on graphs G that exclude fixed minors H. This can be considered as a generalization of the polynomial-time solvable case of planar graphs [40, 31, 32], since any planar graph excludes both $K_{3,3}$ and K_5 as minors.

Graph minors play a fundamental role in graph theory, where they led to the celebrated Graph Minor Theorem [38]: Every graph class C that is closed under taking minors can be expressed by a finite set $F(C)$ of forbidden minors. That is, any given graph G is contained in C if and only if G contains none of the graphs in $F(C)$ as a minor. This holds exemplarily for the class C of planar graphs, where $F(C)$ takes the form of $\{K_{3,3}, K_5\}$.

Graph minors are quite relevant to the problem of counting perfect matchings, since almost all known polynomial-time algorithms for #PerfMatch on restricted graph classes in fact apply to classes that exclude at least one fixed minor. This holds for the class of planar graphs, and for subsequent algorithms on $K_{3,3}$-free graphs [33], graphs of bounded genus [26], K_5-free graphs [39], and graphs of bounded treewidth [46].[4] In an attempt to connect these individual dots, we asked ourselves how #PerfMatch behaves on graph classes excluding an arbitrary fixed minor H, when parameterized by the size of H.

To answer this question, we use the Graph Structure Theorem [37], which describes the structure of graphs excluding fixed minors H. It asserts that, for any fixed graph H, there is a constant $k = k(H)$ such that the H-free graphs can be obtained as *clique-sums* from graphs that are *k-almost-embeddable* in a surface of genus k. Here, a clique-sum of graphs A and B is executed by choosing equal-sized cliques in A and B, then taking the disjoint union of A and B while identifying the chosen cliques, and finally deleting an arbitrary set of edges from the resulting clique in the union. Furthermore, a graph F is k-almost-embeddable in a surface S if we can delete k so-called apex vertices from F such that the resulting graph can be embedded in S without crossings, with the exception of k vortices, namely, k faces into which certain "thin" non-planar graphs may be embedded.

[4] An exceptional tractable class is, e.g., the class $\mathcal{K}$ of complete graphs. This class clearly excludes no fixed minor, but the number of perfect matchings in complete graphs can be computed by a closed formula. This fact is actually subsumed by a more general result that #PerfMatch can be computed in polynomial time on graph classes of bounded clique-width [35]. We consider this result as an exception, since it does not apply to the edge-weighted case of #PerfMatch, whereas the other algorithms in our list do.

As mentioned before, #PerfMatch is FPT on graphs that can be embedded on a surface of genus k [26]. To understand the case of #PerfMatch on general H-minor free grpahs, we hence need to understand the influence of apex vertices, vortices and clique-sums on the complexity of this problem. In joint work with Mingji Xia, we show that #PerfMatch already becomes #W[1]-hard on planar graphs with k apex vertices. This is the first application of the technique of linear combinations of signatures mentioned in Section 2.

Theorem 1 ([19]). *The following problem is* #W[1]*-hard: Given a graph G and a vertex set $A \subseteq V(G)$ such that $G - A$ is planar, determine* #PerfMatch(G)*, parameterized by $|A|$. Furthermore, this problem admits no $n^{o(k/\log k)}$ time algorithm unless* #ETH *fails.*

Theorem 1 implies that #PerfMatch is #W[1]-hard on graphs excluding a fixed minor H, when parameterized by the size of H, since planar graphs with a fixed number of apex vertices exclude fixed minors. By a further non-trivial reduction, we can strengthen Theorem 1 to obtain the #W[1]-hardness of the related problem of counting matchings with exactly k unmatched vertices in planar graphs [15].

Despite these negative results, we can obtain FPT algorithms for restricted classes of excluded minors. In particular, we identify one such class in the thesis, namely, the class of minors that can be drawn in the plane with at most one crossing. This class includes the graphs $K_{3,3}$ and K_5, and hence, this result generalizes some of the algorithms mentioned in the beginning of this section.

Theorem 2 ([12]). *If H is a graph that can be drawn in the plane with at most one crossing, then the problem* #PerfMatch *can be solved in time $O(f(H) \cdot n^4)$ on graphs that exclude H as a minor. Here, f is a computable function.*

The dissertation does not answer the question whether #PerfMatch can actually be solved in time $n^{f(H)}$ on graphs excluding arbitrary fixed minors H. Recent unpublished work by the author however suggests a negative answer.

4 Counting small subgraphs

In the second main part of the thesis, we count small subgraph patterns H on k vertices, such as paths or cycles of size k, in general host graphs G with n vertices. That is, given graphs H and G, we wish to count all subgraphs of G that are isomorphic to H, parameterized by k. This has vast applications in network analysis, see [36].

A simple brute-force approach guarantees a running time of $n^{O(k)}$ for this problem, which may however already be prohibitive for small values of k. Furthermore, as mentioned in Section 1.2, we do not expect $n^{o(k)}$ time algorithms for

k-vertex subgraphs such as cliques under #ETH. We are hence interested in identifying additional properties on H that can be exploited to render the subgraph counting problem FPT. To this end, we introduce the following problem $\#\mathsf{Sub}(\mathcal{H})$ for fixed recursively enumerable graph classes $\mathcal{H}$: Given graphs G and $H \in \mathcal{H}$, count subgraphs of G that are isomorphic to H, parameterized by $|V(H)|$. Our goal is to determine, for each class $\mathcal{H}$, whether $\#\mathsf{Sub}(\mathcal{H})$ is FPT or #W[1]-hard.

For instance, if $\mathcal{H}$ is the class of cycles or the class of paths, then the #W[1]-completeness of $\#\mathsf{Sub}(\mathcal{H})$ was already proven by Flum and Grohe [25], despite the decision versions of these problems being FPT. The same authors also conjectured that $\#\mathsf{Sub}(\mathcal{M})$ is #W[1]-complete for the class $\mathcal{M}$ of matchings. This corresponds to counting matchings with k edges in graphs, and can thus be considered as a parameterized version of #PerfMatch.

In joint work with Markus Bläser [3], we first showed that a weighted version of this problem is #W[1]-hard. The weights in this version are defined analogously as in #PerfMatch. Building upon this, the #W[1]-completeness of the unweighted version was shown later [11]. This first proof was however mostly superseded by a simplified proof that resulted from joint work with Dániel Marx [17], and which exploits linear combinations of signatures. We additionally obtain an almost-tight lower bound for this problem under #ETH:

Theorem 3 ([17])**.** *The problem* $\#\mathsf{Sub}(\mathcal{M})$ *ist* #W[1]*-complete: Given a graph G and $k \in \mathbb{N}$, it is thus* #W[1]*-complete to count matchings with k edges in G. Furthermore, an $n^{o(k/\log k)}$ time algorithm for this problem would refute* #ETH.

This constitutes a useful reduction source to prove #W[1]-hardness of other counting problems. In particular, it allows us to classify the problems $\#\mathsf{Sub}(\mathcal{H})$, since $\#\mathsf{Sub}(\mathcal{M})$ represents the minimal hard case in this setting: It is known that $\#\mathsf{Sub}(\mathcal{H})$ can even be solved in time $n^{O(1)}$ if the graphs in $\mathcal{H}$ only contain matchings of constant size [47]. This applies, e.g., to the class $\mathcal{S}$ of stars, or generally to classes with constant-sized vertex covers. On the other hand, if $\mathcal{H}$ is a graph class whose graphs contain arbitrarily large matchings, then we show in joint work with Dániel Marx that $\#\mathsf{Sub}(\mathcal{M})$ can be reduced to $\#\mathsf{Sub}(\mathcal{H})$. This yields:

Theorem 4 ([17])**.** *Let $\mathcal{H}$ be a recursively enumerable class of graphs. If there is a constant $c \in \mathbb{N}$ such that no graph in $\mathcal{H}$ contains a matching with c edges, then* $\#\mathsf{Sub}(\mathcal{H})$ *can be solved in time $O(n^d)$, where d depends on c. On the other hand, if $\mathcal{H}$ contains arbitrarily large matchings, then* $\#\mathsf{Sub}(\mathcal{H})$ *is* #W[1]*-complete.*

It should be mentioned that this theorem actually shows for each class $\mathcal{H}$ whether $\#\mathsf{Sub}(\mathcal{H})$ is polynomial-time solvable or #W[1]-complete. Assuming that FPT and #W[1] do not coincide, we thus obtain an exact classification of the problems #$\mathsf{Sub}(\mathcal{H})$ that can be solved in polynomial time. Indeed, such a classification would not be possible by merely assuming that P and #P do not coincide, since there exist #P-intermediate cases for #$\mathsf{Sub}(\mathcal{H})$ [10].

5 Conditional lower bounds for counting problems

In the last part of the thesis, we investigate whether classical counting problems can be solved in sub-exponential time, i.e., in time $2^{o(n)}$ on graphs with n vertices. Throughout this part, we assume the exponential-time hypothesis #ETH from Section 1.2. Furthermore, in this setting as well, the problem #PerfMatch does not fail to be a useful reduction source for other hardness results, so we focus on lower bounds for this particular problem.

Building upon the work from [21], proving tight lower bounds for #PerfMatch essentially boils down to finding a "resourceful" way of simulating the edge-weight -1 in the weighted version of #PerfMatch. That is, we define a problem #PerfMatch$^{-1,0,1}$ of counting weighted perfect matchings as in (1), but only on graphs with edge-weights -1 and 1. To avoid confusion, we will henceforth denote #PerfMatch on unweighted graphs by #PerfMatch^{0,1}. It was shown [21] that an algorithm with running time $2^{o(n)}$ for #PerfMatch$^{-1,0,1}$ on graphs with n vertices and $O(n)$ edges would refute #ETH.

To proceed, it is essential to remove the edge-weight -1 from #PerfMatch$^{-1,0,1}$, as it would otherwise be very unclear how to handle this negative weight in reductions to other target problems. A classical solution for such weight removal is based on polynomial interpolation [43]: Given an n-vertex graph G with edge-weights -1 and 1, we can define a graph G_x by replacing each occurrence of -1 with the indeterminate x. The quantity p := #PerfMatch(G_x) is then a polynomial of degree at most $\frac{n}{2}$ in x, and we have $p(-1)$ = #PerfMatch(G) by definition. Hence, if we know the values $p(1), \ldots, p(\frac{n}{2}+1)$, we can use polynomial interpolation to obtain the coefficients of p and thus evaluate $p(-1)$.

For positive integer values of t however, the evaluation of $p(t)$ can be reduced to #PerfMatch^{0,1}: An edge $uv \in E(G)$ of weight $t \in \mathbb{N}$ in a graph G is easily simulated by placing t parallel edges of weight 1 between u and v, then subdividing each of these edges twice. A graph with n vertices, m edges, and edge-weights from $\{1, \ldots, b\}$ is thus transformed to an unweighted graph on $O(n + bm)$ vertices and edges. Using this with the interpolation argument above, we obtain a polynomial-time Turing reduction from #PerfMatch$^{-1,0,1}$ to #PerfMatch^{0,1} which transforms n-vertex graphs G with edge-weights ± 1 to unweighted simple graphs with $O(n^2)$ vertices. This quadratic blowup however prevents us from proving a tight lower bound under #ETH: To obtain an algorithm with running time $2^{o(n)}$ for #PerfMatch$^{-1,0,1}$, and thus refute #ETH, we would need to solve #PerfMatch^{0,1} in time $2^{o(\sqrt{n})}$. One could try to find more resourceful ways of simulating positive weights $t \in \mathbb{N}$, and indeed gadgets on $O(\log t)$ vertices exist [21], thus ruling out $2^{o(n/\log n)}$ time algorithms for #PerfMatch^{0,1}. This is however still not tight, and we can see that any gadgets simulating edges of weight $t \in \mathbb{N}$ *must* have size $\omega(1)$, thus apparently rendering interpolation futile for proving tight lower bounds.

In the dissertation, we solve this problem in two ways: Firstly, we introduce "block interpolation", a general approach to make polynomial interpolation compatible with tight lower bounds under #ETH. This technique relies on the insight that interpolation arguments as above, which are built around polynomials in a single indeterminate x, can often be carried out on polynomials in indeterminates $x_1, \ldots, x_t$, each of which occurs with some maximum degree d. Such polynomials can be interpolated from their evaluations on a grid $\{1, \ldots, d+1\}^t$, which requires only $d + 1$ distinct values to be substituted into each individual indeterminate. By trading off t with d, we obtain reductions that run in sub-exponential time, but only create linear-sized reduction images. This general approach allows us to prove lower bounds for the following problems:

Theorem 5 ([13]). *Unless* #ETH *fails, the problems of counting matchings, perfect matchings, independent sets, and vertex covers cannot be solved in time* $2^{o(n)}$ *on n-vertex graphs.*

Our second solution for weight removal applies only to the specific problem #PerfMatch, but has other benefits: For any graph G with edge-weights ± 1, we show how to construct unweighted graphs G_1 and G_2 such that #PerfMatch(G) is the difference of #PerfMatch(G_1) and #PerfMatch(G_2).

Theorem 6 ([16]). *If G is a graph with n vertices, m edges, and edge-weights* -1 *and* 1*, then we can construct unweighted graphs* G_1 *and* G_2 *on* $O(n+m)$ *vertices and edges such that* #PerfMatch(G) = #PerfMatch(G_1) − #PerfMatch(G_2).

This immediately transfers the known lower bound for #PerfMatch$^{-1,0,1}$ to one for #PerfMatch^{0,1}. However, it also gives us insights into the structural complexity of counting perfect matchings, as it allows us to prove that the following "equality testing" version #PerfMatch$_=$ of #PerfMatch is complete for the class $\mathsf{C_=P}$: Given two unweighted graphs, this problem asks to decide whether they have the same number of perfect matchings. Here, $\mathsf{C_=P}$ can be defined as the class of problems that are polynomial-time many-one reducible to deciding whether two Boolean formulas φ_1, φ_2 have the same number of satisfying assignments. Furthermore, bridging quantitative lower bounds and structural complexity, our proof shows that ETH rules out a $2^{o(n)}$ time algorithm for #PerfMatch$_=$.

Acknowledgments

I want to thank my advisor Markus Bläser for all the guidance and support he gave me during my PhD time and afterwards, and my other co-authors Mingji Xia and Dániel Marx for great collaborations.

References

[1] Manindra Agrawal. Determinant versus permanent. In *Proceedings of the 25th International Congress of Mathematicians, ICM 2006*, volume 3, pages 985–997, 2006.

[2] Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *J. ACM*, 42(4):844–856, 1995.

[3] Markus Bläser and Radu Curticapean. Weighted counting of k-matchings is #W[1]-hard. In *IPEC 2012*, pages 171–181, 2012.

[4] Jin-Yi Cai and Aaron Gorenstein. Matchgates revisited. *Theory of Computing*, 10:167–197, 2014.

[5] Jin-yi Cai, Pinyan Lu, and Mingji Xia. Holant problems and counting CSP. In *STOC 2009*, pages 715–724, 2009.

[6] Jin-yi Cai, Pinyan Lu, and Mingji Xia. Computational complexity of Holant problems. *SIAM J. Comput.*, 40(4):1101–1132, 2011.

[7] Jin-yi Cai, Pinyan Lu, and Mingji Xia. Dichotomy for Holant* problems of Boolean domain. In *SODA 2011*, pages 1714–1728, 2011.

[8] Mark Chavira and Adnan Darwiche. On probabilistic inference by weighted model counting. *Artif. Intell.*, 172(6-7):772–799, 2008.

[9] Jianer Chen, Benny Chor, Mike Fellows, Xiuzhen Huang, David W. Juedes, Iyad A. Kanj, and Ge Xia. Tight lower bounds for certain parameterized NP-hard problems. *Inf. Comput.*, 201(2):216–231, 2005.

[10] Yijia Chen, Marc Thurley, and Mark Weyer. Understanding the complexity of induced subgraph isomorphisms. In *ICALP (1)*, pages 587–596, 2008.

[11] Radu Curticapean. Counting matchings of size k is #W[1]-hard. In *ICALP 2013*, pages 352–363, 2013.

[12] Radu Curticapean. Counting perfect matchings in graphs that exclude a single-crossing minor. *CoRR*, abs/1406.4056, 2014.

[13] Radu Curticapean. Block interpolation: A framework for tight exponential-time counting complexity. In *ICALP 2015*, pages 380–392, 2015.

[14] Radu Curticapean. *The simple, little and slow things count: on parameterized counting complexity*. PhD thesis, Universität des Saarlandes, 2015.

[15] Radu Curticapean. Counting matchings with k unmatched vertices in planar graphs. In *ESA 2016*, pages 33:1–33:17, 2016.

[16] Radu Curticapean. Parity separation: A scientifically proven method for permanent weight loss. In *ICALP 2016*, 2016.

[17] Radu Curticapean and Dániel Marx. Complexity of counting subgraphs: Only the boundedness of the vertex-cover number counts. In *FOCS 2014*, pages 130–139, 2014.

[18] Radu Curticapean and Dániel Marx. Tight conditional lower bounds for counting perfect matchings on graphs of bounded treewidth, cliquewidth, and genus. In *SODA 2016*, pages 1650–1669, 2016.

[19] Radu Curticapean and Mingji Xia. Parameterizing the permanent: Genus, apices, minors, evaluation mod 2^k. In *FOCS 2015*, pages 994–1009, 2015.

[20] Paul Dagum and Michael Luby. Approximating the permanent of graphs with large factors. *Theor. Comput. Sci.*, 102(2):283–305, 1992.

[21] Holger Dell, Thore Husfeldt, Dániel Marx, Nina Taslaman, and Martin Wahlen. Exponential time complexity of the permanent and the Tutte polynomial. *ACM Transactions on Algorithms*, 10(4):21, 2014.

[22] Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, 1999.

[23] Jack Edmonds. Paths, trees, and flowers. In *Classic Papers in Combinatorics*, Modern Birkhï£¡user Classics, pages 361–379. Birkhï£¡user Boston, 1987.

[24] J. Flum and M. Grohe. *Parameterized Complexity Theory (Texts in Theoretical Computer Science. An EATCS Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[25] Jörg Flum and Martin Grohe. The parameterized complexity of counting problems. *SIAM Journal on Computing*, (4):892–922, 2004.

[26] Anna Galluccio and Martin Loebl. On the theory of Pfaffian orientations. I. Perfect matchings and permanents. *Electronic Journal of Combinatorics*, 6, 1998.

[27] Sylvain Guillemot and Florian Sikora. Finding and counting vertex-colored subtrees. In *MFCS 2010*, pages 405–416, 2010.

[28] Russel Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001.

[29] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Computer and Sys. Sci.*, 63(4):512–530, 2001.

[30] Mark Jerrum, Alistair Sinclair, and Eric Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *J. ACM*, 51(4):671–697, 2004.

[31] Pieter W. Kasteleyn. The statistics of dimers on a lattice: I. The number of dimer arrangements on a quadratic lattice. *Physica*, 27(12):1209 – 1225, 1961.

[32] Pieter W. Kasteleyn. Graph Theory and Crystal Physics. In *Graph Theory and Theoretical Physics*, pages 43–110. Academic Press, 1967.

[33] Charles Little. An extension of Kasteleyn's method of enumerating the 1-factors of planar graphs. In *Combinatorial Mathematics*, LNCS, pages 63–72. 1974.

[34] Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Lower bounds based on the Exponential Time Hypothesis. *Bulletin of the EATCS*, 84:41–71, 2011.

[35] Johann A. Makowsky, Udi Rotics, Ilya Averbouch, and Benny Godlin. Computing graph polynomials on graphs of bounded clique-width. In *WG 2006*, pages 191–204, 2006.

[36] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, October 2002.

[37] Neil Robertson and Paul D. Seymour. Graph minors. XVI. Excluding a non-planar graph. *J. Comb. Theory, Ser. B*, 89(1):43 – 76, 2003.

[38] Neil Robertson and Paul D. Seymour. Graph minors. XX. Wagner's conjecture. *J. Comb. Theory, Ser. B*, 92(2):325–357, 2004.

[39] Simon Straub, Thomas Thierauf, and Fabian Wagner. Counting the number of perfect matchings in K_5-free graphs. In *CCC 2014*, pages 66–77, 2014.

[40] H. N. V. Temperley and Michael E. Fisher. Dimer problem in statistical mechanics - an exact result. *Philosophical Magazine*, 6(68):1478–6435, 1961.

[41] Seinosuke Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 20(5):865–877, 1991.

[42] Johan Ugander, Lars Backstrom, and Jon M. Kleinberg. Subgraph frequencies: mapping the empirical and extremal geography of large graph collections. In *WWW 2013*, pages 1307–1318, 2013.

[43] Salil P. Vadhan. The complexity of counting in sparse, regular, and planar graphs. *SIAM J. Comput.*, 31(2):398–427, 2001.

[44] Leslie G. Valiant. The complexity of computing the permanent. *Theoret. Comput. Sci.*, 8(2):189–201, 1979.

[45] Leslie G. Valiant. Holographic algorithms. *SIAM J. Comput.*, 37(5):1565–1594, 2008.

[46] Johan M. M. van Rooij, Hans L. Bodlaender, and Peter Rossmanith. Dynamic programming on tree decompositions using generalised fast subset convolution. In *ESA 2009*, pages 566–577, 2009.

[47] Virginia Vassilevska Williams and Ryan Williams. Finding, minimizing, and counting weighted subgraphs. *SIAM J. Comput.*, 42(3):831–854, 2013.

Mapping the complexity of counting problems

Heng Guo
Queen Mary, University of London
h.guo@qmul.ac.uk

1 Computational Counting

The study of computational counting was initiated by Leslie Valiant in the late 70s. In the seminal papers [32, 33], he defined the computational complexity class #**P**, the counting counterpart of **NP**, and showed that the counting version of tractable decision problems can be #**P**-complete. One such example is counting perfect matchings (#PM). These work revealed some fundamental differences between counting problems and decision problems and stimulated a lot of research in the directions of both structural complexity theory and algorithmic design. One of the crowning results in the first direction is Toda's theorem [31], stating that #**P** contains the whole polynomial hierarchy, and in the second direction we have witnessed the success of Markov chain Monte Carlo algorithms [24, 23].

A typical counting problem is #Sat, which asks the number of satisfying assignments of a given CNF formula. An equivalent way to recast this problem is to give each assignment weight 1 if it satisfies the formula, or 0 otherwise. Then the goal is to compute the sum of all weights. Moreover, this weight can be decomposed into a product of weights of all clauses. Thus we are interested in a sum-of-product quantity (e.g. see Eq. (1) in Section 2), which is usually called the partition function.

In fact, the partition function has received immense attention by statistical physicists long before computer scientists. It is a central quantity from which one can deduce various properties of a field or a system. For example, the famous Lee-Yang theorem [27] showed the lack of phase transitions by studying the (complex) zeroes of the partition function. A goal of particular interest is to give explicit formulas of partition functions for various models [22, 28, 30, 25, 26, 1]. When such a formula is found, the model is called "exactly solved".

In computational complexity terms, exactly solvable models are tractable in the sense that we have polynomial-time algorithms to compute the partition functions. A classical gem is the Fisher-Kasteleyn-Temperley (FKT) algorithm, which counts perfect matchings over planar graphs in polynomial time [30, 25, 26].

Valiant introduced matchgates [35, 34] and holographic reductions to extend the reach of the FKT algorithm [36, 37]. These reductions differ from classical ones by introducing quantum-like superpositions. This novel technique yields polynomial time algorithms for a number of problems for which only exponential-time algorithms were previously known.

One natural question arises: what is the true power of the holographic algorithm? In particular, can we solve #**P**-hard problems by holographic algorithms? Since holographic algorithms can solve quite a few seemingly hard problems, it is difficult to rule out this possibility before giving them a systematic study. Holant problems were proposed as a natural framework to answer this question [13, 14]. The key feature of Holant problems is that they include problems like #PM which is central to the study of holographic algorithms but difficult to express in the traditional counting constraint satisfaction problems (#CSP) framework.[1]

Without settling **P** vs #**P**, we hope to answer this question by achieving computational complexity classifications. In other words, we want to map out the landscape of counting problems in terms of their intrinsic complexity and then understand the holographic algorithm by looking at its "territory". In [20], classifications are obtained for #CSP problems as well as under the Holant framework.

A preponderance of evidence suggests the following putative classification of all counting problems defined by local constraints into *exactly* three categories:

(1) those that are **P**-time solvable over general graphs;

(2) those that are **P**-time solvable over planar graphs but #**P**-hard over general graphs;

(3) those that remain #**P**-hard over planar graphs.

Moreover, category (2) usually consists of precisely those problems solvable by holographic algorithms. It has been unfailing in the classification of Tutte polynomials [39], of spin systems [9], and of #CSP [12, 21]. However, this turns out to be false for Holant problems [5], though only in a technical sense. An additional planar tractable case was found in [5], but holographic algorithms remain the most important (albeit not the only) subroutine to solve this case.

In the following we will survey several classification theorems reported in [20]. In Section 2 we review necessary definitions and notations. In Section 3 we give the theorems. At last, we give some interesting examples of holographic transformations in Section 4.

[1] It is provably impossible to express #PM in certain "vertex" models. See [19].

2 Holant Problems

A *signature grid* $\Omega = (G, \mathcal{F}, \pi)$ is a tuple, where $G = (V, E)$ is a graph, $\mathcal{F}$ is a set of functions, and π is a mapping from the vertex set V to $\mathcal{F}$. A Boolean function $f \in \mathcal{F}$ with arity k is a mapping $\{0, 1\}^k \to \mathbb{C}$, and the mapping π satisfies that the arity of $\pi(v)$ (which is a function $f \in \mathcal{F}$) is the same as the degree of v for any $v \in V$. Here we may consider any function with the range of a ring rather than just $\mathbb{C}$, but we choose $\mathbb{C}$ in this survey for clarity. Let $f_v := \pi(v)$ be the function on v. An assignment σ of edges is a mapping $E \to \{0, 1\}$.[2] The weight of σ is the evaluation $\prod_{v \in V} f_v(\sigma |_{E(v)})$, where $E(v)$ denotes the set of incident edges of v. The (counting version of) Holant problem on the instance Ω is to compute the sum of weights of all assignments; namely,

$$\mathrm{Holant}_\Omega = \sum_{\sigma} \prod_{v \in V} f_v(\sigma |_{E(v)}). \tag{1}$$

We also write $\mathrm{Holant}(\Omega; \mathcal{F})$ when we want to emphasize the function set $\mathcal{F}$.

The term Holant was first coined by Valiant in [37] to denote an exponential sum of the above form. Cai, Xia and Lu first formally introduced this framework of counting problems in [10, 11]. We can view each function f_v as a truth table, and then we can represent it by a vector in $\mathbb{C}^{2^{d(v)}}$, or a tensor in $(\mathbb{C}^2)^{\otimes d(v)}$. The vector or the tensor is called the *signature* of a function. When we say "function", we put a slight emphasis on that it is a mapping. When we say "signature", we put a slight emphasis on that it is ready to go through linear transformations. However most of the time in this article, we use the two terms "function" and "signature" interchangeably without special attention.

A Holant problem is parameterized by a set of functions.

Definition 2.1. *Let $\mathcal{F}$ be a set of functions. Define a counting problem* $\mathrm{Holant}(\mathcal{F})$*:*
Input: *A signature grid $\Omega = (G, \mathcal{F}, \pi)$;*
Output: Holant_Ω.

We will use $\mathrm{Pl\text{-}Holant}(\mathcal{F})$ to denote the problem where the input graph is planar.

The main goal here is to characterize what kind of function set $\mathcal{F}$ makes the problem $\mathrm{Holant}(\mathcal{F})$ tractable (or hard).

We use the following notations to denote some special functions. Let $=_k$ denote the equality function of arity k. Let $\textsc{ExactOne}_k$ denote the function that is one if the input has Hamming weight 1 and zero otherwise. Let $\mathcal{EO}$ be the set

[2]In general we may consider non-Boolean functions $E \to [q]$ for a positive integer $q > 2$. For simplicity in this article we focus on the Boolean case.

of $\textsc{ExactOne}_k$ functions for all integers k. Then Holant($\mathcal{EO}$) is the same as the problem of counting perfect matchings.

A function is symmetric iff its function value is preserved under any permutation of its inputs. A symmetric function f on Boolean variables can be expressed by a compact signature $[f_0, f_1, \ldots, f_k]$, where f_i is the value of f on inputs of Hamming weight i. For the Boolean domain $[2] = \{0, 1\}$, $=_k$ function has the signature $[1, 0, \ldots, 0, 1]$ with $k + 1$ entries, and $\textsc{ExactOne}_k$ has signature $[0, 1, 0, \ldots, 0]$ of $k + 1$ entries.

Multiplying a signature $f \in \mathcal{F}$ by a scaler $c \neq 0$ only changes Holant_Ω by an easy to compute factor. Thus it does not change the complexity of Holant($\mathcal{F}$). So we always view f and cf as the same signature. In other words, we consider the projective space of vectors or tensors.

We use Holant $(\mathcal{F} \mid \mathcal{G})$ to denote the Holant problem over signature grids with a bipartite graph $H = (U, V, E)$, where each vertex in U or V is assigned a signature in $\mathcal{F}$ or $\mathcal{G}$, respectively. Signatures in $\mathcal{F}$ are considered as row vectors (or covariant tensors); signatures in $\mathcal{G}$ are considered as column vectors (or contravariant tensors) (see, for example [16]). Let Pl-Holant $(\mathcal{F} \mid \mathcal{G})$ denote the Holant problem over signature grids with a planar bipartite graph.

2.1 Holographic Reductions

One key technique in the study of Holant problems is holographic reductions. To introduce the idea, it is convenient to consider bipartite graphs. For a general graph, we can always transform it into a bipartite graph while preserving the Holant value as follows. For each edge in the graph, we replace it by a path of length two. (This operation is called the *2-stretch* of the graph and yields the edge-vertex incidence graph.) Each new vertex is assigned the binary $\textsc{Equality}$ signature $(=_2) = [1, 0, 1]$. Recall that Holant $(\mathcal{F} \mid \mathcal{G})$ denotes the Holant problem over signature grids with a bipartite graph $H = (U, V, E)$, where each vertex in U or V is assigned a signature in $\mathcal{F}$ or $\mathcal{G}$, respectively. Hence we have that Holant($\mathcal{F}$) $\equiv_T$ Holant $(=_2 \mid \mathcal{F})$.

For a 2-by-2 matrix T and a signature set $\mathcal{F}$, define $T\mathcal{F} = \{g \mid \exists f \in \mathcal{F}$ of arity $n,\ g = T^{\otimes n} f\}$, and similarly for $\mathcal{F}T$. Whenever we write $T^{\otimes n} f$ or $T\mathcal{F}$, we view the signatures as column vectors; similarly for $fT^{\otimes n}$ or $\mathcal{F}T$ as row vectors.

Let T be an invertible 2-by-2 matrix. The holographic transformation defined by T is the following operation: given a signature grid $\Omega = (H, \pi)$ of Holant $(\mathcal{F} \mid \mathcal{G})$, for the same bipartite graph H, we get a new grid $\Omega' = (H, \pi')$ of Holant $\left(\mathcal{F}T \mid T^{-1}\mathcal{G}\right)$ by replacing each signature in $\mathcal{F}$ or $\mathcal{G}$ with the corresponding signature in $\mathcal{F}T$ or $T^{-1}\mathcal{G}$.

Theorem 2.2 (Valiant's Holant Theorem [37])**.** *If $T \in \mathbb{C}^{2\times 2}$ is an invertible matrix, then we have* $\text{Holant}_{\Omega}(\mathcal{F} \mid \mathcal{G}) = \text{Holant}_{\Omega'}(\mathcal{F}T \mid T^{-1}\mathcal{G})$.

Therefore, an invertible holographic transformation does not change the complexity of the Holant problem in the bipartite setting. By Theorem 2.2, we have that

$$\text{Holant}(\mathcal{F}) \equiv \text{Holant}\left([1,0,1]T^{\otimes 2} \mid T^{-1}\mathcal{F}\right)$$

where $T \in \mathbf{GL}_2(\mathbb{C})$ is nonsingular. This leads to the notion of $\mathscr{C}$*-transformable*.

Definition 2.3. *Let $\mathcal{F}$ and $\mathscr{C}$ be two sets of signatures. Say $\mathcal{F}$ is $\mathscr{C}$-transformable if there exists a $T \in \mathbf{GL}_2(\mathbb{C})$ such that $[1,0,1]T^{\otimes 2} \in \mathscr{C}$ and $\mathcal{F} \subseteq T\mathscr{C}$.*

The following lemma is immediate.

Lemma 2.4. *If $\mathcal{F}$ is $\mathscr{C}$-transformable, then we have the following reductions.*

$$\text{Holant}(\mathcal{F}) \leq_T \text{Holant}(\mathscr{C}); \qquad \text{Pl-Holant}(\mathcal{F}) \leq_T \text{Pl-Holant}(\mathscr{C}).$$

A consequence of the lemma above is that, if Holant($\mathscr{C}$) (or Pl-Holant($\mathscr{C}$)) is tractable, then Holant($\mathcal{F}$) (or Pl-Holant($\mathcal{F}$)) is tractable for any $\mathscr{C}$-transformable set $\mathcal{F}$.

2.2 Counting Constraint Satisfaction Problems

An instance of counting constraint satisfaction problems (#CSP($\mathcal{F}$)) has the following bipartite view. We have a set of vertices standing for variables and another set for functions (or constraints). Connect a variable vertex to a constraint vertex if the variable appears in the constraint. This bipartite graph is also known as the *constraint graph*. Moreover, each variable can be viewed as an Equality function, as it forces the same value for all adjacent edges. Under this view, we see that

$$\#\text{CSP}(\mathcal{F}) \equiv_T \text{Holant}\left(\mathcal{EQ} \mid \mathcal{F}\right),$$

where $\mathcal{EQ} = \{=_1, =_2, =_3, \dots\}$ is the set of Equality functions of all arities.

The relationship between #CSP and Holant problems is the following:

$$\begin{aligned} \#\text{CSP}(\mathcal{F}) &\equiv_T \text{Holant}(\mathcal{EQ} \cup \mathcal{F}); \\ \text{Pl-}\#\text{CSP}(\mathcal{F}) &\equiv_T \text{Pl-Holant}(\mathcal{EQ} \cup \mathcal{F}). \end{aligned} \tag{2}$$

Reductions from left to right are trivial. For the other direction, we take a signature grid Ω for the problem on the right and create a bipartite signature grid Ω' for the problem on the left such that both signature grids have the same Holant value. We simply create the equivalent bipartite grid Ω'' of Ω by replace each edge with a path of length 2 with $=_2$ in the middle point, as described earlier. Then we contract all Equality signatures that are connected with each other, resulting in Ω' where Equality signatures are on one side and signatures from $\mathcal{F}$ on the other.

3 Dichotomy Theorems

In this section we survey several classification theorems. We will discuss #CSP first and then turn our attention to Holant problems. We will restrict our attention mostly to symmetric functions throughout this section.

3.1 Counting Constraint Satisfaction Problems

The first classification theorem for #CSP is the dichotomy by Creignou and Hermann [8], which is for unweighted Boolean functions. This dichotomy has been later generalized for real weights [2], and complex weights [15]. Moreover, even beyond the Boolean domain, the complexity of #CSP have been successfully classified. A complete dichotomy theorem for complex weighted functions with any constant domain was obtained after a series of research [3, 17, 4]. However, in the following we will only look at the more relevant Boolean case.

There are two basic tractable cases for complex Boolean #CSP. The first kind is called "*product-type*". A function is of the product-type if it can be decomposed into a product of unary functions, weighted equality functions, and weighted binary disequalities. The algorithm for this case is simple. We just pick an initial assignment and then propagate. The structure of this problem dictates that there are at most 2 assignments with non-zero weights for each component.

The other tractable case is called "*affine-type*". It is a generalization of the fact that we can count the number of solutions to a linear system by computing the rank of the system. (Notice that a linear system can be characterized by parity functions.) However, it is non-trivial to generalize this simple fact to complex functions. When we are dealing with complex weights, there are more potential cancellations that are beneficial to improve the efficiency. For the complete detail, see [15].

The sweeping power of dichotomy theorems is that we know the above two cases are the only tractable cases. Any other problem in this framework is #**P**-hard.

To understand the power of holographic algorithms, we want to consider #CSP defined on planar graphs. The first result on this direction is by Cai, Lu, and Xia [12], and it is later generalized to complex Boolean symmetric functions [21].

The new tractable cases in planar graphs are solved exactly by holographic algorithms. There are two main ingredients in Valiant's holographic algorithm [37]. The overall strategy is to reduce to the FKT algorithm, which counts perfect matchings in planar graphs. The first ingredient is what functions can be expressed by perfect matchings. These functions are named *matchgates*. (For a complete theory of matchgates, see [6].) The second ingredient is holographic transformations, which we have explained in Section 2.1. The crucial observation for #CSP is that, the only transformation that we need to consider is the Hadamard matrix, $\left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]$.

An informal summary of the main theorem in [21] is the following.

Theorem 3.1 (informal)**.** *Let $\mathcal{F}$ be any set of symmetric, complex-valued functions in Boolean variables. Then* Pl-#CSP($\mathcal{F}$) *is* #**P**-*hard unless*

1. *$\mathcal{F}$ is of product-type;*
2. *$\mathcal{F}$ is of affine-type;*
3. *$\mathcal{F}$ belongs to matchgates under $\left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]$.*

In all exceptional cases Pl-#CSP($\mathcal{F}$) *is computable in polynomial time.*

3.2 Holant Problems

Because of the reduction in (2), if #CSP($\mathcal{F}$) is tractable, then Holant($\mathcal{F}$) is as well, since the Holant problem is tractable even for the larger set of functions $\mathcal{EQ} \cup \mathcal{F}$. Thus we see immediately that any tractable cases in Theorem 3.1 translates to tractable cases for Holant problems as well. However, the key feature of Holant problem is the availability of holographic transformations. Indeed, Lemma 2.4 indicates that the "transformable" set of any tractable function set is also tractable.

This is not the only difference between Holant problems and #CSP. The structure of Holant problems allows more cancellation and more potential algorithms. It is indeed the case for Boolean functions, by the discovery of "vanishing" functions [7]. Vanishing functions are constraints, that when applied to any signature grid, produce a zero Holant value. We can not introduce the whole theory here. Instead, we will illustrate the idea by going through examples.

A simple example of vanishing functions is a tensor product of the unary function $(1\ i)$, i.e., a constraint function of the form $(1\ i)^{\otimes k}$ on k variables. The function value is i^t if $0 \le t \le k$ many of the inputs are 1. This function on a vertex (of degree k) can be replaced by k copies of the unary function $(1\ i)$ on k new vertices, each connected to an incident edge. Whenever two copies of $(1\ i)$ meet in the evaluation of Holant_Ω in (1), they annihilate each other since they give the value $(1\ i) \cdot (1\ i) = 0$.

Now consider a function f which is a sum of tensor products of unary functions, where in each product there are more than half $(1\ i)$'s. We view f as in a "superposition" of these tensor products. In a grid composed by f, we may first assign one of the tensor products to each vertex, then evaluate the whole grid. There are exponentially many ways to assign the products, but if we sum over all possible assignments, the Holant value is recovered. On the other hand, in each of these exponentially many terms, there are more than half $(1\ i)$'s and at least two of them meet. It results in making the whole evaluation 0. This argument is valid for any assignment. In summary, we managed to rewrite the Holant sum into a sum of exponentially many terms, each of which is 0. Hence this function is vanishing. In general, it is shown that all vanishing functions are generalizations of the kind we described here [7].

These ghostly vanishing functions are like the elusive dark matter. They do not actually contribute any value to the Holant sum. However in order to give a complete dichotomy for Holant problems, it turns out to be essential that we capture these vanishing functions. There is another similarity with dark matter. Their contribution to the Holant sum is not directly observed. Yet in terms of the dimension of the algebraic variety they constitute, they make up the vast majority of the tractable symmetric functions. Furthermore, when combined with others, they provide a large substrate to produce non-vanishing and tractable function sets.

Similarly to vanishing functions, there is an extra tractable case when we move from planar #CSP to planar Holant problems. The complete description is quite technical and may require several pages. We just briefly explain the idea here. The main cause for the tractability is the structure of planar graphs. We perform some "global" operation to find edges that have to be fixed to 0 or 1. Such edges must exist due to planarity, unless the instance has already fall into one of the known tractable cases. Thus, the strategy is to do this "fixing" move until it is solvable by known algorithms. This is a strange case in that it escapes the usual formulation of holographic algorithms, yet holographic algorithms are the main non-trivial ingredient once the "fixing" is done.

Informally, the main theorems of [7, 5] can be summarized as follows.

Theorem 3.2 (informal)**.** *Let $\mathcal{F}$ be any set of symmetric, complex-valued functions in Boolean variables. Then* Pl-Holant($\mathcal{F}$) *is* #**P**-*hard unless*

1. *$\mathcal{F}$ is transformable to product-type functions;*
2. *$\mathcal{F}$ is transformable to affine-type functions;*
3. *$\mathcal{F}$ is tractable due to vanishing functions;*
4. *$\mathcal{F}$ is transformable to matchgates;*
5. *$\mathcal{F}$ belongs to the extra planar tractable case.*

In all exceptional cases Pl-Holant($\mathcal{F}$) *is computable in polynomial time. If $\mathcal{F}$ belongs to cases 1,2,3, then* Holant($\mathcal{F}$) *is tractable, and otherwise* Holant($\mathcal{F}$) *is* #**P**-*hard.*

4 Holographic Transformations

Holographic transformations are the central technique to obtain Theorem 3.1 and Theorem 3.2. In this last section, let us see some holographic transformations in action.

4.1 Ising model and the "subgraphs" world

We first review a classical equivalence between the Ising model and even subgraphs. It was observed as early as in [38] and has been rediscovered several times. In the seminal approximation algorithm for the Ising model [24], this equivalence played a central role. We show that it can be easily obtained using the "modern" language of holographic transformations.

The partition function of a ferromagnetic Ising model on a graph $G = (V, E)$ with parameter β is defined as

$$Z_{Ising}(\beta) = \sum_{\sigma \in \{0,1\}^V} \beta^{m(\sigma)}.$$

where $\sigma \in \{0, 1\}^V$ is an assignment of vertices and $m(\sigma)$ is the number of monochromatic edges (either (0,0) or (1,1)) under σ. On the other hand, a subgraph $S \subseteq E$ is called *even* if every vertex in the induced subgraph (V, S) has an even degree. Denote by Ω_{even} the state space of all such even subgraphs of G. Define the partition function with parameter p:

$$Z_{even}(p) = \sum_{S \in \Omega_{even}} p^{|S|}(1 - p)^{|E \setminus S|}.$$

We will show that

$$Z_{Ising}(\beta) = 2^{|V|}\beta^{|E|} Z_{even}\left(\frac{1}{2}\left(1 - \frac{1}{\beta}\right)\right). \tag{3}$$

Let Ising be a binary function so that

$$\textsc{Ising}(x_1, x_2) := \begin{cases} \beta & \text{if } x_1 = x_2; \\ 1 & \text{otherwise.} \end{cases}$$

Using our notation, Ising is $[\beta, 1, \beta]$. Then we may express Z_{Ising} as Equality functions on vertices and the Ising function on edges; namely, Holant($\mathcal{EQ}$ | Ising).

The even subgraph constraint can be formulated as choosing a subset of edges subject to the following Even constraint on vertices (of degree d):

$$\textsc{Even}_d(x_1, \cdots, x_d) := \begin{cases} 1 & \text{if } \bigoplus_i x_i = 0; \\ 0 & \text{otherwise.} \end{cases}$$

Choosing edges can be interpreted as binary EQUALITY functions $[1, 0, 1]$ on edges (cf. Section 2.2). Moreover, the parameter p makes these functions weighted. Define WEQ as the binary weighted Equality function $[1 - p, 0, p]$. Then Z_{Even} is just $\text{Holant}\,(\text{EVEN}_1, \text{EVEN}_2, \cdots \mid \text{WEQ})$. To recover (3), set $p = \frac{1}{2}\left(1 - \frac{1}{\beta}\right)$.

To see the equivalence (3), do a holographic transformation by $H = \left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]$:

$$\text{Holant}\,(\mathcal{EQ} \mid \text{ISING}) \equiv \text{Holant}\left(\mathcal{EQ}H \mid (H^{-1})^{\otimes 2}\text{ISING}\right).$$

Now we just need to verify

$$(=_d)H^{\otimes d} = 2\text{EVEN}_d;$$
$$(H^{-1})^{\otimes 2}\text{ISING} = \beta\text{WEQ}.$$

The first line can be verified as

$$\begin{aligned}(=_d)H^{\otimes d} &= \left([1, 0]^{\otimes d} + [0, 1]^{\otimes d}\right)\left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]^{\otimes d} \\ &= [1, 1]^{\otimes d} + [1, -1]^{\otimes d} \\ &= [2, 0, 2, 0, \dots] = 2\text{EVEN}_d.\end{aligned}$$

Noticing that $H^{-1} = \frac{1}{2}\left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]$, the second line can be verified as

$$\begin{aligned}(H^{-1})^{\otimes 2}\text{ISING} &= \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}\begin{pmatrix} \beta \\ 1 \\ 1 \\ \beta \end{pmatrix} = \frac{1}{4}\begin{pmatrix} 2\beta + 2 \\ 0 \\ 0 \\ 2\beta - 2 \end{pmatrix} \\ &= \beta\left[\frac{1}{2}\left(1 + \frac{1}{\beta}\right), 0, \frac{1}{2}\left(1 - \frac{1}{\beta}\right)\right] \\ &= \beta[1 - p, 0, p] = \beta\text{WEQ}.\end{aligned}$$

4.2 Shearer's Condition

Lovász local lemma [18] is an important tool in combinatorics. Shearer [29] gave the optimal condition for the Lovász local lemma on a fixed dependency graph $G = (V, E)$. Verifying Shearer's condition is a non-trivial task and usually boils down to deciding whether the independence polynomial is positive for G, where each vertex has a negative weight. Let $\mathbf{p} = (p_v)_{v \in V}$ be the set of weights. Let $\mathcal{I}$ be the collection of independent sets of G. Define the following quantity:

$$q(\mathbf{p}) := \sum_{I \in \mathcal{I}} (-1)^{|I|} \prod_{v \in I} p_v.$$

We are interested in whether $q(\mathbf{p})$ is positive or not.

Here we discuss a particular example such that $p_v = 2^{-d_v}$ where d_v is the degree of vertex v.[3] We will see that $q(\mathbf{p}) > 0$ if and only if G is a tree. Holographic transformations will play a central role in the proof below.

We first claim that $q(\mathbf{p})$ is equivalent to

$$\text{Holant}\,([1, 1/2, 0] \mid [1, 0, \ldots, 0, -1])\,.$$

In this bipartite view, variables are half-edges of the graph. If v is chosen in the independent set, then all of its adjacent half-edges are assigned 1. Since we are taking an independent sets, we cannot take both endpoints simultaneously. This is why the edge function forbids $(1, 1)$. The weight of 2^{-d_v} can be viewed as distributed to edges adjacent to v, where each edge gets $1/2$. If a vertex is chosen, it also contributes an extra -1 factor to $q(\mathbf{p})$. Hence the Equality function on the right hand side is weighted.

Let $H = H^{-1} = \frac{1}{\sqrt{2}} \left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]$ be the Hadamard matrix. Then we can do the holographic transformation by H.

$$\begin{aligned}
&\text{Holant}\,([1, 1/2, 0] \mid [1, 0, \ldots, 0, -1]) \\
\equiv\, &\text{Holant}\left([1, 1/2, 0](H^{-1})^{\otimes 2} \mid H^{\otimes d_v}[1, 0, \ldots, 0, -1]\right) \\
\equiv\, &\text{Holant}\left([1, 1/2, 0](H^{-1})^{\otimes 2} \mid H^{\otimes d_v}[1, 0, \ldots, 0, -1]\right) \\
\equiv\, &\text{Holant}\left([1, 1/2, 0] \mid 2^{1-d_v/2}\textsc{Odd}_{d_v}\right),
\end{aligned}$$

where $\textsc{Odd}_{d_v}$ denotes the function of arity d_v that is satisfied only if an odd number of input variables is 1. This can be easily verified as

$$\begin{pmatrix} 1 & 1/2 & 1/2 & 0 \end{pmatrix} \cdot \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1/2 & 1/2 & 0 \end{pmatrix},$$

and

$$\begin{aligned}
H^{\otimes d_v}[1, 0, \ldots, 0, -1] &= 2^{-d_v/2} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}^{\otimes d_v} \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix}^{\otimes d_v} - \begin{pmatrix} 0 \\ 1 \end{pmatrix}^{\otimes d_v} \right) \\
&= 2^{-d_v/2} \left(\begin{pmatrix} 1 \\ 1 \end{pmatrix}^{\otimes d_v} - \begin{pmatrix} 1 \\ -1 \end{pmatrix}^{\otimes d_v} \right) = 2^{1-d_v/2}\textsc{Odd}_{d_v}.
\end{aligned}$$

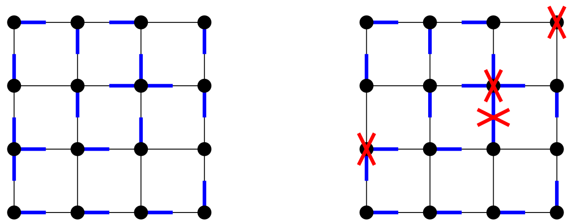
Notice that all functions in $\text{Holant}\,([1, 1/2, 0] \mid \textsc{Odd}_{d_v})$ are non-negative. This implies that $q(\mathbf{p}) \geq 0$ for any graph G.

[3]For those who are interested, this is the setting of sink-free orientations. Random variables are directions of edges and sinks are "bad" events.

Let Holant(G) denote the partition function of Holant $([1, 1/2, 0] \mid \textsc{Odd}_{d_v})$ on a graph G. Due to the discussion above, we have that

$$q(\mathbf{p}) = \prod_{v \in V} 2^{1-d_v/2}\,\text{Holant}(G) = 2^{n-m}\,\text{Holant}(G), \tag{4}$$

where $n = |V|$ and $m = |E|$.



(a) A satisfying assignment (b) An unsatisfying assignment

Figure 1: An example of the constraint satisfaction problem on G. Blue half-edges are assigned 1. Red crosses mark constraints that are not satisfied.

Consider the following constraint satisfaction problem on the graph G. Assign each half-edge 0 or 1, but not both 1. Denote the assignment by σ. For each vertex v, we require odd number of adjacent half-edges assigned 1. An example can be found in Figure 1. Let S be the set of satisfying assignments. It is easy to see that

$$\text{Holant}(G) = \sum_{\sigma \in S} 2^{-|\sigma|}, \tag{5}$$

where $|\sigma|$ is the number of half-edges assigned 1.

Theorem 4.1. *Let G be a connected graph. If G is a tree, then $q(\boldsymbol{p}) = 0$; otherwise $q(\boldsymbol{p}) > 0$.*

Proof. By (4) and (5), $q(\mathbf{p}) > 0$ if and only if S is not empty.

If G is a tree, then $S = \emptyset$. To see this, choose an arbitrary vertex v as the root of G. All leaves of G has degree 1. Hence their adjacent half-edges must be assigned 1. Remove all leaves. Again the new leaves also force their adjacent half-edges to be 1. Repeat this procedure until we have only the root v left. Then there is no assignment to satisfy the odd parity constraint of v.

Otherwise G is not a tree, and there exists a cycle C in G. We will construct a satisfying assignment. Pick an arbitrary half-edge in C and assign 1 to it. Then we

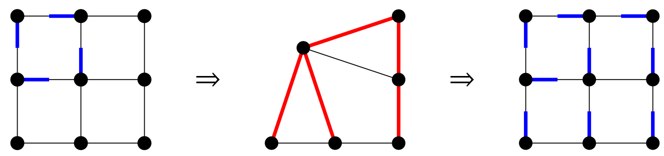


Figure 2: An example of constructing the constraint satisfaction problem on a 3-by-3 grid: (1) choose the top-left cycle and contract; (2) find a spanning tree of the new graph (red); (3) map the assignment back to the original.

may follow C to assign 1's so that every vertex in C has exactly one adjacent half-edge chosen. Contract all vertices on C to one vertex v and get a new graph G'. Pick an arbitrary spanning tree T of G'. Similarly to the last case, we may assign 1 to half-edges adjacent to leaves recursively until v is left. Edges that are not in T are assigned $(0, 0)$ to its half-edges. This gives an assignment of all edges in G. It is not hard to verify that in this assignment, all vertices are adjacent to exactly one chosen half-edge. Hence this is a satisfying assignment and $q(\mathbf{p}) > 0$. $\square$

References

[1] R.J. Baxter. *Exactly solved models in statistical mechanics*. Academic press London, 1982.

[2] A. Bulatov, M. Dyer, L.A. Goldberg, M. Jalsenius, and D. Richerby. The complexity of weighted boolean #CSP with mixed signs. *Theoretical Computer Science*, 410(38-40):3949–3961, 2009.

[3] Andrei A. Bulatov. The complexity of the counting constraint satisfaction problem. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *ICALP (1)*, volume 5125 of *Lecture Notes in Computer Science*, pages 646–661. Springer, 2008.

[4] Jin-Yi Cai and Xi Chen. Complexity of counting CSP with complex weights. In *STOC*, pages 909–920, 2012.

[5] Jin-Yi Cai, Zhiguo Fu, Heng Guo, and Tyson Williams. A Holant dichotomy: Is the FKT algorithm universal? In *FOCS*, pages 249–260, 2015.

[6] Jin-Yi Cai and Aaron Gorenstein. Matchgates revisited. *Theory Comput.*, 10(7):167–197, 2014.

[7] Jin-Yi Cai, Heng Guo, and Tyson Williams. A complete dichotomy rises from the capture of vanishing signatures. In *STOC*, pages 249–260, 2013.

[8] Nadia Creignou and Miki Hermann. Complexity of generalized satisfiability counting problems. *Inf. Comput.*, 125(1):1–12, 1996.

[9] Jin-Yi Cai and Michael Kowalczyk. Spin systems on k-regular graphs with complex edge functions. *Theor. Comput. Sci.*, 461:2–16, 2012.

[10] Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Holographic algorithms by Fibonacci gates and holographic reductions for hardness. In *FOCS*, pages 644–653, 2008.

[11] Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Holant problems and counting CSP. In Michael Mitzenmacher, editor, *STOC*, pages 715–724. ACM, 2009.

[12] Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Holographic algorithms with matchgates capture precisely tractable planar #CSP. In *FOCS*, pages 427–436, 2010.

[13] Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Computational complexity of Holant problems. *SIAM J. Comput.*, 40(4):1101–1132, 2011.

[14] Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Holographic reduction, interpolation and hardness. *Computational Complexity*, 21(4):573–604, 2012.

[15] Jin-Yi Cai, Pinyan Lu, and Mingji Xia. The complexity of complex weighted Boolean #CSP. *J. Comput. System Sci.*, 80(1):217–236, 2014.

[16] C. T. J. Dodson and T. Poston. *Tensor Geometry*. Graduate Texts in Mathematics 130. Springer-Verlag, New York, 1991.

[17] Martin E. Dyer and David Richerby. An effective dichotomy for the counting constraint satisfaction problem. *SIAM J. Comput.*, 42(3):1245–1274, 2013.

[18] Paul Erdős and László Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. *Infinite and finite sets, volume 10 of Colloquia Mathematica Societatis János Bolyai*, pages 609–628, 1975.

[19] M. Freedman, L. Lovász, and A. Schrijver. Reflection positivity, rank connectivity, and homomorphism of graphs. *J. AMS*, 20:37–51, 2007.

[20] Heng Guo. *Complexity Classification of Exact and Approximate Counting Problems*. PhD thesis, University of Wisconsin-Madison, 2015.

[21] Heng Guo and Tyson Williams. The complexity of planar Boolean #CSP with complex weights. In *ICALP*, pages 516–527, 2013.

[22] E. Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik A Hadrons and Nuclei*, 31(1):253–258, 1925.

[23] Mark Jerrum and Alistair Sinclair. Approximating the permanent. *SIAM J. Comput.*, 18(6):1149–1178, 1989.

[24] Mark Jerrum and Alistair Sinclair. Polynomial-time approximation algorithms for the ising model. *SIAM Journal on Computing*, 22(5):1087–1116, 1993.

[25] P. W. Kasteleyn. The statistics of dimers on a lattice. *Physica*, 27:1209–1225, 1961.

[26] P. W. Kasteleyn. Graph theory and crystal physics. In (F. Harary, editor, *Graph Theory and Theoretical Physics*, pages 43–110. Academic Press, London, 1967.

[27] T.D. Lee and C.N. Yang. Statistical theory of equations of state and phase transitions. II. Lattice gas and Ising model. *Physical Review*, 87(3):410–419, 1952.

[28] L. Onsager. Crystal statistics. I. A two-dimensional model with an order-disorder transition. *Physical Review*, 65(3-4):117–149, 1944.

[29] James B. Shearer. On a problem of Spencer. *Combinatorica*, 5(3):241–245, 1985.

[30] H. N. V. Temperley and M. E. Fisher. Dimer problem in statistical mechanics-an exact result. *Philosophical Magazine*, 6:1061–1063, 1961.

[31] Seinosuke Toda. PP is as hard as the polynomial-time hierarchy. *SIAM J. Comput.*, 20(5):865–877, 1991.

[32] Leslie G. Valiant. The complexity of computing the permanent. *Theor. Comput. Sci.*, 8:189–201, 1979.

[33] Leslie G. Valiant. The complexity of enumeration and reliability problems. *SIAM J. Comput.*, 8(3):410–421, 1979.

[34] Leslie G. Valiant. Expressiveness of matchgates. *Theor. Comput. Sci.*, 289(1):457–471, 2002.

[35] Leslie G. Valiant. Quantum circuits that can be simulated classically in polynomial time. *SIAM J. Comput.*, 31(4):1229–1254, 2002.

[36] Leslie G. Valiant. Accidental algorthims. In *FOCS*, pages 509–517, 2006.

[37] Leslie G. Valiant. Holographic algorithms. *SIAM J. Comput.*, 37(5):1565–1594, 2008.

[38] Bartel L. van der Waerden. Die lange Reichweite der regelmäßigen Atomanordnung in Mischkristallen. *Zeitschrift für Physik*, 118(7):473–488, 1941.

[39] Dirk Vertigan. The computational complexity of Tutte invariants for planar graphs. *SIAM J. Comput.*, 35(3):690–712, 2005.

MONOIDS AS STORAGE MECHANISMS

Georg Zetzsche
Technische Universität Kaiserslautern,
Germany

An important theme of automata theory is to study mathematical models of computing devices with regard to what behavior they can exhibit and what we can infer about such a device when given a description.

These two types of questions each have their own motivation. The first type addresses *expressiveness*. This aspect is important to understand because it explains what we can compute with limited resources and what systems we can describe with the respective models. The second type of questions explores the *analyzability* of models. This perspective is instrumental when we want to algorithmically verify properties of systems, which, due to the advent of increasingly complex and concurrent systems, has become a task of significant importance.

The perspectives of expressiveness and analyzability are deeply intertwined: They are conflicting qualities insofar as the more expressive a model is, the more difficult it usually is to analyze. For these reasons, it has become a strong driving force of today's research in theoretical computer science to understand how we can provide models that are expressive enough for a given type of systems and yet are simple enough to be amenable to analysis. This thesis contributes by studying the relationship between the computational properties of automata with storage and the employed storage mechanism.

Automata with storage In a tradition initiated by Turing in the introduction of the eponymous machine, automata theory yielded a rich variety of models that comprise a finite-state control and a potentially infinite data repository. The models are obtained by imposing restrictions on how the data can be stored, manipulated, and retrieved, while permitting arbitrary use of the finite-state control.

In terms of hard- and software systems that can be represented by such models, this means we can precisely reflect arbitrary control flows, but we abstract from certain aspects of data access. For example, pushdown automata can correctly imitate the control flow and calling stack of a recursive program, but heap memory cannot be represented. A form of data respository, together with the permitted modes of access, is called a *storage mechanism*. Examples of storage mechanisms include Turing machine tapes, pushdown storages, and various kinds of counters.

Instead of investigating the properties of an individual model of computation, the present work attempts to provide general insights into how expressiveness and analyzability of a model of computation are affected by the structure of the storage mechanism. To this end, it presents generalizations of results about concrete storage mechanisms to larger classes of storage mechanisms. These generalizations will characterize those storage mechanisms for which the given result remains true and for which it fails.

Storage mechanisms as monoids In order to speak of classes of storage mechanisms, we need an overarching framework that accommodates each of the concrete storage mechanisms we wish to address. Such a framework is provided by interpreting storage mechanisms as *monoids*.

Suppose a storage mechanism consists of a (potentially infinite) set of states, a finite set of functions representing its available operations, an initial state, and a collection of valid final states. To account for operations that are not always applicable, such as a pop operation for a stack symbol that is not currently at the top, the functions can be partial functions. For example, a pushdown storage with stack alphabet Γ consists of the set Γ^* as its set of states, the operations push_a and pop_a for each $a \in \Gamma$, and the empty word ε as its initial state and its final state (assuming that it accepts with an empty stack). As partial functions, the operations push_a and pop_a are defined as

$$\mathsf{push}_a \colon \Gamma^* \to \Gamma^*, \quad w \mapsto wa \qquad\qquad \mathsf{pop}_a \colon \Gamma^* \nrightarrow \Gamma^*, \quad wa \mapsto w.$$

(here, we denote partial functions by $\nrightarrow$). Note that pop_a is defined on precisely those words that end in a.

Another example is the Minsky counter, which has $\mathbb{N}$, the set of natural numbers, as its set of states and has inc (*increment*), dec (*decrement*), and zero (*zero test*) as its operations:

$$\mathsf{inc} \colon \mathbb{N} \to \mathbb{N}, \quad n \mapsto n+1, \qquad \mathsf{dec} \colon \mathbb{N} \nrightarrow \mathbb{N}, \quad n \mapsto n-1, \qquad \mathsf{zero} \colon \mathbb{N} \nrightarrow \mathbb{N}, \quad 0 \mapsto 0.$$

Note that here, the decrement operation is undefined for state 0 and the zero test operation is defined only in state 0.

To such a storage mechanism, we can associate the monoid of all compositions of available operations. Let us examine what this yields in the case of a pushdown store as above. If we compose push_a and pop_b for $a \neq b$, we obtain the function 0, which is defined nowhere: After pushing an a, popping b cannot be defined. Moreover, composing 0 with any other operation yields 0 again. If, however, we only consider compositions where such incompatible push and pop do not occur, the reader can verify that we always get functions of the form $P_{u,v}$ for $u, v \in \Gamma^*$, where

$$\begin{aligned} P_{u,v} \colon \Gamma^* &\rightharpoonup \Gamma^*, \\ wu &\mapsto wv, \end{aligned}$$

is defined on precisely those words with suffix u. Therefore, the resulting monoid has the elements $\{0\} \cup \{P_{u,v} \mid u, v \in \Gamma^*\}$.

Let us consider the case of a Minsky counter. Any composition of just the increment and decrement operations yields an element $C_{r,s}$ such that

$$\begin{aligned} C_{r,s} \colon \mathbb{N} &\rightharpoonup \mathbb{N}, \\ n + r &\mapsto n + s, \end{aligned}$$

which is defined on all numbers $\geq r$. If the composition involves a zero test, then it is either 0 as above or it is defined on only one element $r \in \mathbb{N}$ and of the form $D_{r,s}$, for which

$$\begin{aligned} D_{r,s} \colon \mathbb{N} &\rightharpoonup \mathbb{N}, \\ r &\mapsto s. \end{aligned}$$

Hence, the corresponding monoid comprises the set $\{0, C_{r,s}, D_{r,s} \mid r, s \in \mathbb{N}\}$.

Monoids as storage mechanisms The advantage of interpreting storage mechanisms as monoids is that we can go in the other direction and interpret *monoids as storage mechanisms*: The elements of the monoid determine the set of states as well as the set of operations and the identity element is the final state. This allows us to use algebraic constructions to synthesize similar storage mechanisms and thus identify *what structural traits of the mechanism are responsible for which computational properties*. For example, we will define monoids by graphs that may contain self-loops. We will then see that graphs with no self-loops or edges correspond to pushdown storages. If the graph has no self-loops, but is otherwise a clique, it is equivalent to counters without zero tests (that cannot go below zero). This is usually called a set of *partially blind counters*. Moreover, if the graph is a clique and has self-loops everywhere, we obtain counters that can go below zero and are only zero tested in the end of the computation, hence a collection of *blind counters*.

This means we can regard these individual storage as points on a spectrum and examine where exactly the computational properties remain true and where they cease to hold. For example, it is known that automata with a pushdown or with blind counters accept languages with semilinear Parikh images, which is not true of partially blind counters. We can now study which graphs exactly guarantee semilinearity of the accepted languages.

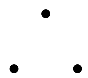
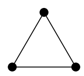
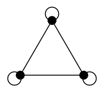
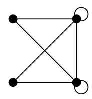
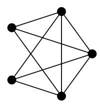
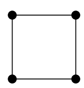
Graph Γ	Monoid $\mathbb{M}\Gamma$	Storage mechanism
	$\mathbb{B}^{(3)}$	Pushdown (with three symbols)
	$\mathbb{B}^3$	Three partially blind counters
	$\mathbb{Z}^3$	Three blind counters
	$\mathbb{B}^{(2)} \times \mathbb{Z}^2$	Pushdown (with two symbols) and two blind counters
	$\mathbb{B}^{(2)} \times \mathbb{B}^3$	Pushdown and three partially blind counters
	$\mathbb{B}^{(2)} \times \mathbb{B}^{(2)}$	Two pushdowns (with two symbols each)

Table 1: Examples of storage mechanisms

Valence automata We investigate monoids as storage mechanisms by deploying them in the framework of valence automata. A valence automaton over a monoid M is a finite automaton in which each edge carries an input word and an element of M. Let us define this formally. Let M be a monoid. A *valence automaton over M* is a tuple $A = (Q, X, M, E, q_0, F)$, where

- Q is a finite set of *states*,
- X is an alphabet, called its *input alphabet*,
- $E \subseteq Q \times X^* \times M \times Q$ is a finite set of *edges* or *transitions*,
- $q_0 \in Q$ is its *initial state*, and

- $F \subseteq Q$ is its set of *final states*.

For triples (q, u, x) and (q', u', x') from $Q \times X^* \times M$, let

$$(q, u, x) \to_A (q', u', x') \qquad \text{if } u' = uw \text{ and } x' = xz \text{ for some } (q, w, z, q') \in E.$$

In this work, we measure the expressive power of valence automata by the class of languages they can accept. The *language accepted by A* is defined as

$$L(A) = \{w \in X^* \mid (q_0, \varepsilon, 1) \to_A^* (f, w, 1) \text{ for some } f \in F\}.$$

In other words, the automaton accepts all words that label paths from an initial state to a final state such that the product of the monoid elements is the identity.

Valence automata are not a new concept and have been studied before by several authors from various perspectives. What distinguishes this work from earlier ones is that it systematically generalizes results for individual models of automata with storage. Specifically, for each of a series of results about concrete storage mechanisms, it presents a broader class of monoids and identifies those members of the class to which the result carries over.

As a rich source of monoids to represent well-known storage mechanims, this work also introduces *graph monoids*. They are defined by graphs, which often allows us to relate the combinatorial properties of the graphs with the computational properties of the resulting storage mechanisms.

Main contributions

The following are the main contributions of this work.

Graph monoids Oftentimes, characterizing *all* monoids with a given computational property is infeasible; or it would result in a mere reformulation of the property and thus not be meaningful.

Therefore, it is useful to have a class of monoids that is large enough to accomodate storage mechanisms from the literature, but small enough to permit meaningful characterizations. To this end, this work introduces the class of *graph monoids*, which are defined by graphs. Let us sketch their definition. Suppose we have an undirected graph $\Gamma = (V, E)$ that may have self-loops. This means, $E \subseteq \{S \subseteq V \mid |S| \leq 2\}$. For brevity, we call a vertex *looped* if it has a self-loop; otherwise we call it *unlooped*. We define the alphabet $X_\Gamma = \{a_v, \bar{a}_v \mid v \in V\}$ and consider the smallest congruence $\equiv_\Gamma$ on X_Γ^* with

$$a_v \bar{a}_v \equiv_\Gamma \varepsilon \qquad \text{for each } v \in V, \text{ and} \tag{1}$$

$$xy \equiv_\Gamma yx \qquad \text{for each } x \in \{a_v, \bar{a}_v\}, \; y \in \{a_w, \bar{a}_w\} \text{ with } \{v, w\} \in E. \tag{2}$$

In other words, for each vertex $v \in V$, we have a positive element a_v and a negative element $\bar{a}_v$. Then, (1) means that $a_v\bar{a}_v$ cancels to the identity and (2) tells us that two elements (whether they are positive or negative) may commute if their vertices are adjacent. The resulting monoid is now defined as

$$\mathbb{M}\Gamma \quad = \quad X_\Gamma^*/{\equiv_\Gamma}.$$

It is not hard to see that by choosing appropriate graphs, one can realize pushdown automata, partially blind counter automata, blind counter automata, and Turing machines, but also various combinations thereof.

For example, suppose Γ contains just one unlooped vertex. Then $X_\Gamma = \{a_v, \bar{a}_v\}$ and $w \equiv_\Gamma \varepsilon$ if and only if $|w|_{a_v} = |w|_{\bar{a}_v}$ and $|p|_{a_v} \geq |p|_{\bar{a}_v}$ for every prefix p of w. In other words, if we interpret a_v and $\bar{a}_v$ as *increment* and *decrement*, respectively, then this means w is a sequence of counter actions that leads from 0 to 0 and keeps the counter non-negative. Since $[w] = 1$ if and only if $w \equiv_\Gamma \varepsilon$, this means $\mathbb{M}\Gamma$ represents a partially blind counter.

Furthermore, if Γ consists of one looped vertex, then $w \equiv_\Gamma \varepsilon$ if and only if $|w|_{a_v} = |w|_{\bar{a}_v}$. Hence, $\mathbb{M}\Gamma$ represents a blind counter.

Suppose Γ consists of two vertices x, y without any self-loops. If we then for $z \in \{x, y\}$, interpret a_z as a push_z operation and $\bar{a}_z$ as a pop_z operation, then $w \equiv_\Gamma \varepsilon$ if and only if w transforms the empty stack into the empty stack. Thus, $\mathbb{M}\Gamma$ realizes a pushdown store with a two-letter stack alphabet.

Moreover, if Γ is obtained from the disjoint graphs Γ_0 and Γ_1 by adding edges between any vertex from Γ_0 and any vertex from Γ_1, then $w \equiv_\Gamma \varepsilon$ if and only if $w_i \equiv_{\Gamma_i} \varepsilon$ for $i \in \{0, 1\}$, where w_i is obtained from w by deleting all symbols corresponding to letters from Γ_{1-i}. Hence, $\mathbb{M}\Gamma$ allows us to use the storage mechanisms realized by $\mathbb{M}\Gamma_0$ and $\mathbb{M}\Gamma_1$ independently.

Table 1 presents examples of graphs and the corresponding storage mechanisms. Here, the symbol $\mathbb{B}$ denotes the monoid $\mathbb{M}\Gamma$ where Γ consists of one unlooped vertex. Moreover, $M^{(n)}$ denotes the n-fold free product of the monoid M. Analogously, M^n is the n-fold direct product.

Increasing expressiveness We present an algebraic characterization of those monoids that *increase the expressiveness* in the following sense: Without the storage mechanism, finite automata only accept regular languages. Hence, we describe those monoids M for which valence automata over M can accept non-regular languages. In fact, we show that this also characterizes those monoids for which deterministic valence automata are expressively weaker and those for which valence grammars can generate non-context-free languages. Valence grammars are a concept related to valence automata and equip context-free grammars with a monoid storage. While the characterization of monoids that increase expressiveness in valence automata has been obtained independently by Render [17] in her thesis, the latter characterization for valence grammars answers an open question raised by Fernau and Stiebe [3].

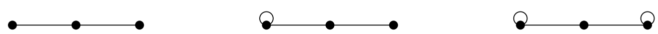
These results have been published in [20].

Emptiness problem Afterwards, we turn to the decidability of the *emptiness problem*. This is a type of reachability problem and one of the most basic problems in the algorithmic analysis of system models. Therefore, we are interested in which storage mechanisms permit a decision procedure.

There is a serious obstacle to a complete characterization: Using graph monoids, one can realize a pushdown storage with partially blind counters (see Table 1), for which the deciability of the emptiness problem remains a long-standing open question [11, 16].

However, if we forbid the subgraphs corresponding to these mechanisms, we can characterize those with a decidable emptiness problem. The result generalizes the decidability for pushdown automata and for partially blind counter automata (or equivalently, Petri nets).

The complete statement requires some terminology. A graph Γ is said to be a *PPN-graph* if it is isomorphic to one of the following three graphs:



One can show that every such storage mechanisms allows us to simulate automata with a pushdown and one partially blind counter: PPN stands for *pushdown Petri net*. A graph Γ is called *PPN-free* if it has no PPN-graph as an induced subgraph. The *comparability graph* of a tree t is a simple graph with the same vertices as t, but has an edge between two vertices whenever one is a descendant of the other in t. A simple graph is a *transitive forest* if it is the disjoint union of comparability graphs of trees. Let Γ^- denote the graph obtained from Γ by deleting all self-loops.

Theorem 1. *Let Γ be PPN-free. Then the following conditions are equivalent:*

1. *Emptiness is decidable for valence automata over $\mathbb{M}\Gamma$.*

2. *Γ^- is a transitive forest.*

Moreover, we present an intuitive, more mechanical, description of (a) the mechanisms shown to be decidable and (b) the storage mechanisms where decidability remains open.

These mechanisms instances of *stacked counters*. Stacked counter storage mechanisms are obtained by alternating two transformations of storage mechanisms: *building stacks* (of configurations of an existing mechanism) and *adding counters* (to an existing mechanism). Building stacks works as follows: Given one storage mechanism, we construct a new one whose configurations are stacks (i.e. sequences) of configurations of the old one. During a computation, one can then start a new entry, manipulate the topmost entry (as prescribed by the old mechanism) and pop the topmost entry if empty[1]. On the level of monoids, this corresponds to transforming M into $\mathbb{B} * M$. Adding counters is a simpler transformation: In the new mechanism, we have a counter in addition to the old mechanism. On the monoid level, this means we turn M into $\mathbb{Z} \times M$ (*adding a blind counter*) or into $\mathbb{B} \times M$ (*adding a partially blind counter*).

The mechanisms of (a) are obtained from partially blind counters by *building stacks* and *adding blind counters*. Formally, $\mathsf{SC}^{\pm}$ is the smallest class of monoids that contains $\mathbb{B}^n$ for every $n \geq 0$ and has the property that if M belongs to $\mathsf{SC}^{\pm}$, then both $\mathbb{B} * M$ and $\mathbb{Z} \times M$ belong to $\mathsf{SC}^{\pm}$ as well. Note that the monoids in $\mathsf{SC}^{\pm}$ are not precisely those satisfying the conditions of Theorem 1, but they are expressively equivalent and serve to provide an intuition.

The mechanisms of (b) are defined similarly: They are obtained from partially blind counters by *building stacks* and *adding partially blind counters*. As monoids, these mechanisms are represented by the class SC^{+}, which is the smallest class containing $\mathbb{B}^n$ for $n \geq 0$ such that if M belongs to SC^{+}, we also have $\mathbb{B} * M$ and $\mathbb{B} \times M$ in SC^{+}. Again, these mechanisms are expressively equivalent to those where Theorem 1 leaves the decidability status open.

The mechanisms corresponding to SC^{+} are a natural generalization of Reinhardt's priority counter machines but also of pushdown storages with partially blind counters. In particular, they are a promising candidate for a quite powerful model where reachability might still be decidable.

Theorem 1 extends a result of Lohrey and Steinberg [13], which characterizes those graph groups with a decidable rational subset membership problem. Where Lohrey and Steinberg rely on semilinearity arguments, we use a reduction to the reachability problem of priority multicounter machines, which has been proven decidable by Reinhardt [16].

This result has been published in [22].

Boolean closure We are also concerned with closure properties of the languages accepted by valence automata. Since it is well-known that the regular languages are closed under the Boolean operations (union, intersection, and complementation), we ask for which monoids M, the class of languages accepted by valence automata over M is *closed under the Boolean operations*.

Aside from understanding closure properties of automata models, this question

[1]Note that this is akin to (but not quite the same as) the step from order-n pushdowns to order-$(n+1)$ pushdowns. However, in contrast to higher-order pushdown automata, there is no operation to copy the topmost entry.

is relevant to the decidability of the first-order theory of structures: Identifying new monoids that admit these closure properties and decidability of the emptiness problem would yield an extended notion of automatic structures [10], whose first-order theory would be decidable.

Our result is a rather negative answer and goes beyond valence automata. It is shown here that every language class that is closed under the Boolean operations and rational transductions and contains an *arbitrary* non-regular language already includes the whole arithmetical hierarchy. The crucial idea is an encoding of counter values of a Minsky machine by Myhill-Nerode classes of the non-regular language.

It follows in particular that every language class induced by valence automata beyond the regular languages either fails to be closed under the Boolean operations or lacks virtually all decidability properties.

This result has been published in [14, 23].

Context-freeness We compare the expressiveness of storage mechanisms with that of context-free grammars. Specifically, we ask which monoids cause valence automata to only accept context-free languages. We characterize those graph products M of monoids for which valence automata over M accept only context-free languages. This means, in particular, that we extend a group-theoretic result of Lohrey and Sénizergues [12], which characterizes those graph products of groups where the resulting group is virtually free.

This result has been published in [1].

Semilinearity We study generalizations of Parikh's Theorem [15], which states that the Parikh image of each context-free language is semilinear. This result is an extraordinarily useful tool, both for proving non-expressibility result and in the algorithmic analysis of formal languages. It has been extended to so many other language classes that the term 'a Parikh theorem' has come to mean a statement guaranteeing effective semilinearity. This type of results has countless applications. Especially in cooperation with Presburger arithmetic, it facilitates a number of decision procedures.

Therefore, understanding what storage mechanisms admit a Parikh theorem is useful for clarifying expressiveness, but especially in order to analyze automata. Hence, we study which monoids guarantee semilinearity of the accepted language class. The first presented result is a characterization of those graph monoids that guarantee semilinear Parikh images. As explained above, this generalizes the semilinearity results for pushdown automata and blind multicounter automata.

A *looped clique* is a graph where every vertex is looped and any two vertices are adjacent.

Theorem 2. *Valence automata over $\mathbb{M}\Gamma$ have effectively semilinear Parikh images if and only if:*

1. *Γ^- is a transitive forest and*

2. *the neighborhood of every unlooped vertex in Γ is a looped clique.*

Moreover, we identify another type of stacked counters as expressively complete among those mechanisms with semilinearity. They are similar to the mechanisms in the results on the emptiness problem. Namely, they are obtained by alternatingly *building stacks* and *adding blind counters*. Furthermore, stacked counters exhibit a range of properties desirable for analysis and they offer a way to model recursive programs with numeric data types (Hague and Lin [6] have applied a model that is subsumed by stacked counter automata).

More precisely, SC^- is the smallest class of monoids that contains the trivial monoid $\mathbf{1}$ and has the property that if M belongs to SC^-, then both $\mathbb{B} * M$ and $\mathbb{Z} \times M$ belong to SC^- as well. To summarize, we have three types of stacked counters: (i) SC^-, where we only have blind counters, (ii) $\mathsf{SC}^\pm$, where we start with partially blind counters, but after building stacks, we can only add blind counters, and (iii) SC^+, where we start with partially blind counters and can add them even after building stacks.

These results have been published in [1].

Silent transitions For each storage mechanism, an important question is whether silent transitions (i.e. those which read no input but can manipulate the storage) are necessary to accept all languages. Indeed, if silent transitions can be eliminated, we can decide the membership status of a given input word by examining a finite number of paths through the automaton. Therefore, we ask for which monoids we can avoid silent transitions. We show that among a class of storage mechanisms, stacked counters of the type SC^- are those where this is possible. Again, this generalizes the corresponding fact for (i) pushdown automata, (ii) blind multicounter automata, and (iii) automata with access to a pushdown storage and blind counters. Results (i) and (ii) had been established by Greibach [4, 5] and (iii) is due to Hoogeboom [9].

These results have been published in [21].

Computing downward closures We also consider the computation of downward closures. It is well-known that the downward closure, i.e. the set of (not necessarily contiguous) subwords, of every language is regular [7, 8]. Moreover, computing a finite automaton for the downward closure of a given language would make a range of analysis techniques applicable. However, this cannot be done in general. In fact, there are only few known methods for computing downward closures for languages. It is shown here that for all those graph monoids that guarantee semilinearity (equivalently, for stacked counters), downward closures can be computed. This generalizes the computability of downward closures for context-free languages, as obtained by van Leeuwen [18] and Courcelle [2].

This result has been published in [19].

Bibliography

[1] P. Buckheister and G. Zetzsche. "Semilinearity and Context-Freeness of Languages Accepted by Valence Automata". In: *Proc. of the 38th International Symposium on Mathematical Foundations of Computer Science (MFCS 2013)*. Vol. 8087. LNCS. Springer-Verlag, 2013, pp. 231–242.

[2] B. Courcelle. "On constructing obstruction sets of words". In: *Bulletin of the EATCS* 44 (1991), pp. 178–186.

[3] H. Fernau and R. Stiebe. "Sequential grammars and automata with valences". In: *Theoretical Computer Science* 276 (2002), pp. 377–405.

[4] S. A. Greibach. "Remarks on blind and partially blind one-way multicounter machines". In: *Theoretical Computer Science* 7.3 (1978), pp. 311 –324.

[5] S. A. Greibach. "A New Normal-Form Theorem for Context-Free Phrase Structure Grammars". In: *Journal of the ACM* 12.1 (1965), pp. 42–52.

[6] M. Hague and A. W. Lin. "Model Checking Recursive Programs with Numeric Data Types". In: *Proc. of the 23rd International Conference on Computer Aided Verification (CAV 2000)*. Ed. by G. Gopalakrishnan and S. Qadeer. Vol. 6806. LNCS. Berlin/Heidelberg: Springer-Verlag, 2011, pp. 743–759.

[7] L. H. Haines. "On free monoids partially ordered by embedding". In: *Journal of Combinatorial Theory* 6.1 (1969), pp. 94–98.

[8] G. Higman. "Ordering by divisibility in abstract algebras." In: *Proceedings of the London Mathematical Society. Third Series* 2 (1952), pp. 326–336.

[9] H. J. Hoogeboom. "Context-Free Valence Grammars - Revisited". In: *Proc. of the 5th International Conference on Developments in Language Theory (DLT 2001)*. Ed. by W. Kuich, G. Rozenberg, and A. Salomaa. Vol. 2295. LNCS. Berlin/Heidelberg: Springer-Verlag, 2002, pp. 293–303.

[10] B. Khoussainov and A. Nerode. "Automatic Presentations of Structures". In: *LCC: International Workshop on Logic and Computational Complexity*. Vol. 960. LNCS. Berlin/Heidelberg: Springer-Verlag, 1995, pp. 367–392.

[11] J. Leroux, G. Sutre, and P. Totzke. "On the Coverability Problem for Pushdown Vector Addition Systems in One Dimension". In: *Proc. of the 42nd International Colloquium on Automata, Languages and Programming (ICALP 2015)*. Berlin/Heidelberg: Springer-Verlag, 2015, pp. 324–336.

[12] M. Lohrey and G. Sénizergues. "When Is a Graph Product of Groups Virtually-Free?" In: *Communications in Algebra* 35.2 (2007), pp. 617–621.

[13] M. Lohrey and B. Steinberg. "The submonoid and rational subset membership problems for graph groups". In: *Journal of Algebra* 320.2 (2008), pp. 728–755.

[14] M. Lohrey and G. Zetzsche. "On Boolean closed full trios and rational Kripke frames". In: *Proc. of the 31st International Symposium on Theoretical Aspects of Computer Science (STACS 2014)*. Vol. 25. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2014, pp. 530–541.

[15] R. J. Parikh. "On Context-Free Languages". In: *Journal of the ACM* 13.4 (1966), pp. 570–581.

[16] K. Reinhardt. "Reachability in Petri Nets with Inhibitor Arcs". In: *Electronic Notes in Theoretical Computer Science* 223 (2008). Proceedings of the Second Workshop on Reachability Problems in Computational Models (RP 2008), pp. 239–264.

[17] E. Render. "Rational Monoid and Semigroup Automata". PhD thesis. University of Manchester, 2010.

[18] J. van Leeuwen. "Effective constructions in well-partially-ordered free monoids". In: *Discrete Mathematics* 21.3 (1978), pp. 237–252.

[19] G. Zetzsche. "Computing Downward Closures for Stacked Counter Automata". In: *Proc. of the 32nd International Symposium on Theoretical Aspects of Computer Science (STACS 2015)*. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015, pp. 743–756.

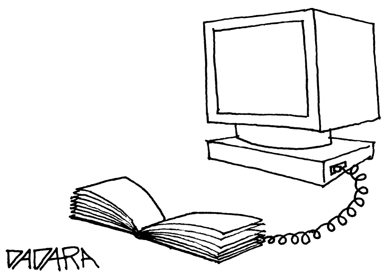
[20] G. Zetzsche. "On the Capabilities of Grammars, Automata, and Transducers Controlled by Monoids". In: *Proc. of the 38th International Colloquium on Automata, Languages and Programming (ICALP 2011)*. Vol. 6756. LNCS. Springer-Verlag, 2011, pp. 222–233.

[21] G. Zetzsche. "Silent Transitions in Automata with Storage". In: *Proc. of the 40th International Colloquium on Automata, Languages and Programming (ICALP 2013)*. Vol. 7966. LNCS. Springer-Verlag, 2013, pp. 434–445.

[22] G. Zetzsche. "The Emptiness Problem for Valence Automata or: Another Decidable Extension of Petri Nets". In: *Proc. of the 9th International Workshop on Reachability Problems (RP 2015)*. Vol. 9328. LNCS. Springer-Verlag, 2015, pp. 166–178.

[23] G. Zetzsche, M. Lohrey, and D. Kuske. *On Boolean closed full trios and rational Kripke frames*. Extended version of [14]. To appear in Theory of Computing Systems. 2016.

Book Introduction by the Authors



Book Introduction by the Authors

Invited by

Kazuo Iwama

iwama@kuis.kyoto-u.ac.jp
Bulletin Editor
Kyoto University, Japan

The Golden Ticket
P, NP, and the Search for the Impossible*

Lance Fortnow

Georgia Institute of Technology

fortnow@cc.gatech.edu

As a reader of this bulletin you already understand the importance of theoretical computer science and why we care so much about the P versus NP problem. While the P v NP problem gets mentioned in the United States occasionally in television shows, video games, even the Unofficial Guide to Disney World [6] (for the traveling salesman problem of getting around the park), the general public knows little about this important problem. It's a rare popular science book that discusses P v NP, theoretical computer science or computer science at all.

In 2008, Moshe Vardi asked me to write a survey on the P versus NP problem for the Communications of the ACM. Instead of a typically technical survey, I focused on the ideas of P v NP and the challenges and opportunities that it affords us. That survey [2] would engender over a quarter million downloads. The broader computer science community wanted to understand the P versus NP problem, not as a technical relationship between deterministic and nondeterministic Turing machines, but as a concept that guides how we solve challenging algorithmic problems.

Given the popularity of the survey, I decided I could convert that survey into a book with each section of the survey expanded into a chapter for the book. Instead of aiming for a computer science savvy audience, like I did for the survey, I decided to write for the scientifically-interested general public. More specifically I aimed the book at a high school student who wanted to know more about computer science than just programming. When I went to high school I got inspired by books like Carl Sagan's *Cosmos* [5] or Douglas Hofstadter's *Gödel, Escher, Bach* [3]. I wanted to do the same, to inspire young adults to think about computer science through the lens of the P v NP problem.

Four years later I completed the book, *The Golden Ticket: P, NP and the Search for the Impossible*. The name refers to Roald Dahl's *Charlie and the Chocolate Factory* [1] where golden tickets hidden in candy bars gave admission

*Princeton University Press, 2013, 192pp, ISBN 978-0691156491, goldenticket.fortnow.com.

to a rare tour of the fabled Willy Wonka Chocolate Factory. The search for that golden ticket makes a great analogy to the P versus NP problem in many ways.

The Golden Ticket has done reasonably well for a popular science book with good reviews and some awards including being named one of Amazon's top 20 science books for 2013. The book has been translated into Chinese, Japanese, Finnish and Russian and a paperback edition will be released in spring 2017.

This book is not meant for the readers of the bulletin but perhaps your friends or family members that you would like to share the importance of what theoretical computer science is or better yet, a teenager who might be inspired by the excitement of our field.

To give a flavor of this book what follows is a summary of each chapter.

Chapter 1: The Golden Ticket

This introductory chapter sketches out the P versus NP problem through a series of examples, from the traditional traveling salesman to controlling a robotic hand.

Chapter 2: The Beautiful World

My science fiction chapter that looks at a world where P = NP, not just in the formal sense but supposing we had a truly efficient and practical algorithm for NP-complete problems. Instead of focusing on the negative (crypto is dead), I look instead to the positive (we cure cancer!).

Chapter 3: P and NP

We give examples of P and NP problems through several examples. Never in the book do I define P and NP in the formal sense–rather I go for intuitive notions of these concepts. I develop a world called Frenemy, where every pair of people are either friends and enemies and use that world to describe a series of graph problems, without actually talking about graphs.

Chapter 4: The Hardest Problems in NP

Focusing on NP-complete problems again in an intuitive way giving many examples (Rubik's Cube is easy, Sudoku is hard), as well as a short history of P, NP and NP-completeness and the naming of those classes based on Donald Knuth's wonderful and hilarious SIGACT News article [4].

Chapter 5: The Prehistory of P versus NP

What led up to the P v NP problem? I describe the history in the western world from Turing to Karp. I also talk about the development of the study of Perebor in Russia and how the cold war and mathematical politics made complexity research in the Soviet Union trickier and more dangerous.

Chapter 6: Dealing with Hardness

NP-completeness is not the death knoll for a computational problem, just an indication that we won't find an efficient algorithm that gives an exact solution all of the time. I discuss many approaches in dealing with hard problems including brute force, heuristics, small parameters and approximation. Sometimes you need to find a different problem or just accept that you cannot solve what you want to solve.

Chapter 7: Proving P ≠ NP

How have people tried to settle the P v NP problem and why haven't those techniques worked? This chapter discusses diagonalization and circuits, as well as reviewing the Deolalikar story and common mistakes made in the all too many P v NP "proofs" that get sent my way.

Chapter 8: Secrets

Cryptography from Ceasar to RSA, zero-knowledge via Sudoku, secure computation and pseudorandomness. Does P = NP kill cryptography? Not, but it will make it more challenging.

Chapter 9: Quantum

A very high level view of quantum physics, quantum computing, quantum cryptography and teleportation.

Chapter 10: The Future

What are the future challenges of computing (already dated), from parallelism, big data and the Internet of things. Final line of the book: As long as P versus NP remains a mystery we do not know what we cannot do, and that's liberating.

Conclusion

I truly aimed to tell the story of P and NP in an understandable and intuitive manner. How well I succeeded is not my call to make. I aimed for the book to help people understand this great question and get them interested in P v NP and theoretical ideas in computer science.

References

[1] R. Dahl. *Charlie and the Chocolate Factory*. Alfred A. Knopf, New York, 1964.

[2] L. Fortnow. The status of the P versus NP problem. *Communications of the ACM*, 52(9):78–86, September 2009. Review Article.

[3] D. Hofstadter. *Gödel, Esher, Bach: An Eternal Golden Braid*. Basic Books, New York, 1979.

[4] D. Knuth. A terminological proposal. *SIGACT News*, 6(1):12–18, 1974.

[5] C. Sagan. *Cosmos*. Random House, New York, 1980.

[6] B. Sehlinger and L. Testa. *The Unofficial Guide to Disney World 2010*. Wiley, New York, 2010.

Announcement

ICALP 2017

44th International Colloquium on Automata, Languages and Programming
10-14 July 2017, Warsaw, Poland
http://icalp17.mimuw.edu.pl/

Call for Papers

The 44th International Colloquium on Automata, Languages, and Programming (ICALP) will take place in Warsaw, Poland, on 10-14 July 2017. ICALP is the main conference and annual meeting of the European Association for Theoretical Computer Science (EATCS). As usual, ICALP will be preceded by a series of workshops, which will take place on July 10.

Important dates

Submission deadline: Friday, February 17, 2017, 23:59 PST (Pacific Standard Time, UTC-8)
Author notification: April 14, 2017
Final manuscript due: April 30, 2017

Deadlines are firm; late submissions will not be considered.

Proceedings

ICALP proceedings are published in the Leibniz International Proceedings in Informatics (LIPIcs) series. This is a series of high-quality conference proceedings across all fields in informatics established in cooperation with Schloss Dagstuhl - Leibniz Center for Informatics. LIPIcs volumes are published according to the principle of Open Access, i.e., they are available online and free of charge.

Invited Speakers

Mikołaj Bojanczyk (University of Warsaw, Poland)
Monika Henzinger (University of Vienna, Austria)
Mikkel Thorup (University of Copenhagen, Denmark)

Topics

Papers presenting original research on all aspects of theoretical computer science are sought. Typical but not exclusive topics of interest are:

Track A: Algorithms, Complexity and Games

- Algorithmic Game Theory
- Approximation Algorithms
- Combinatorial Optimization
- Combinatorics in Computer Science
- Computational Biology
- Computational Complexity
- Computational Geometry
- Cryptography
- Data Structures
- Design and Analysis of Algorithms
- Machine Learning
- Parallel, Distributed and External Memory Computing
- Randomness in Computation
- Quantum Computing

Track B: Logic, Semantics, Automata and Theory of Programming

- Algebraic and Categorical Models
- Automata, Games, and Formal Languages
- Emerging and Non-standard Models of Computation
- Databases, Semi-Structured Data and Finite Model Theory
- Principles and Semantics of Programming Languages
- Logic in Computer Science, Theorem Proving and Model Checking

- Models of Concurrent, Distributed, and Mobile Systems
- Models of Reactive, Hybrid and Stochastic Systems
- Program Analysis and Transformation
- Specification, Refinement, Verification and Synthesis
- Type Systems and Theory, Typed Calculi

Track C: Foundations of Networked Computation: Models, Algorithms and Information Management

- Algorithmic Aspects of Networks and Networking
- Formal Methods for Network Information Management
- Foundations of Privacy, Trust and Reputation in Networks
- Mobile and Wireless Networks and Communication
- Network Economics and Incentive-Based Computing Related to Networks
- Networks of Low Capability Devices
- Network Mining and Analysis
- Overlay Networks and P2P Systems
- Specification, Semantics, Synchronization of Networked Systems
- Theory of Security in Networks

Submission Guidelines

Authors are invited to submit an extended abstract of no more than 12 pages, excluding references, in the LIPIcs style (http://www.dagstuhl.de/en/publications/lipics/instructions-for-authors/) presenting original research on the theory of Computer Science. The usage of pdflatex and the LIPIcs style file (see http://drops.dagstuhl.de/styles/lipics-v2016/lipics-v2016-authors/lipics-v2016-sample-article.tex and http://drops.dagstuhl.de/styles/lipics-v2016/lipics-v2016-authors.tgz) are mandatory: papers that deviate significantly from the required format may be rejected without consideration of merit. All submissions are electronic via EasyChair: https://easychair.org/conferences/?conf=icalp2017

All technical details necessary for a proper evaluation of a submission must be included in the 12-page submission or in a clearly-labelled appendix, to be consulted at the discretion of program committee members. Authors are encouraged to also make full versions of their submissions freely accessible in an on-line repository such as ArXiv, HAL, ECCC.

Submissions should be made to the appropriate track of the conference. No prior publication or simultaneous submission to other publication outlets (either a conference or a journal) is allowed.

Should I submit my paper to Track A or Track C?

While the scope of Tracks A and B are generally well understood given their long history, the situation for Track C may be less obvious. In particular, some clarifications may be helpful regarding areas of potential overlap, especially between Tracks A and C.

The aim for Track C is to be the leading venue for theory papers truly motivated by networking applications, and/or proposing theoretical results relevant to real networking, certified analytically, but not necessarily tested practically. The motivation for the track was the lack of good venues for theory papers motivated by applications in networking. On the one hand, the good networking conferences typically ask for extended experiments and/or simulations, while the TCS community is hardly able to do such experiments or simulations. On the other hand, the good conferences on algorithms tend to judge a paper based only on its technical difficulty and on its significance from an algorithmic perspective, which may not be the same as when judging the paper from the perspective of impact on networks.

Several areas of algorithmic study of interest to track C have a broad overlap with track A. Graph algorithmics can belong in either, though if the work is not linked to networking, it is more appropriate in track A. Algorithmic game theory is another area of major overlap. Aspects involving complexity, the computation of equilibria and approximations, belong more in Track A, while results with applications in auctions, networks and some aspects of mechanism design belong in Track C.

Finally, it should be noted that algorithms and complexity of message-passing based distributed computing belong squarely in track C, while certain other aspects of distributed computing do not fall under its scope.

Best Paper Awards

As in previous editions of ICALP, there will be best paper and best student paper awards for each track of the conference. In order to be eligible for a best student paper award, a paper should be authored only by students and should be marked as such upon submission.

Committees

Track A: Algorithms, complexity, and games
Piotr Indyk (MIT, USA, Chair)
Peyman Afshani (Aarhus University)
Pankaj Agarwal (Duke)
Karl Bringmann (Max-Planck-Institute)
Arkadev Chattopadhyay (Tata Institute of Fundamental Research)
Shiri Chechik (Tel-Aviv University)
Alina Ene (University of Warwick)
Yuval Filmus (Technion)
Parikshit Gopalan (Microsoft)
Roberto Grossi (Universitá di Pisa)
Anupam Gupta (CMU)
Yuval Ishai (Technion)
Michael Kapralov (EPFL)
Robert Kleinberg (Cornell)
Pinyan Lu (Shanghai University)
Frederic Magniez (Université Paris Diderot)
Mohammad Mahdian (Google)
Daniel Marx (Hungarian Academy of Sciences)
Danupon Nanongkai (KTH Royal Institute of Technology)
Jelani Nelson (Harvard)
Marcin Pilipczuk (University of Warsaw)
Piotr Sankowski (University of Warsaw)
Thomas Sauerwald (University of Cambridge)
Christian Scheidler (University of Paderborn)
Christian Sohler (TU Dortmund)
Kavitha Telikepalli (Indian Institute of Science)
Vinod Vaikuntanathan (MIT)
László A. Végh (London School of Economics)
Suresh Venkatasubramanian(Utah)
Thomas Vidick (Caltech)
Hoeteck Wee (ENS)
Seth Weinberg (Princeton)
Oren Weinmann (University of Haifa)

Track B: Logic, semantics, automata and theory of Programming
Anca Muscholl (Univ. of Bordeaux, chair)
Pablo Barcelo (Universidad de Chile)
Achim Blumensath (Masaryk Univ., Brno)
Thomas Brihaye (Univ. de Mons)
Krishnendu Chatterjee (IST Austria)

Thierry Coquand (Univ. of Gothenburg)
Anuj Dawar (Univ. of Cambridge)
Jörg Endrullis (Vrije Universiteit Amsterdam)
Dana Fisman (Univ. of Pennsylvania)
Martin Hoffmann (Ludwig-Maximilian Univ.)
Radha Jagadeesan (DePaul Univ.)
Stefan Kiefer (Univ. of Oxford)
Emanuel Kieronski (Univ. of Wrocław)
Stefan Kreutzer (TU Berlin)
Salvatore La Torre (Univ. degli Studi di Salerno)
Antony Widjaja Lin (Yale-NUS College, Singapore)
Wim Martens (Univ. of Bayreuth)
Paul-André Mellies (IRIF, Paris)
Luca Padovani (Univ. di Torino)
Catiuscia Palamidessi (INRIA Saclay, LIX)
Giovanni Pighizzini (Univ. of Milano)
Jean-Éric Pin (IRIF, Paris)
Alexandra Silva (University College London)
Jean-Marc Talbot (Aix-Marseille Univ.)
Thomas Wilke (Univ. of Kiel)
Mahesh Viswanathan (Univ. of Illinois)
James Worell (Univ. of Oxford)

Track C: Foundations of networked computation: Models, algorithms and information management

Fabian Kuhn (University of Freiburg, Germany, Chair)
Ittai Abraham (VMware Research, USA)
Antonio Fernandez Anta (IMDEA Research, Spain)
James Aspnes (Yale U., USA)
Keren Censor-Hillel (Technion, Israel)
Yuval Emek (Technion, Israel)
George Giakkoupis (INRIA Rennes, France)
Seth Gilbert (National U. of Singapore)
Mohsen Ghaffari (currently at MIT, will move to ETH Zurich)
Bernhard Haeupler (CMU, USA)
Amos Korman (CNRS & U. Paris Diderot, France)
Adrian Kosowski (INRIA, Paris)
Christoph Lenzen (MPI Saarbrücken, Germany)
Toshimitsu Masuzawa (Osaka U. Japan)

Konstantinos Panagiotou (LMU Munich, Germany)
Merav Parter (MIT)
Boaz Patt-Shamir (Tel Aviv U., Israel)
Yvonne-Anne Pigonlet (ABB Research, Switzerland)
Alessandro Panconesi (Sapienza U., Rome, Italy)
Sergio Rajsbaum (UNAM, Mexico)
Andrea Richa (Arizona State U., USA)
Hsin-Hao Su (MIT, USA)
Jukka Suomela (Aalto U., Finland)
Philipp Woelfel (U. Calgary, Canada)

CRS 2017

The 12th International Computer Science Symposium in Russia,
8-12 June 2017, Kazan, Russia
http://logic.pdmi.ras.ru/csr2017

Call for Papers

CSR is an annual conference that intends to reflect the broad scope of international cooperation in computer science. The topics covered vary from year to year, but in general try to cover as much of the contemporary computer science as possible. They include, but are not limited to:

algorithms and data structures
combinatorial optimization
constraint solving
computational complexity
cryptography
combinatorics in computer science
formal languages and automata
algorithms for concurrent and distributed systems, networks
applications of logic to computer science, e.g. proof theory, model checking and verification
formal and algorithmic aspects of bio-informatics
current challenges such as quantum computing

Important dates Please note the following dates!

Deadline for submissions: December 15, 2016 (23:59 anywhere on Earth).
Notification of acceptance: February 15, 2017.
Conference dates: June 8-12, 2017.

Invited lectures There will be plenary lectures by:

Thierry Coquand (Chalmers), distinguished opening lecture
Javier Esparza (Munich)

Elham Kashefi (Paris and Edinburgh)
Andrew McGregor (Amherst)
Ronitt Rubinfeld (MIT)
Marc Zeitoun (Bordeaux)

Submission and publication Authors are invited to submit papers presenting original research in the conference topics, in electronic form (pdf format) via EasyChair.

Submissions must be unpublished, not under review for publication elsewhere, and provide sufficient information to judge their merits.

Submissions must be in English, and not exceed 12 pages, including the title page, in Springer's LNCS LaTeX style (instructions available here). Additional material, to be read at the discretion of reviewers and PC members, may be provided in a clearly marked appendix or by reference to a manuscript on a web site.

Accepted papers will be published in an LNCS volume by Springer. Instructions for formatting camera-ready versions will be communicated to the authors of accepted papers.

For an accepted paper to be included in the proceedings, one of the authors must commit to presenting the paper at the conference.

Additionally, selected papers will be invited to a special issue of Theory of Computing Systems and will be refereed according to the journal's procedure.

Yandex Awards for the best paper and for the best student paper will be awarded by the PC.

Organizers and sponsors CSR is organized by Kazan Federal university (Volga region), with the support of (tentative list):

The Russia Foundation for Basic Research, `www.rfbr.ru/rffi/eng/info`
Kazan Federal University, `www.kpfu.ru/eng`
Yandex, `www.yandex.com`

Organizing committee (tentative) The organizing committee can be reached at the following address: `CSR2017.Kazan@gmail.com`.

Farid Ablayev (KFU), **chair**	Aida Gainutdinova (KFU)
Anton Marchenko (KFU)	Daniil Musatov (MIPT)
Alexander Vasiliev (KFU)	Valeria Volkova (KFU)
Mansur Ziatdinov (KFU)	Marsel Sitdikov (KFU)

Program Committee List of members

Farid Ablayev (Kazan, RU)
Isolde Adler (Leeds, UK)
Véronique Bruyère (Mons, BE)
Hubie Chen (San Sebastian, ES)
Stéphanie Delaune (Cachan, FR)
Elena Grigorescu (Purdue, US)
K.N. Kumar (Chennai, IN)
Meena Mahajan (Chennai, IN)
Catuscia Palamidessi (Saclay, FR)
Kavitha Telikepalli (Mumbai, IN)
Szymon Torunczyk (Warsaw, PL)
Mikhail Volkov (Ekaterinburg, RU)
Pascal Weil (Bordeaux, FR), **chair**
Ittai Abraham (Jerusalem, IL)
Frédérique Bassino (Paris Nord, FR)
Maike Buchin (Bochum, DE)
Anuj Dawar (Cambridge, UK)
Anna Frid (Marseille, FR)
Krishna S (Mumbai, IN)
Frédéric Magniez (Paris, FR)
Grigory Marshalko (Moscow, RU)
Victor Selivanov (Novosibirsk, RU)
Thomas Thierauf (Aalen, DE)
Hélène Touzet (Lille, FR)
Dorothea Wagner (Karlsruhe, DE)

Steering Committee for the CSR series List of members

Edward A. Hirsch (St.Petersburg, RU)
Juhani Karhumäki (Turku, FI)
Ernst W. Mayr (Munich, DE)
Alexander Razborov (Chicago, US and Moscow, RU)
Mikhail Volkov (Ekaterinburg, RU)

Previous CSR conferences CSR 2016 took place in St.Petersburg, Russia. The list of previous CSR webpages can be found at `http://logic.pdmi.ras.ru/~csr/`

European

Association for

Theoretical

Computer

Science

E A T C S

EATCS

HISTORY AND ORGANIZATION

EATCS is an international organization founded in 1972. Its aim is to facilitate the exchange of ideas and results among theoretical computer scientists as well as to stimulate cooperation between the theoretical and the practical community in computer science.

Its activities are coordinated by the Council of EATCS, which elects a President, Vice Presidents, and a Treasurer. Policy guidelines are determined by the Council and the General Assembly of EATCS. This assembly is scheduled to take place during the annual **I**nternational **C**olloquium on **A**utomata, **L**anguages and **P**rogramming (ICALP), the conference of EATCS.

MAJOR ACTIVITIES OF EATCS

- Organization of ICALP;
- Publication of the "Bulletin of the EATCS;"
- Award of research and academic career prizes, including the EATCS Award, the Gödel Prize (with SIGACT), the Presburger Award, the EATCS Distinguished Dissertation Award, the Nerode Prize (joint with IPEC) and best papers awards at several top conferences;
- Active involvement in publications generally within theoretical computer science.

Other activities of EATCS include the sponsorship or the cooperation in the organization of various more specialized meetings in theoretical computer science. Among such meetings are: CIAC (Conference of Algorithms and Complexity), CiE (Conference of Computer Science Models of Computation in Context), DISC (International Symposium on Distributed Computing), DLT (International Conference on Developments in Language Theory), ESA (European Symposium on Algorithms), ETAPS (The European Joint Conferences on Theory and Practice of Software), LICS (Logic in Computer Science), MFCS (Mathematical Foundations of Computer Science), WADS (Algorithms and Data Structures Symposium), WoLLIC (Workshop on Logic, Language, Information and Computation), WORDS (International Conference on Words).

Benefits offered by EATCS include:
- Subscription to the "Bulletin of the EATCS;"
- Access to the Springer Reading Room;
- Reduced registration fees at various conferences;
- Reciprocity agreements with other organizations;
- 25% discount when purchasing ICALP proceedings;
- 25% discount in purchasing books from "EATCS Monographs" and "EATCS Texts;"
- Discount (about 70%) per individual annual subscription to "Theoretical Computer Science;"
- Discount (about 70%) per individual annual subscription to "Fundamenta Informaticae."

Benefits offered by EATCS to Young Researchers also include:
- Database for Phd/MSc thesis
- Job search/announcements at Young Researchers area

(1) THE ICALP CONFERENCE

ICALP is an international conference covering all aspects of theoretical computer science and now customarily taking place during the second or third week of July. Typical topics discussed during recent ICALP conferences are: computability, automata theory, formal language theory, analysis of algorithms, computational complexity, mathematical aspects of programming language definition, logic and semantics of programming languages, foundations of logic programming, theorem proving, software specification, computational geometry, data types and data structures, theory of data bases and knowledge based systems, data security, cryptography, VLSI structures, parallel and distributed computing, models of concurrency and robotics.

Sites of ICALP meetings:

- Paris, France 1972
- Saarbrücken, Germany 1974
- Edinburgh, UK 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, UK 1990
- Madrid, Spain 1991
- Wien, Austria 1992
- Lund, Sweden 1993
- Jerusalem, Israel 1994
- Szeged, Hungary 1995
- Paderborn, Germany 1996
- Bologne, Italy 1997
- Aalborg, Denmark 1998
- Prague, Czech Republic 1999
- Genève, Switzerland 2000
- Heraklion, Greece 2001
- Malaga, Spain 2002
- Eindhoven, The Netherlands 2003
- Turku, Finland 2004
- Lisabon, Portugal 2005
- Venezia, Italy 2006
- Wrocław, Poland 2007
- Reykjavik, Iceland 2008
- Rhodes, Greece 2009
- Bordeaux, France 2010
- Zürich, Switzerland 2011
- Warwick, UK 2012
- Riga, Latvia 2013
- Copenhagen, Denmark 2014
- Kyoto, Japan 2015
- Rome, Italy 2016

(2) THE BULLETIN OF THE EATCS

Three issues of the Bulletin are published annually, in February, June and October respectively. The Bulletin is a medium for *rapid* publication and wide distribution of material such as:

- EATCS matters;
- Technical contributions;
- Columns;
- Surveys and tutorials;
- Reports on conferences;
- Information about the current ICALP;
- Reports on computer science departments and institutes;
- Open problems and solutions;
- Abstracts of Ph.D. theses;
- Entertainments and pictures related to computer science.

Contributions to any of the above areas are solicited, in electronic form only according to formats, deadlines and submissions procedures illustrated at `http://www.eatcs.org/bulletin`. Questions and proposals can be addressed to the Editor by email at `bulletin@eatcs.org`.

(3) OTHER PUBLICATIONS

EATCS has played a major role in establishing what today are some of the most prestigious publication within theoretical computer science.

These include the *EATCS Texts* and the *EATCS Monographs* published by Springer-Verlag and launched during ICALP in 1984. The Springer series include *monographs* covering all areas of theoretical computer science, and aimed at the research community and graduate students, as well as *texts* intended mostly for the graduate level, where an undergraduate background in computer science is typically assumed.

Updated information about the series can be obtained from the publisher.

The editors of the EATCS Monographs and Texts are now M. Henzinger (Vienna), J. Hromkovič (Zürich), M. Nielsen (Aarhus), G. Rozenberg (Leiden), A. Salomaa (Turku). Potential authors should contact one of the editors.

EATCS members can purchase books from the series with 25% discount. Order should be sent to:

Prof.Dr. G. Rozenberg, LIACS, University of Leiden,
P.O. Box 9512, 2300 RA Leiden, The Netherlands

who acknowledges EATCS membership and forwards the order to Springer-Verlag.

The journal *Theoretical Computer Science*, founded in 1975 on the initiative of EATCS, is published by Elsevier Science Publishers. Its contents are mathematical and abstract in spirit, but it derives its motivation from practical and everyday computation. Its aim is to understand the nature of computation and, as a consequence of this understanding, provide more efficient methodologies.

The Editor-in-Chief of the journal currently are D. Sannella (Edinburgh), L. Kari and P.G. Spirakis (Patras).

ADDITIONAL EATCS INFORMATION

For further information please visit `http://www.eatcs.org`, or contact the President of EATCS:

Prof. Paul Spirakis,
Email: `president@eatcs.org`

EATCS MEMBERSHIP

DUES

The dues are € 30 for a period of one year (two years for students / Young Researchers). Young Researchers, after paying, have to contact `secretary@eatcs.org`, in order to get additional years. A new membership starts upon registration of the payment. Memberships can always be prolonged for one or more years.

In order to encourage double registration, we are offering a discount for SIGACT members, who can join EATCS for € 25 per year. We also offer a five-euro discount on the EATCS membership fee to those who register both to the EATCS and to one of its chapters. Additional € 25 fee is required for ensuring the *air mail* delivery of the EATCS Bulletin outside Europe.

HOW TO JOIN EATCS

You are strongly encouraged to join (or prolong your membership) directly from the EATCS website `www.eatcs.org`, where you will find an online registration form and the possibility of secure online payment. Alternatively, contact the Secretary Office of EATCS:

Mrs. Efi Chita,
Computer Technology Institute & Press (CTI)
1 N. Kazantzaki Str., University of Patras campus,
26504, Rio, Greece
Email: `secretary@eatcs.org`,
Tel: +30 2610 960333, Fax: +30 2610 960490

If you are an EATCS member and you wish to prolong your membership or renew the subscription you have to use the Renew Subscription form. The dues can be paid via paypal and all major credit cards are accepted.

For adittional information please contact the Secretary of EATCS:

Prof. Ioannis Chatzigiannakis,
via Ariosto 25, floor II, room B214,
Sapienza University of Rome,
Rome 00185, Italy
Email: `secretary@eatcs.org`,
Tel: +39 677274073, Fax: +39 0677274002