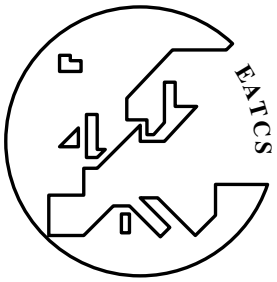
ISSN 0252–9742

Bulletin

of the

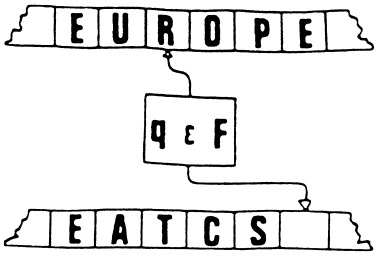
European Association for Theoretical Computer Science

EATCS



Number 113 June 2014

Council of the European Association for Theoretical Computer Science



President:	Luca Aceto	Iceland
Vice Presidents:	Paul Spirakis	Greece
	Burkhard Monien	Germany
	Antonin Kucera	Czech Republic
Treasurer:	Dirk Janssens	Belgium
Bulletin Editor:	Kazuo Iwama	Kyoto, Japan

Lars Arge	Denmark	Elvira Mayordomo	Spain
Jos Baeten	The Netherlands	Luke Ong	UK
Paul Beame	USA	Catuscia Palamidessi	France
Mikolaj Bojanczyk	Poland	David Peleg	Israel
Josep Díaz	Spain	Giuseppe Persiano	Italy
Zoltán Ésik	Hungary	Alberto Policriti	Italy
Fedor Fomin	Norway	Alberto Marchetti Spaccamela	Italy
Pierre Fraigniaud	France	Vladimiro Sassone	UK
Leslie Ann Goldberg	UK	Roger Wattenhofer	Switzerland
Monika Henzinger	Austria	Thomas Wilke	Germany
Christos Kaklamanis	Greece	Peter Widmayer	Switzerland
Juhani Karhumaki	Finland	Gerhard Wöeginger	The Netherlands

Past Presidents:

Maurice Nivat	(1972–1977)	Mike Paterson	(1977–1979)
Arto Salomaa	(1979–1985)	Grzegorz Rozenberg	(1985–1994)
Wilfred Brauer	(1994–1997)	Josep Díaz	(1997–2002)
Mogens Nielsen	(2002–2006)	Giorgio Ausiello	(2006–2009)
Burkhard Monien	(2009–2012)		

Secretary Office:	Ioannis Chatzigiannakis	Greece
	Efi Chita	Greece

EATCS Council Members

email addresses

Luca Aceto . . . . . luca@ru.is
Lars Arge . . . . . large@madalgo.au.dk
Jos Baeten . . . . . jos.baeten@cwi.nl
Paul Beame . . . . . beame@cs.washington.edu
Mikolaj Bojanczyk . . . . . bojan@mimuw.edu.pl
Josep Díaz . . . . . diaz@lsi.upc.es
Zoltán Ésik . . . . . ze@inf.u-szeged.hu
Fedor Fomin . . . . . fomin@ii.uib.no
Pierre Fraigniaud . . pierre.fraigniaud@liafa.univ-paris-diderot.fr
Leslie Ann Goldberg . . . . . leslie.goldberg@cs.ox.ac.uk
Monika Henzinger . . . . . monika.henzinger@univie.ac.at
Dirk Janssens . . . . . Dirk.Janssens@ua.ac.be
Christos Kaklamanis . . . . . kakl@ceid.upatras.gr
Juhani Karhumäki . . . . . karhumak@cs.utu.fi
Antonin Kucera . . . . . tony@fi.muni.cz
Elvira Mayordomo . . . . . elvira@unizar.es
Burkhard Monien . . . . . bm@uni-paderborn.de
Luke Ong . . . . . luke.Ong@cs.ox.a.uk
Catuscia Palamidessi . . . . . catuscia@lix.polytechnique.fr
David Peleg . . . . . peleg@wisdom.weizmann.ac.il
Giuseppe Persiano . . . . . giuper@dia.unisa.it
Alberto Policriti . . . . . alberto.policriti@uniud.it
Alberto Marchetti Spaccamela . . . . . alberto@dis.uniroma1.it
Vladimiro Sassone . . . . . vs@ecs.soton.ac.uk
Roger Wattenhofer . . . . . wattenhofer@tik.ee.ethz.ch
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 2ε using the class `beatcs.cls` (a version of the standard LaTeX 2ε `article` class). All sources, including figures, and a reference PDF version must be bundled in a ZIP file.

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

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

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

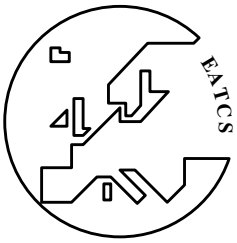
Deadlines for submissions of reports are January, May and September 15th, respectively for the February, June and October issues. Editorial decisions about submitted technical contributions will normally be made in 6/8 weeks. Accepted papers will appear in print as soon as possible thereafter.

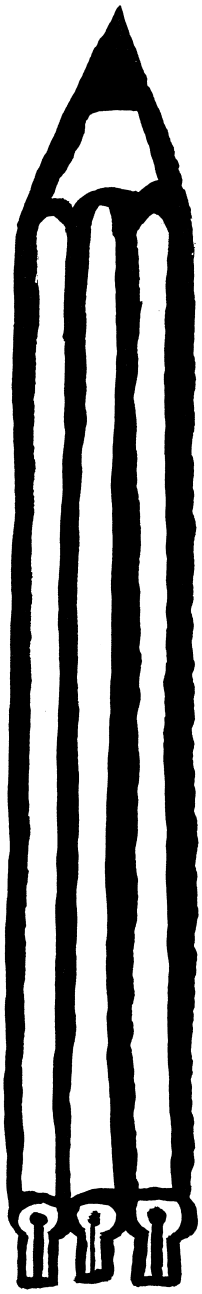
The Editor welcomes proposals for surveys, tutorials, and thematic issues of the Bulletin dedicated to currently hot topics, as well as suggestions for new regular sections.

Table of Contents

Announcements

EATCS Matters





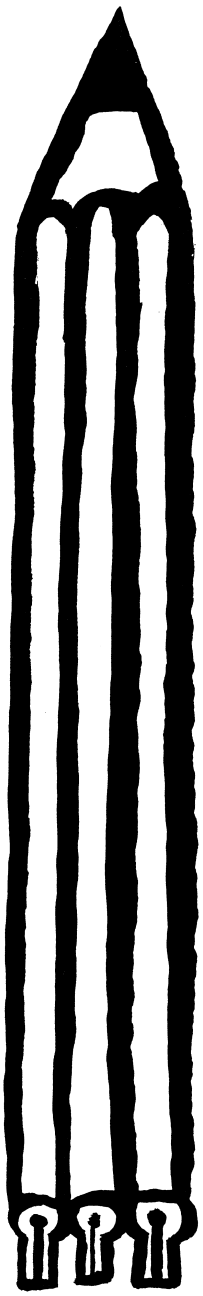
Dear colleagues,

As usual, the June issue of the Bulletin will be available just before ICALP, the flagship conference of the EATCS, which hosts the annual meeting of the council of our association and its general assembly. I hope that many of you will be at ICALP 2014, which has a mouth-watering scientific programme and an exciting collection of cultural and social events to boot. Thore Husfeldt and his team at the IT University in Copenhagen are working very hard on the final details of the organization of the 41st ICALP, which I am sure will be truly memorable.

Apart from the invited and contributed talks, ICALP 2014 will feature the presentation of the EATCS Award 2014 to Gordon Plotkin, of the Presburger Award 2014 to David Woodruff and of the Gödel Prize 2014 to Ronald Fagin, Amnon Lotem, and Moni Naor. Moreover, during the conference, we will honour the first group of EATCS Fellows, consisting of

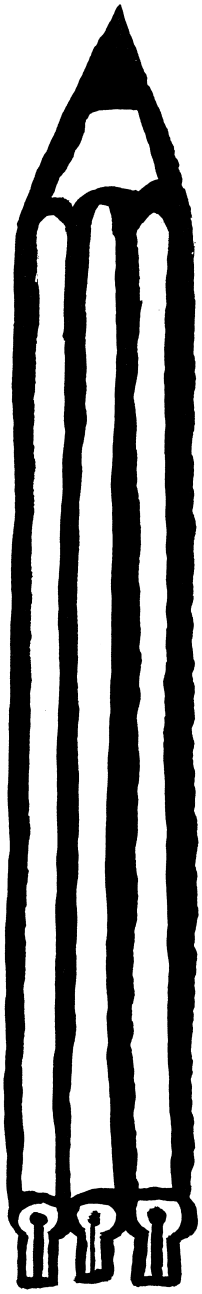
- *Susanne Albers (Technische Universität München, Germany) for "her contributions to the design and analysis of algorithms, especially online algorithms, approximation algorithms, algorithmic game theory and algorithm engineering";*
- *Giorgio Ausiello (Universitá di Roma La Sapienza, Italy) for "the impact of his scientific work in the field of algorithms and computational complexity and for his service to the scientific community";*

- *the late Wilfried Brauer (Technische Universität München, Germany) for "outstanding contributions to the foundation and organization of the European TCS community";*
- *Herbert Edelsbrunner (Institute of Science and Technology Austria and Duke University, USA) for "his tremendous impact on the field of computational geometry";*
- *Mike Fellows (Charles Darwin University, Australia) for "his role in founding the field of parameterized complexity theory, which has become a major subfield of research in theoretical computer science, and for being a leader in computer science education";*
- *Yuri Gurevich (Microsoft Research, USA) for "his development of abstract state machines and for outstanding contributions to algebra, logic, game theory, complexity theory and software engineering";*
- *Monika Henzinger (University of Vienna, Austria) for "being one of the pioneers of web algorithms, algorithms that deal with problems of the world wide web";*
- *Jean-Eric Pin (LIAFA, CNRS and University Paris Diderot, France) for "outstanding contributions to the algebraic theory of automata and languages in connection with logic, topology, and combinatorics and service to the European TCS community";*
- *Paul Spirakis (University of Liverpool, UK, and University of Patras, Greece)*



for "seminal papers on Random Graphs and Population Protocols, Algorithmic Game Theory, as well as Robust Parallel Distributed Computing"; and

- *Wolfgang Thomas (RWTH Aachen University, Germany) for "foundational contributions to the development of automata theory as a framework for modelling, analyzing, verifying and synthesizing information processing systems."*

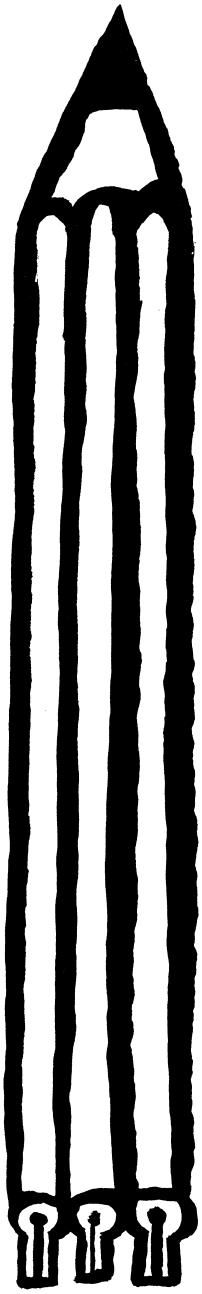


I thank the members of the award and fellow committees for their work in the selection of this stellar set of award recipients and fellows. It will be a great honour to celebrate the work of these colleagues during ICALP 2014.

The number of submissions for ICALP 2014 was a record 484 (319 for Track A, 106 for Track B and 59 for Track C). The number of submissions for Track A also set a new record for that track. The PCs for the three tracks, which were chaired by Elias Koutsoupias (Track A), Javier Esparza (Track B) and Pierre Fraigniaud (Track C), did a sterling job in the selection of the contributed papers for the conference and in the selection of the best papers and best student papers.

The invited talks and the talks by the award recipients at ICALP 2014 will be recorded and will be streamed live during the conference. For the first time, the general assembly of the EATCS will also be streamed live on the net and there will be a live Twitter feed, which will enable our members who are unable to attend the conference to take an active part in the event. I look forward to seeing the result of this experiment, which I do believe is worth trying for the sake of inclusiveness and openness.

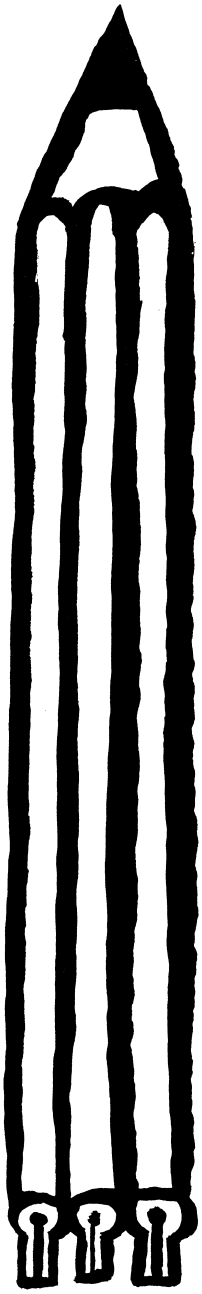
Since the February issue of the Bulletin was published, our community has lost Georgy Maximovich Adelson-Velsky (1922-2014)), Alberto Bertoni (1946-2014), Wilfried Brauer (1937-2014) and Robert McNaughton. Adelson-Velsky is best known for being the co-inventor of the AVL tree,



which was the first known balanced binary search tree data structure, in 1962. Bertoni was one of the fathers of theoretical computer science in Italy, a member of the council of the EATCS and one of the early founders, and former president, of the Italian Chapter of the EATCS. Brauer was one of the former presidents of the EATCS and one of the first authors in the emerging field of theoretical computer science in the 1960s and early 1970s. McNaughton was a pioneer and master of the field of automata and formal language theory. The community will miss them.

Apart from ICALP, the EATCS is involved in many initiatives and uses its (limited, alas) financial resources to support young researchers and meritorious activities in Theoretical Computer Science. By way of example, I remind you that the EATCS Young Researcher School Series, will kick off this year with a school on Automata, Logic and Games *organized by Tony Kucera. Moreover, soon after ICALP 2014, we will issue the first call for nominations for the EATCS Distinguished Dissertation Awards, which will be presented to two outstanding doctoral theses in theoretical computer science starting from 2015. Finally, the EATCS Council has decided to provide some modest financial support to the Conference on Computational Complexity (CCC), which, after an open discussion involving the members of the CCC community, recently decided to leave IEEE and to become an independent event.*

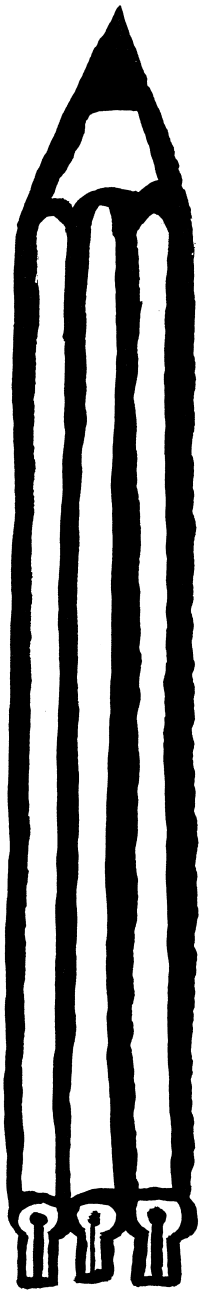
The above-mentioned activities are just a sample of the increasingly many ones in which the EATCS is involved. In addition,



we are also strengthening our ties with other sister organizations (such as the European Association for Computer Science Logic and the recently established ACM's Special Interest Group on Logic and Computation). In particular, we are working on stipulating reciprocity agreements with those organizations and on the establishment of new joint prizes. As usual, let me remind you that you are always most welcome to send me your comments, criticisms and suggestions for improving the impact of the EATCS on the theoretical-computer-science community at president@eatcs.org.

I hope that you will appreciate the steps that the council of the EATCS has taken on several fronts, even though there is still much more that we could do if we had suitable resources. I am truly grateful to our institutional sponsors and to our members for their support over the years. If you are not already a member, I hope that you will join the EATCS and encourage your colleagues and students to do so. The EATCS membership fee is low and, by becoming a member, you will contribute to the activities of our organization and will support the development of theoretical computer science, broadly construed. I look forward to seeing many of you in Copenhagen for ICALP 2014 and to discussing ways of improving the impact of the EATCS within the theoretical-computer-science community at the general assembly.

Luca Aceto, Reykjavik, Iceland
June 2014



Dear Reader,

I have just finished with my travel arrangements to Copenhagen for ICALP 2014. As you know, ICALP 2015 will be held in Kyoto, my city. Of course I know pretty well what ICALP meetings look like in general, but even so the trip this time has a bit of special meaning to me in the sense that I need to remind myself what is important to make the meeting more attractive and more comfortable.

You will meet several articles/reports related to EATCS/ICALP in the next October issue. So, this June issue may tend to be quiet (June is a rainy season and in fact quiet in Japan). Well, true. But this is even better to read technical stuff peacefully: We have five columns including two new ones: The Algorithmics Column by Gerhard Woeginger and The Concurrency Column by Nobuko Yoshida. Our community includes a lot of different disciplines and it is not very easy to write professional surveys so as to be accessible from all people. One easy answer to this problem is just to include relatively many surveys on different topics. Thus this issue is nice, which I am sure will allow you to spend a nice time in the next weekend.

Another specific point I would like to make is the "Book Introduction by the Authors" section. This issue includes the contribution by Stasys Jukna about his book "Boolean Function Complexity." If you are interested in circuit complexity, you definitely cannot miss this nice article (I have a small concern that some of you even feel that you already know all about the

contents and do not have to buy one...) I should strongly like to make this section regular and one of the features of our Bulletin. I need your help; looking forward to receiving your suggestions and/or information on books for this section.

See you in Copenhagen very soon!

Kazuo Iwama, Kyoto
June 2014

Wilfried Brauer (1937–2014) in memoriam

Personal recollections

At the end of February this year we received the sad tidings of the passing of *Wilfried Brauer*. Although not quite unexpected, the tidings still left us with a feeling of sorrow and longing. We had lost a close friend, a remarkable scientist and an influential administrator. Wilfried was one of the early pioneers of theoretical computer science in Europe. He was active in the founding stages of EATCS and the IFIP working group TC–1, and made significant contributions also to the working group TC–3. Through his activities as EATCS President, IFIP Vice President and the Chairman of the Gesellschaft für Informatik, as well as through his scientific work and that of his students, Wilfried Brauer made a lasting contribution to the theoretical computer science community. This is visible also in his many decorations, such as honorary doctorates from the University of Hamburg and the Freie Universität Berlin, Werner Heisenberg Medal and IFIP Isaac L. Auerbach Award.

However, the purpose of this writing is not to dwell on such formal matters. We would rather want to bring forward happenings and recollections from the forty years we had the privilege of knowing Wilfried and working with him.

Wilfried could handle difficult matters in a smooth and balanced way. As far as we remember, he never lost his temper. Another very characteristic feature of Wilfried was that age seemed to have no influence on his outer appearance. He was still in the new millennium the same joking boyish Wilfried we got to know in the early 70's.

Wilfried belonged to the early small European community working in theoretical computer science. We got to know Wilfried and his wife *Ute* at Oberwolfach meetings in the early 70's. Wilfried seemed to know everybody well and was interested in new emerging fields of study. Lindenmayer systems constituted such a field. Working in L systems, we got invitations to Hamburg. During such visits we also enjoyed hospitality in Brauers' home at Gustav Leo Strasse.

Most of our meetings with Wilfried were connected to the work with Springer–Verlag. The book series *EATCS Monographs in Theoretical Com-*

puter Science was launched at ICALP in Antwerp in 1983. The representatives of Springer–Verlag were then Gerhard Rossbach and Ingeborg Mayer. Rossbach was replaced by Hans Wössner at the end of the 80's.

Wilfried, the two of us and the two representatives of Springer-Verlag met a couple of times yearly, usually at ICALP conferences and in Grzegorz's home in Bilthoven. Most of the time Ute accompanied Wilfried. Then Grzegorz's wife *Maja* and Ute had a special "ladies' program".

The Bilthoven meetings gradually developed a specific format allowing ample time for work. Discussions continued during the Dutch breakfast by Grzegorz and lunch with Maja's "monograph soup" as the main course. Our meetings often culminated with a magic show of Grzegorz. Wilfried joined the enthusiasm of the audience.

Wilfried's experience and personal connections were invaluable for the success of our book series. Especially in delicate matters he was able to provide us with important information by contacting appropriate referees.

Wilfried did not attend the last Bilthoven meeting in 2007. We sent him a picture of owls with the text *Bilthoven owls miss the other wise owl.*

Wilfried and Ute became our close family friends. For instance, they wrote a paper about the jeep problem with the subtitle *How to bring a birthday present to Salosauna.* The present, a small teddy bear, became the most precious toy for Arto's granddaughter and was named Wilfried. Wilfried and the bear Wilfried appeared together at the ICALP in Vienna in 1992, as seen from photos in the EATCS Bulletin. The bear Wilfried is still in good shape.

Wilfried was a great fan of classical music and attended concerts and opera performances with Ute. If there was an interesting performance in another country, the distance constituted no obstacle for them. Wilfried explained that it is often difficult for him to get rid of thoughts concerning work. During the overture of an opera he might still think about phone calls he has to make. But then everything else vanishes, and he is in the world of the opera. It was a superb present for Arto when Wilfried hosted his visit to the Bayreuth Festival in 2005.

Our dear friend Wilfried, we miss you. We miss your wise advice and your relaxing dinners where fish dishes had to be excluded. We miss our discussions about professional, as well as other matters. *Sit tibi terra levis.*

Bilthoven and Turku, March 2014

Grzegorz Rozenberg *Arto Salomaa*

OBITUARY

ALBERTO BERTONI

(1946-2014)

Alberto Bertoni passed away on February 10, 2014, after a long struggle with a cancer that resisted surgery and therapy. This is a tremendous loss for his wife Luciana, for his friends and colleagues, and for the community of theoretical computer science in which he played a prominent role.

Alberto was born in Barlassina, Italy, the 17^{th} of July, 1946. He obtained the degree in Physics, cum laude, at the University of Milan, 22^{nd} July, 1970. He was Assistant Professor in Cybernetics at the Department of Physics, University of Milan, from 1976 to 1980. In 1981 he obtained a position as full professor in Computer Science and, after a short period at the University of Cosenza, he came back to Milan and was one of the founders of the Department of Information Sciences of the University of Milan and one of the organizers and first professors of the degree in Information Sciences, a degree that did not exist before in Milan.

In the booming decade from 1980 to 1990 the number of students rapidly increased to about 5000, and Alberto passionately devoted much of his energies to an intense and varied teaching activity. In 35 years he taught courses that covered many aspects of algorithms and theoretical computer science, but also of combinatorics and discrete mathematics. These course ranged from first or second year classes on Algebra, Algorithms and Data Structures, Analysis and Design of Algorithms, Formal Languages and Compilers, to more advanced courses on Signal Processing, Neural Networks, Computability, to very specialized courses on research related topics in the areas of Structural Complexity Theory, Algorithms and Combinatorics, Signal Processing, Combinatorial Optimization, Game Theory for the PhD students in Computer Science.

His lectures were always well prepared and fascinating, and he was able to captivate the students attention even when explaining very complex topics.

He was advisor of more than 200 laurea theses in the degrees of Computer Science, Mathematics, Physics and more than 20 PhD theses in Computer Science, Mathematics and Engineering.

To his disciples and advisees Alberto taught not only the notions, the methods and the technicalities of the different topics of theoretical computer science, but above all the love for pure research itself.

In fact, Alberto was a very gifted researcher, guided by his curiosity and enthusiasm, with a rare capability of identifying interesting research problems, formalizing them, and finding solutions.

His research activity covered an impressive range in the area of Theoretical Computer Science: in computability and complexity, probabilistic and quantum machines, formal languages, computational learning, theoretical aspects of neural networks and genetic models. This research is documented by more than 120 papers in international journals and conference proceedings. In particular, in complexity theory he solved open problems on probabilistic automata and studied problems of simulation among computational models (for instance he proved that the enumeration problems in the class #-PSPACE can be solved by arithmetic RAMs with a polynomial number of operations) and the classification of counting and ranking problems. Furthermore, he studied the minimum amount of resources such as space, head inversions and non determinism degree needed to recognize non regular languages in some models of Turing Machines. Similar techniques were applied to picture languages, showing that the class of unary tiling recognizable picture languages is characterized by languages accepted by Turing machines with bounds on space and head inversions.

An important example of his ability to apply deep mathematical concepts to problems arising in computer science is his proposal to use the theory of free partially commutative monoids to model concurrent processes. This idea linked the theory of trace languages to the more general context of formal languages, to

which Alberto and his research group contributed many results on membership problems and on characterization of classes of trace languages.

In the area of random generation and counting algorithms, he designed a linear algorithm for random generation of words in regular languages with fixed number of occurrences of the symbols, and also gave results on asymptotic estimation of the number of words in regular languages with fixed number of occurrences of the symbols, with applications to pattern statistics. More recently, he introduced new models of quantum automata, and compared them with stochastic automata, exploring the advantages of using quantum devices in computation over probabilistic models. Furthermore, he gave significant contributions to the area of bioinformatics, designing and experimenting supervised and unsupervised learning algorithms based on random projections with application to biomolecular data clustering.

The Italian and European community of theoretical computer science owe much to Alberto also for his promotional and organizing activity.

He was co-promoter of the Italian Chapter of the European Association of Theoretical Computer Science (ICh-EATCS) and first President of the Chapter for 6 years. He was for 6 years the Italian member in the Council of the European Association of Theoretical Computer Science.

He contributed to the birth of the Italian Society for Neural Networks (SIREN), and was member of its Scientific Council. He was member of the Academic Senate for the revision of the Statute of the University of Milan. He was member of Scientific Committee of the Institute for Applied Mathematics and Informatics of the CNR (IAMI-CNR), and member IFIP TC1. He was the Director of the PhD school in Computer Science, Milano-Torino, for 4 years and President of the Council of the degree in Computer Science, University of Milan, for 6 years.

He was the Director of the Department of Information Sciences, University of Milan, for 6 years (2003-09). He was member of the programme committee of several International Conferences (CAAP, STACS, AdPeNets, DLT, MFCS, SOFSEM, . . .), and of the Editorial Board of Theoretical Informatics and Applications.

Those who had the fortune to study and work with Alberto will always remember his strong personality, his honesty, his warm friendship, his scientific generosity, his clarity and originality, and also his passion for the mountains which he transmitted to many of his students and collaborators.

Giancarlo Mauri and Nicoletta Sabadini
June 2014

Institutional Sponsors

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

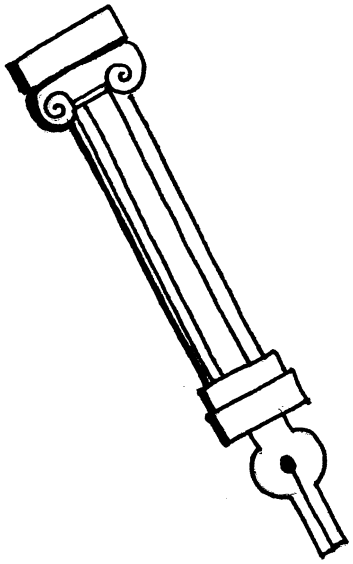
CWI, Centum Wiskunde & Informatica
Amsterdam, The Netherlands

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

BEATCS no 113

The Complexity of Valued Constraint Satisfaction

Peter Jeavons* Andrei Krokhin† Stanislav Živný‡

Abstract

We survey recent results on the broad family of problems that can be cast as valued constraint satisfaction problems. We discuss general methods for analysing the complexity of such problems, give examples of tractable cases, and identify general features of the complexity landscape.

1 Introduction

Computational problems from many different areas involve finding an assignment of values to a set of variables, where that assignment must satisfy some specified feasibility conditions and optimise some specified objective function. In many such problems the objective function can be represented as a sum of functions, each of which depends on some subset of the variables. Examples include: Gibbs energy minimisation, Markov Random Fields (MRF), Conditional Random Fields (CRF), Min-Sum Problems, Minimum Cost Homomorphism, Constraint Optimisation Problems (COP) and Valued Constraint Satisfaction Problems (VCSP) [6, 23, 68, 85, 87, 89].

We focus in this article on a generic framework for such problems that captures their general form. Bringing all such problems into a common framework draws attention to common aspects that they all share, and allows a very general algebraic approach for analysing their complexity to be developed. The primary motivation for this line of research is to understand the general picture of complexity within this general framework, rather than to develop specialised techniques for specific applications. We will give an overview of this algebraic approach, and the results that have been obtained by using it.

*Department of Computer Science, University of Oxford, Peter.Jeavons@cs.ox.ac.uk

†School of Engineering and Computing Sciences, University of Durham, Andrei.Krokhin@durham.ac.uk (Andrei Krokhin is supported by the UK EPSRC grant EP/H000666/1)

‡Department of Computer Science, University of Oxford, standa@cs.ox.ac.uk
(Stanislav Živný is supported by a Royal Society University Research Fellowship)

The generic framework we use is the *valued constraint satisfaction problem* (VCSP), defined formally as follows. Throughout the paper, let D be a fixed finite set and let $\overline{\mathbb{Q}} = \mathbb{Q} \cup \{\infty\}$ denote the set of rational numbers with (positive) infinity.

Definition 1. We denote the set of all functions $\phi : D^m \to \overline{\mathbb{Q}}$ by $\mathbf{\Phi}_D^{(m)}$ and let $\mathbf{\Phi}_D = \bigcup_{m \geq 1} \mathbf{\Phi}_D^{(m)}$. We will often call the functions in $\mathbf{\Phi}_D$ *cost functions* over D.

Let $V = \{x_1, \ldots, x_n\}$ be a set of variables. A *valued constraint* over V is an expression of the form $\phi(\mathbf{x})$ where $\mathbf{x} \in V^m$ and $\phi \in \mathbf{\Phi}_D^{(m)}$. The number m is called the *arity* of the constraint, the function ϕ is called the *constraint function*, and the tuple $\mathbf{x}$ the *scope* of the constraint.

We will call the elements of D *labels* (for variables), and say that the cost functions in $\mathbf{\Phi}_D$ take *values*.

Definition 2. An instance of the *valued constraint satisfaction problem* (VCSP) is specified by a finite set $V = \{x_1, \ldots, x_n\}$ of variables, a finite set D of labels, and an *objective function* Φ expressed as follows:

$$\Phi(x_1, \ldots, x_n) = \sum_{i=1}^{q} \phi_i(\mathbf{x}_i) \tag{1}$$

where each $\phi_i(\mathbf{x}_i)$, $1 \leq i \leq q$, is a valued constraint over V. Each constraint can appear multiple times in Φ.

The goal is to find an *assignment* of labels to the variables (or *labelling*) that minimises Φ.

Note that the value of the function Φ for any assignment of labels to the variables in V is given by the sum of the values taken by the constraints; this value will sometimes be called the *cost* of the assignment. An infinite value for any constraint indicates an infeasible assignment.

If the constraint functions in some VCSP instance are finite-valued, i.e., take only finite values, then every assignment is feasible, and the problem is to identify an assignment with minimum possible cost (i.e., we need to deal only with the optimisation issue). On the other hand, if each constraint function in an instance takes only two values: one finite value (possibly specific to the constraint) and ∞, then all feasible assignments are equally good, and so the only question is whether any such assignment exists (i.e., we need to deal only with the feasibility issue). If we have neither of the above cases then we need to deal with both feasibility and optimisation.

In Section 2 we give examples to show that many standard combinatorial optimisation problems can be conveniently expressed in the VCSP framework. In Section 3 we define certain algebraic properties of the constraints that can be used

to identify many tractable cases. Section 4 describes the basics of a recently developed general algebraic theory for analysing the complexity of different forms of valued constraints. In Section 5 we use this algebraic theory to identify several tractable and intractable cases, and in Section 6 we discuss approximation. In Section 7 we discuss the oracle model for representing the objective function. Finally, Section 8 gives a brief summary and identifies some open problems.

2 Problems and frameworks captured by the VCSP

In this section we will give examples of specific problems and previously studied frameworks that can be expressed as VCSPs with restricted forms of constraints.

Definition 3. Any set $\Gamma \subseteq \mathbf{\Phi}_D$ is called a *valued constraint language* over D, or simply a *language*. We will denote by VCSP(Γ) the class of all VCSP instances in which the constraint functions are all contained in Γ.

Valued constraint languages may be infinite, but it will be convenient to follow [11, 17] and define the complexity of a valued constraint language in terms of its finite subsets. We assume throughout that P$\neq$NP.

Definition 4. A valued constraint language Γ is called *tractable* if VCSP(Γ') can be solved (to optimality) in polynomial time for every finite subset $\Gamma' \subseteq \Gamma$, and Γ is called *intractable* if VCSP(Γ) is NP-hard for some finite $\Gamma' \subseteq \Gamma$.

One advantage of defining tractability in terms of finite subsets is that the tractability of a valued constraint language is independent of whether the cost functions are represented explicitly (say, via full tables of values, or via tables for the finite-valued parts) or implicitly (via oracles).

Example 5 (NAE-SAT)**.** Let $D = \{0, 1\}$ and let Γ_{nae} be the language that contains just the single ternary cost function $\phi_{\mathsf{nae}} : D^3 \to \overline{\mathbb{Q}}$ defined by

$$\phi_{\mathsf{nae}}(x, y, z) \stackrel{\text{def}}{=} \begin{cases} \infty & \text{if } x = y = z \\ 0 & \text{otherwise} \end{cases}.$$

The problem VCSP(Γ_{nae}) is exactly the Not-All-Equal Satisfiability problem, also known as the 3-Uniform Hypergraph 2-Colourability problem. This problem is NP-hard [33], so Γ_{nae} is intractable.

Example 6 (Max-k-Cut)**.** Let Γ_{xor} be the language that contains just the single binary cost function $\phi_{\mathsf{xor}} : D^2 \to \overline{\mathbb{Q}}$ defined by

$$\phi_{\mathsf{xor}}(x, y) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}.$$

The problem VCSP(Γ_{xor}) corresponds to the problem of minimising the number of monochrome edges in a k-colouring (where $k = |D|$) of the graph G formed by the scopes of the constraints. This problem is known as the Maximum k-Cut problem (or simply Max-Cut when $|D| = 2$), and is NP-hard [33].

Hence, for any choice of D, the language Γ_{xor} is intractable.

Example 7 (Potts model)**.** Let Γ_{Potts} be the language that contains all unary cost functions and the single binary cost function $\phi_{\mathsf{Potts}}\colon D^2 \to \overline{\mathbb{Q}}$ defined by

$$\phi_{\mathsf{Potts}}(x,y) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } x \neq y \end{cases}.$$

The problem VCSP(Γ_{Potts}) corresponds to finding the minimum energy state of the Potts model from statistical mechanics (with external field) [72]. This model is also used as the basis for a standard Markov Random Field approach to a wide variety of problems in machine vision [6]. For $|D| = 2$, the function ϕ_{Potts} is submodular (see Example 18) and we will show that this implies that Γ_{Potts} is tractable. For $|D| > 2$, Γ_{Potts} is intractable as it includes, as a special case, the multiway cut problem, which is NP-hard [27].

Example 8 ((s,t)-Min-Cut)**.** Let $G = (V, E)$ be a directed weighted graph such that for every $(u, v) \in E$ there is a weight $w(u, v) \in \mathbb{Q}_{\geq 0}$ and let $s, t \in V$ be distinguished source and target nodes. Recall that an (s,t)-*cut* C is a subset of vertices V such that $s \in C$ but $t \notin C$. The weight, or the size, of an (s,t)-cut C is defined as $\sum_{(u,v)\in E, u\in C, v\notin C} w(u,v)$. The (s,t)-Min-Cut problem consists in finding a minimum-weight (s,t)-cut in G. We can formulate the search for a minimum-weight (s,t)-cut in G as a VCSP instance as follows.

Let $D = \{0, 1\}$. For any label $d \in D$ and cost $c \in \overline{\mathbb{Q}}$, we define

$$\eta_d^c(x) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } x = d \\ c & \text{if } x \neq d \end{cases}.$$

For any weight $w \in \mathbb{Q}_{\geq 0}$, we define

$$\phi_{\mathsf{cut}}^w(x,y) \stackrel{\text{def}}{=} \begin{cases} w & \text{if } x = 0 \text{ and } y = 1 \\ 0 & \text{otherwise} \end{cases}.$$

We denote by Γ_{cut} the set $\{\eta_0^\infty, \eta_1^\infty\} \cup \{\phi_{\mathsf{cut}}^w \mid w \in \mathbb{Q}_{\geq 0}\}$. A minimum-weight (s,t)-cut in a graph G with set of nodes $V = \{x_1, \ldots, x_n\}$ corresponds to the set of variables assigned the label 0 in a minimal cost assignment to the VCSP instance defined by

$$\Phi(x_1, \ldots, x_n) \stackrel{\text{def}}{=} \eta_0^\infty(s) + \eta_1^\infty(t) + \sum_{(x_i,x_j)\in E} \phi_{\mathsf{cut}}^{w(x_i,x_j)}(x_i, x_j).$$

The unary constraints ensure that the source and target nodes must be assigned the labels 0 and 1, respectively, in any minimal cost assignment.

Furthermore, it is an easy exercise to show that any instance of VCSP(Γ_{cut}) on n variables can be solved in $O(n^3)$ time by a reduction to (s,t)-Min-Cut and then using the standard algorithm [35]. Hence Γ_{cut} is tractable.

Example 9 (Minimum Vertex Cover)**.** The Minimum Vertex Cover problem asks for a minimum size set W of vertices in a given graph $G = (V, E)$ such that each edge in E has at least one endpoint in W. This problem is NP-hard [33].

Let $D = \{0, 1\}$. We define

$$\phi_{\mathsf{vc}}(x, y) \stackrel{\text{def}}{=} \begin{cases} \infty & \text{if } x = y = 0 \\ 0 & \text{otherwise} \end{cases}.$$

We denote by Γ_{vc} the language $\{\phi_{\mathsf{vc}}, \eta_0^1\}$, where η_0^1 is the function defined in Example 8 that imposes unit cost for any variable assigned the label 1. A minimum vertex cover in a graph G with set of vertices $V = \{x_1, \ldots, x_n\}$ corresponds to the set of vertices assigned the label 1 in some minimum cost assignment to the VCSP(Γ_{vc}) instance defined by

$$\Phi(x_1, \ldots, x_n) \stackrel{\text{def}}{=} \sum_{x_i \in V} \eta_0^1(x_i) + \sum_{(x_i, x_j) \in E} \phi_{\mathsf{vc}}(x_i, x_j).$$

The binary constraints ensure that in any minimal cost assignment at least one endpoint of each edge belongs to the vertex cover.

Furthermore, it is easy to convert any instance of VCSP(Γ_{vc}) to an equivalent instance of Minimum Vertex Cover by repeatedly assigning the label 1 to all variables which do not appear in the scope of any unary constraints and removing these variables and all constraints involving them. Hence Γ_{vc} is intractable.

We will now show how several broad frameworks previously studied in the literature can be expressed as special cases of the VCSP with restricted languages. We will discuss algorithms and complexity classifications for them in Section 5.

Example 10 (CSP)**.** The standard constraint satisfaction problem (CSP) over any fixed set of possible labels D can be seen as the special case of the VCSP where all cost functions take only the values 0 or ∞, representing allowed (satisfying) and disallowed tuples, respectively. Such constraints and cost functions are sometimes called *crisp*. In other words, the CSP can be seen as VCSP(Γ_{crisp}), where Γ_{crisp} is the language consisting of all cost functions on some fixed set D with range $\{0, \infty\}$. Note that the CSP can also be cast as the homomorphism problem for relational structures [29] (cf. Example 11).

Since the CSP includes many known NP-hard problems, such as NAE-SAT (Example 5) and Graph-3-Colouring, the language Γ_{crisp} is clearly intractable. However, many tractable subsets of Γ_{crisp} have been identified [77, 52, 29, 11, 7, 12, 49, 3, 4], mostly through an algebraic approach whose extension we discuss in Section 4. There are many surveys on the complexity of the CSP, see the books [25, 26], and also [14, 42].

Feder and Vardi conjectured that the CSP exhibits a *dichotomy*: that is, every finite language $\Gamma \subseteq \Gamma_{\mathsf{crisp}}$ is either tractable or intractable [29], thus excluding problems of intermediate complexity, as given by Ladner's Theorem (assuming P≠NP) [66]. The *Algebraic Dichotomy* conjecture, which we state formally and discuss in Section 5, specifies the precise boundary between tractable and intractable crisp languages [11].

Example 11 (Graph Homomorphism)**.** Given two digraphs $G = (V(G), E(G))$ and $H = (V(H), E(H))$, a mapping $f : V(G) \to V(H)$ is a *homomorphism* from G to H if f preserves edges, that is, $(u, v) \in E(G)$ implies $(f(u), f(v)) \in E(H)$.

The problem whether an input digraph G admits a homomorphism to a fixed digraph H is also known as the H-Colouring problem and has been actively studied in graph theory [41, 42].

For any graph H, let $D = V(H)$ and let Γ_H be the language that contains just the single binary cost function $\phi_H : D^2 \to \overline{\mathbb{Q}}$ defined by

$$\phi_H(x, y) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } (x, y) \in E(H) \\ \infty & \text{otherwise} \end{cases} .$$

For any digraph H, the problem VCSP(Γ_H), which is a special case of the CSP (Example 10), corresponds to the H-colouring problem, where the input graph G is given by the scopes of the constraints. If we add all unary crisp functions to Γ_H then the resulting VCSP is known as List H-Colouring [41, 42].

It is known that both the Feder-Vardi conjecture and the Algebraic Dichotomy conjecture are equivalent to their restrictions to the H-colouring problem [13, 29].

Example 12 (Max-CSP)**.** An instance of the (weighted) maximum constraint satisfaction problem (Max-CSP) is an instance of the CSP where the goal is to maximise the (weighted) number of satisfied constraints.

When seeking the optimal solution, maximising the number of satisfied constraints is the same as minimising the number of unsatisfied constraints. Hence for any instance Φ of the Max-CSP, we can define a corresponding VCSP instance Φ' in which each constraint c of Φ is associated with a constraint over the same scope in Φ' which assigns cost 0 to tuples allowed by c, and cost 1 to tuples disallowed by c. It follows that Max-CSP is equivalent to VCSP(Γ_{Max}), where Γ_{Max} is the language consisting of cost functions whose values are restricted to zero and one.

For $D = \{0, 1\}$, the complexity of all subsets of Γ_{Max} has been completely classified in [58]. Initial results for languages over arbitrary finite sets appeared in [15]. A complete complexity classification will be discussed in Section 5.

Example 13 (Min-Cost-Hom)**.** Let Γ_{unary} consist of all unary cost functions and let $\Gamma_{\mathsf{mc}} = \Gamma_{\mathsf{crisp}} \cup \Gamma_{\mathsf{unary}}$ (where Γ_{crisp} is defined in Example 10). Problems of the form VCSP(Γ) with $\Gamma \subseteq \Gamma_{\mathsf{mc}}$ have been studied under the name of the Minimum-Cost Homomorphism problem (or Min-Cost-Hom) [39, 43, 81, 80, 85, 86]. Note that the first three of these papers assume that $\Gamma_{\mathsf{unary}} \subseteq \Gamma$, while the last three do not. In [39, 43] Γ is assumed to be of the form $\{\phi_H\} \cup \Gamma_{\mathsf{unary}}$, where ϕ_H is a binary crisp cost function, as in Example 11.

In any instance of VCSP(Γ_{mc}), the crisp constraints specify the CSP part, i.e., the feasibility aspect of the problem, while the unary constraints specify the optimisation aspect. More precisely, the unary constraints specify the costs of assigning labels to individual variables. Complexity classifications for special cases of Min-Cost-Hom will be discussed in Section 5.

Example 14 (Min-Ones)**.** An instance of the Boolean Minimum Ones (Min-Ones) problem is an instance of the CSP over $D = \{0, 1\}$ where the goal is to satisfy all constraints and minimise the number of variables assigned the label 1. Such instances correspond to Min-Cost-Hom instances over $\{0, 1\}$ in which all unary constraints are of the form η_0^1 as defined in Example 8 (which impose a unit cost for any variables assigned the label 1). A classification of the complexity of all subsets of this language was obtained in [25].

Example 15 (Min-Sol)**.** The Minimum Solution problem (Min-Sol) [53, 54] is a generalisation of Min-Ones from Example 14 to larger sets of labels where the only allowed unary cost function is a particular finite-valued injective function. Thus, this problem is also a subproblem of Min-Cost-Hom. Known complexity classifications for Min-Sol problems will be discussed in Section 5.

3 Polymorphisms and weighted polymorphisms

To develop general tools to classify the complexity of different valued constraint languages, we will now define certain algebraic properties of cost functions.

A function $f : D^k \to D$ is called a k-ary *operation* on D. The k-ary *projections*, defined for all $1 \leq i \leq k$, are the operations $\mathrm{e}_i^{(k)}$ such that $\mathrm{e}_i^{(k)}(x_1, \ldots, x_k) = x_i$. For any tuples $\mathbf{x_1}, \ldots, \mathbf{x_k} \in D^m$, we denote by $f(\mathbf{x_1}, \ldots, \mathbf{x_k})$ the tuple in D^m obtained by applying f to $\mathbf{x_1}, \ldots, \mathbf{x_k}$ componentwise.

Any valued constraint language Γ defined on D can be associated with a set of operations on D, known as the polymorphisms of Γ, and defined as follows.

Definition 16 (Polymorphism)**.** Let $\phi : D^m \to \overline{\mathbb{Q}}$ be a cost function and let $\mathrm{Feas}(\phi) = \{\mathbf{x} \in D^m \mid \phi(\mathbf{x}) \text{ is finite}\}$ be the *feasibility relation* of ϕ. We say that an operation $f : D^k \to D$ is a *polymorphism* of ϕ if, for any $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_k} \in \mathrm{Feas}(\phi)$ we have that $f(\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_k}) \in \mathrm{Feas}(\phi)$.

For any valued constraint language Γ over a set D, we denote by $\mathrm{Pol}(\Gamma)$ the set of all operations on D which are polymorphisms of all $\phi \in \Gamma$. We denote by $\mathrm{Pol}^{(k)}(\Gamma)$ the k-ary operations in $\mathrm{Pol}(\Gamma)$.

Note that the projections are polymorphisms of all valued constraint languages.

For $\{0, \infty\}$-valued cost functions (relations) this notion of polymorphism corresponds precisely to the standard notion of polymorphism for relations [5, 52]. This notion of polymorphism has played a key role in the analysis of complexity for the CSP [52, 11]. However, for the analysis of the VCSP we need a more flexible notion that assigns weights to a collection of polymorphisms.

Definition 17 (Weighted Polymorphism)**.** Let $\phi : D^m \to \overline{\mathbb{Q}}$ be a cost function and let $C \subseteq \mathrm{Pol}^{(k)}(\phi)$ be a collection of k-ary polymorphisms. A function $\omega : C \to \mathbb{Q}$ is called a k-ary *weighted polymorphism* of ϕ on C if it satisfies the following conditions:

- $\sum_{f \in C} \omega(f) = 0$;
- if $\omega(f) < 0$, then f is a projection;
- for any $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_k} \in \mathrm{Feas}(\phi)$

$$\sum_{f \in C} \omega(f)\phi(f(\mathbf{x_1}, \ldots, \mathbf{x_k})) \;\leq\; 0\,. \tag{2}$$

We define $\mathrm{supp}(\omega) = \{f \mid \omega(f) > 0)$ to be the *positive support* of ω.

Remark. The definition of a weighted polymorphism can be re-stated in probabilistic terms, as follows. Consider Inequality (2) and assume that it is non-trivial, i.e., not all weights $\omega(f)$ are equal to 0. Let c be the smallest (negative) weight $\omega(f)$ that appears there. Add $\sum_{i=1}^{k} |c| \cdot \phi(\mathrm{e}_i^{(k)}(\mathbf{x_1}, \ldots, \mathbf{x_k})) = \sum_{i=1}^{k} |c| \cdot \phi(\mathbf{x}_i)$ to both sides of Inequality (2). Note that all weights of operations on the left-hand side are now non-negative. Normalise by dividing both sides by $|c| \cdot k$ and view the (new) weights of operations on the left-hand side as a probability distribution μ over a subset of $\mathrm{Pol}^{(k)}(\phi)$. We can then re-write Inequality (2) as follows:

$$\mathbb{E}_{f \sim \mu}[\phi(f(\mathbf{x_1}, \ldots, \mathbf{x_k}))] \leq \mathrm{avg}\{\phi(\mathbf{x_1}), \ldots, \phi(\mathbf{x_k})\}. \tag{3}$$

Thus, one can identify (non-trivial) k-ary weighted polymorphisms of ϕ with probability distributions μ over subsets of $\mathrm{Pol}^{(k)}(\phi)$ satisfying Inequality (3) for all $\mathbf{x_1}, \ldots, \mathbf{x_k} \in \mathrm{Feas}(\phi)$.

This is illustrated in Figure 1, which should be read from left to right. Let $C = \{f_1, \ldots, f_n\} \subseteq \mathrm{Pol}^{(k)}(\phi)$ and let μ be a probability distribution on C. Starting with the m-tuples $\mathbf{x_1}, \ldots, \mathbf{x_k}$, we first apply operations $f_1, \ldots, f_n$ to these tuples componentwise, thus obtaining the m-tuples $\mathbf{x'_1}, \ldots, \mathbf{x'_n}$. Inequality 3 amounts to comparing the average of the values of ϕ applied to the tuples $\mathbf{x_1}, \ldots, \mathbf{x_k}$, which corresponds to projections, with the weighted sum of the values of ϕ applied to the tuples $\mathbf{x'_1}, \ldots, \mathbf{x'_n}$, which is the expected value of $\phi(f(\mathbf{x_1}, \ldots, \mathbf{x_k}))$ as f is drawn from μ.

$$
\begin{array}{lllll}
\mathbf{x_1} & \mathbf{x_1}[1] & \mathbf{x_1}[2] & \ldots & \mathbf{x_1}[m] \\
\mathbf{x_2} & \mathbf{x_2}[1] & \mathbf{x_2}[2] & \ldots & \mathbf{x_2}[m] \\
\vdots & & & \vdots & \\
\mathbf{x_k} & \mathbf{x_k}[1] & \mathbf{x_k}[2] & \ldots & \mathbf{x_k}[m]
\end{array}
\quad \xrightarrow{\phi} \quad
\left.\begin{array}{l}
\phi(\mathbf{x_1}) \\ \phi(\mathbf{x_2}) \\ \vdots \\ \phi(\mathbf{x_k})
\end{array}\right\} \frac{1}{k} \sum_{i=1}^{k} \phi(\mathbf{x_i})
$$

$$
\begin{array}{lllll}
\mathbf{x'_1} = f_1(\mathbf{x_1}, \ldots, \mathbf{x_k}) & \mathbf{x'_1}[1] & \mathbf{x'_1}[2] & \ldots & \mathbf{x'_1}[m] \\
\mathbf{x'_2} = f_2(\mathbf{x_1}, \ldots, \mathbf{x_k}) & \mathbf{x'_2}[1] & \mathbf{x'_2}[2] & \ldots & \mathbf{x'_2}[m] \\
\vdots & & & \vdots & \\
\mathbf{x'_n} = f_n(\mathbf{x_1}, \ldots, \mathbf{x_k}) & \mathbf{x'_n}[1] & \mathbf{x'_n}[2] & \ldots & \mathbf{x'_n}[m]
\end{array}
\quad \xrightarrow{\phi} \quad
\left.\begin{array}{l}
\phi(\mathbf{x'_1}) \\ \phi(\mathbf{x'_2}) \\ \vdots \\ \phi(\mathbf{x'_n})
\end{array}\right\} \sum_{i=1}^{n} \Pr_{\mu}[f_i] \phi(\mathbf{x'_i})
$$

Figure 1: Probabilistic definition of a weighted polymorphism.

If ω is a weighted polymorphism of ϕ, then we say that ϕ *admits* ω as a weighted polymorphism. We say that a language Γ admits a weighted polymorphism ω if ω is a weighted polymorphism of every cost function $\phi \in \Gamma$.

Weighted polymorphisms were introduced in [19] and have allowed a general algebraic theory of complexity for valued constraints to be developed, as we will describe in Section 4.

Certain special kinds of weighted polymorphisms were introduced in earlier papers, but have now been subsumed by the more general theory described here. For example, the notion of a *fractional polymorphism* was introduced in [16]. For finite-valued functions, this notion coincides with the notion of a weighted polymorphism.

A more restricted form of weighted polymorphism was introduced earlier in [17] and is known as a *multimorphism*. This is essentially a k-ary weighted polymorphism where the values of $\omega(f)$ are all integers, and the values of $\omega(f)$ for projection operations are all equal to -1. Using the probabilistic view, this means that the probability of each operation in a k-ary weighted polymorphism is of the form ℓ/k where $\ell \in \mathbb{Z}$.

One can specify a k-ary multimorphism as a k-tuple $\mathbf{f} = \langle f_1, \ldots, f_k \rangle$ of k-ary operations f_i on D, where each operation f for which $\omega(f)$ is positive appears $\omega(f)$ times, and then the definition simplifies as follows: for all $\mathbf{x_1}, \ldots, \mathbf{x_k} \in D^m$,

$$\sum_{i=1}^{k} \phi(f_i(\mathbf{x_1}, \ldots, \mathbf{x_k})) \;\leq\; \sum_{i=1}^{k} \phi(\mathbf{x_i})\,. \tag{4}$$

Weighted polymorphisms (including the special cases of fractional polymorphisms and multimorphisms) have proved to be a valuable tool for identifying tractable valued constraint languages, as we will illustrate in this Section.

Example 18 (Submodularity)**.** For any finite set V, a rational-valued function h defined on subsets of V is called a *set function*. A set function h is called *submodular* if for all subsets S and T of V,

$$h(S \cap T) + h(S \cup T) \;\leq\; h(S) + h(T). \tag{5}$$

Submodular functions are a key concept in operational research and combinatorial optimisation (see, e.g. [30, 78, 84] for extensive information about them). They are often considered to be a discrete analogue of convex functions. Examples of submodular functions include cuts in graphs, matroid rank functions, and entropy functions. There are combinatorial algorithms for minimising submodular functions in polynomial time (see [78, 30], and also [51]).

If we set $D = \{0, 1\}$, then any set function h on V can be associated with a ($|V|$-ary) cost function ϕ defined on the characteristic vectors of subsets of V. The union and intersection operations on subsets correspond to the Min and Max operations on the associated characteristic vectors. Hence h is submodular if and only if the associated cost function ϕ satisfies the following inequality:

$$\phi(\mathrm{Min}(\mathbf{x}_1, \mathbf{x}_2)) + \phi(\mathrm{Max}(\mathbf{x}_1, \mathbf{x}_2)) - \phi(\mathbf{x}_1) - \phi(\mathbf{x}_2) \leq 0\,.$$

But this means that ϕ admits the 2-ary weighted polymorphism ω_{sub}, defined by:

$$\omega_{sub}(f) \stackrel{\text{def}}{=} \begin{cases} -1 & \text{if } f \in \{e_1^{(2)}, e_2^{(2)}\} \\ +1 & \text{if } f \in \{\mathrm{Min}, \mathrm{Max}\} \\ \;\;0 & \text{otherwise.} \end{cases}.$$

This is equivalent to saying that ϕ admits $\langle \mathrm{Min}, \mathrm{Max} \rangle$ as a multimorphism.

Example 19 (Generalised Submodularity)**.** Let D be a finite *lattice*, i.e., a partially ordered set, where each pair of elements $\{a, b\}$ has a least upper bound, $\vee(a, b)$, and a greatest lower bound, $\wedge(a, b)$. We denote by Γ_{sub} the set of all cost functions over D that admit $\langle \vee, \wedge \rangle$ as a multimorphism. Using a polynomial-time strongly combinatorial algorithm for minimising submodular functions, it was shown in [17] that Γ_{sub} is tractable when D is a totally ordered lattice (i.e., a *chain*). More general lattices will be discussed in Section 5 and Section 7.

Example 20 (Max)**.** We denote by Γ_{max} the set of all cost functions (over some fixed finite totally ordered set D) that admit $\langle \mathrm{Max}, \mathrm{Max} \rangle$ as a multimorphism, where $\mathrm{Max} : D^2 \to D$ is the binary operation returning the larger of its two arguments. Note that Γ_{max} includes all monotonic decreasing finite-valued cost functions, as well as some non-monotonic crisp cost functions [17]. It was shown in [17] that Γ_{max} is tractable.

Example 21 (Min)**.** We denote by Γ_{min} the set of all cost functions (over some fixed finite totally ordered set D) that admit $\langle \mathrm{Min}, \mathrm{Min} \rangle$ as a multimorphism, where $\mathrm{Min} : D^2 \to D$ is the binary operation returning the smaller of its two arguments. The tractability of Γ_{min} was established in [17].

Example 22 (Bisubmodularity)**.** For a given finite set V, bisubmodular functions are functions defined on pairs of disjoint subsets of V with a requirement similar to Inequality 5 (see [30, 71] for the precise definition). Examples of bisubmodular functions include rank functions of delta-matroids [30].

A property equivalent to bisubmodularity can be defined on cost functions on the set $D = \{0, 1, 2\}$. We define two binary operations Min_0 and Max_0 as follows:

$$\mathrm{Min}_0(x, y) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } 0 \neq x \neq y \neq 0 \\ \mathrm{Min}(x, y) & \text{otherwise} \end{cases},$$

$$\mathrm{Max}_0(x, y) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } 0 \neq x \neq y \neq 0 \\ \mathrm{Max}(x, y) & \text{otherwise} \end{cases}.$$

We denote by Γ_{bis} the set of finite-valued cost functions that admit $\langle \mathrm{Min}_0, \mathrm{Max}_0 \rangle$ as a multimorphism. The language Γ_{bis} can be shown to be tractable using the results of [71] (see also [30]).

The definitions of Min_0 and Max_0 still make sense when $D = \{0, 1, 2 \ldots, k\}$, $k \geq 3$. In that case, functions on D that admit $\langle \mathrm{Min}_0, \mathrm{Max}_0 \rangle$ as a multimorphism are called *k-submodular*; they were introduced in [46].

Example 23 (Skew Bisubmodularity)**.** Let $D = \{0, 1, 2\}$. Recall the definition of operations Min_0 and Max_0 from Example 22. We define

$$\mathrm{Max}_1(x, y) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } 0 \neq x \neq y \neq 0 \\ \mathrm{Max}(x, y) & \text{otherwise} \end{cases}.$$

A function ϕ: $D^m \to \overline{\mathbb{Q}}$ is called *α-bisubmodular* [48], for some real $0 < \alpha \leq 1$, if ϕ admits the weighted polymorphism ω defined by $\omega(\mathrm{Min}_0) = 1$, $\omega(\mathrm{Max}_0) = \alpha$, $\omega(\mathrm{Max}_1) = (1 - \alpha)$, and $\omega(e_1^{(2)}) = \omega(e_2^{(2)}) = -1$. Note that 1-bisubmodular functions are (ordinary) bisubmodular functions as defined in Example 22. It is shown in [48] that each distinct value of α is associated with a distinct class of α-bisubmodular functions. The tractability of α-bisubmodular valued constraint languages will be discussed in Section 5.

Example 24 ((Symmetric) Tournament Pair)**.** A binary operation $f : D^2 \to D$ is called a *tournament* operation if (i) f is commutative, i.e., $f(x, y) = f(y, x)$ for all $x, y \in D$; and (ii) f is conservative, i.e., $f(x, y) \in \{x, y\}$ for all $x, y \in D$. The *dual* of a tournament operation is the unique tournament operation g satisfying $x \neq y \Rightarrow g(x, y) \neq f(x, y)$.

A *tournament pair* is a pair $\langle f, g\rangle$, where both f and g are tournament operations. A tournament pair $\langle f, g\rangle$ is called *symmetric* if g is the dual of f.

Let Γ be an arbitrary language that admits a symmetric tournament pair as a multimorphism. It was shown in [18], by a reduction to the minimisation problem for submodular functions (cf. Example 19), that any such Γ is tractable. It is shown in [62] that any finite-valued language that admits a symmetric tournament pair multimorphism also admits the submodularity multimorphism with respect to some totally ordered lattice on D (cf. Example 19).

Now let Γ be an arbitrary language that admits any tournament pair as a multimorphism. It was shown in [18], by a reduction to the symmetric tournament pair case, that any such Γ is also tractable.

Example 25 (1-Defect)**.** Let b and c be two distinct elements of D and let $(D; <)$ be a partial order which relates all pairs of elements except for b and c. We call $\langle f, g\rangle$, where $f, g : D^2 \to D$ are two binary operations, a *1-defect* if f and g are both commutative and satisfy the following conditions:

- If $\{x, y\} \neq \{b, c\}$, then $f(x, y) = \mathrm{Min}(x, y)$ and $g(x, y) = \mathrm{Max}(x, y)$.
- If $\{x, y\} = \{b, c\}$, then $\{f(x, y), g(x, y)\} \cap \{x, y\} = \emptyset$, and $f(x, y) < g(x, y)$.

The tractability of languages that admit a 1-defect multimorphism was shown in [57], and was used in the classification of the Max-CSP over a four-element set (see Section 5).

Example 26 (Majority)**.** A ternary operation $f : D^3 \to D$ is called a majority operation if $f(x, x, y) = f(x, y, x) = f(y, x, x) = x$ for all $x, y \in D$.

Let $\mathbf{f} = \langle f_1, f_2, f_3\rangle$ be a triple of ternary operations such that f_1, f_2 and f_3 are all majority operations. Let $\phi : D^m \to \overline{\mathbb{Q}}$ be an m-ary cost function that admits $\mathbf{f}$ as a multimorphism. By Inequality (4), for all $\mathbf{x}, \mathbf{y} \in D^m$, $3\phi(\mathbf{x}) \leq \phi(\mathbf{x}) + \phi(\mathbf{x}) + \phi(\mathbf{y})$ and $3\phi(\mathbf{y}) \leq \phi(\mathbf{y}) + \phi(\mathbf{y}) + \phi(\mathbf{x})$. Therefore, if both $\phi(\mathbf{x})$ and $\phi(\mathbf{y})$ are finite, then we have $\phi(\mathbf{x}) \leq \phi(\mathbf{y})$ and $\phi(\mathbf{y}) \leq \phi(\mathbf{x})$, and hence $\phi(\mathbf{x}) = \phi(\mathbf{y})$. In other words, the range of ϕ is $\{c, \infty\}$, for some finite $c \in \overline{\mathbb{Q}}$.

Let Γ_{Mjty} be the set of all cost functions that admit as a multimorphism some triple $\mathbf{f} = \langle f_1, f_2, f_3\rangle$ of arbitrary ternary majority operations. The tractability of Γ_{Mjty} was shown in [17].

Example 27 (Minority)**.** A ternary operation $f : D^3 \to D$ is called a minority operation if $f(x, x, y) = f(x, y, x) = f(y, x, x) = y$ for all $x, y \in D$. Let Γ_{Mnty} be the set of cost functions that admit as a multimorphism some triple $\mathbf{f} = \langle f_1, f_2, f_3 \rangle$ of arbitrary ternary minority operations. A similar argument to the one in Example 26 shows that the cost functions in Γ_{Mnty} have range $\{c, \infty\}$, for some finite $c \in \overline{\mathbb{Q}}$. The tractability of Γ_{Mnty} was shown in [17].

Example 28 (MJN)**.** Let $\mathbf{f} = \langle f_1, f_2, f_3 \rangle$ be three ternary operations such that f_1 and f_2 are majority operations, and f_3 is a minority operation. Let Γ_{MJN} be the set of cost functions that admit $\mathbf{f}$ as a multimorphism. The tractability of Γ_{MJN} was shown in [63], generalising an earlier tractability result for a specific $\mathbf{f}$ of this form from [17].

Other tractable valued constraint languages defined by weighted polymorphisms include the so-called $L^\#$-convex languages [30], as well as the weakly and strongly tree-submodular languages defined in [60]. Hirai [45] recently introduced a framework of submodular functions on modular semilattices (defined by a type of weighted polymorphism) that generalises many examples given above, including standard submodularity, k-submodularity, skew bisubmodularity, and tree submodularity. See [45] for the natural, but somewhat technical, definition of this very general framework.

4 A general algebraic theory of complexity

We have seen in the previous section that many tractable cases of the VCSP can be defined by having a particular weighted polymorphism. The algebraic theory developed in [19] establishes that, in fact, every tractable valued constraint language can be exactly characterised by its weighted polymorphisms. This extends (parts of) the algebraic theory previously developed for the CSP [10, 11, 52] that has led to significant advances in understanding the landscape of complexity for the CSP over the last 10 years (e.g., [2, 3, 4, 7, 8, 9, 12, 49, 67]). In this section, we will give a brief overview of the main results of this new algebraic theory for the VCSP. We refer the reader to [19] for full details and proofs.

We first recall some basic terminology from universal algebra [5, 79]. We denote by $\mathbf{O}_D$ the set of all finitary operations on D and by $\mathbf{O}_D^{(k)}$ the k-ary operations in $\mathbf{O}_D$. Let $f \in \mathbf{O}_D^{(k)}$ and $g_1, \ldots, g_k \in \mathbf{O}_D^{(\ell)}$. The *superposition* of f and $g_1, \ldots, g_k$ is the ℓ-ary operation $f[g_1, \ldots, g_k]$ such that $f[g_1, \ldots, g_k](x_1, \ldots, x_\ell) = f(g_1(x_1, \ldots, x_\ell), \ldots, g_k(x_1 \ldots, x_\ell))$.

A set $F \subseteq \mathbf{O}_D$ is called a *clone* of operations if it contains all the projections on D and is closed under superposition. It is easy to verify that the set of operations $\mathrm{Pol}(\Gamma)$ is a clone. Clones are actively studied in universal algebra; for example,

all (countably many) clones on $D = \{0, 1\}$ are known, but the situation is known to be much more complicated for larger sets D (see, e.g., [5, 79]).

For each $F \subseteq \mathbf{O}_D$ we define Clone(F) to be the smallest clone containing F. For any clone C, we use $C^{(k)}$ to denote the k-ary operations in C.

Now we consider the effect of extending a valued constraint language $\Gamma \subseteq \mathbf{\Phi}_D$ to a possibly larger valued constraint language. We first define and study a notion of *expressibility* for valued constraint languages. This notion has played a key role in the analysis of complexity for the CSP and VCSP [11, 52, 17, 89].

Definition 29. We say that an m-ary cost function ϕ is *expressible* over a constraint language Γ if there exists a instance $\Phi \in \text{VCSP}(\Gamma)$ with variables $V = \{x_1, \ldots, x_n, y_1, \ldots, y_m\}$, such that

$$\phi(y_1, \ldots, y_m) = \min_{x_1, \ldots, x_n} \Phi(x_1, \ldots, x_n, y_1, \ldots, y_m)\,.$$

Definition 30. A valued constraint language $\Gamma \subseteq \mathbf{\Phi}_D$ is called a *weighted relational clone* if it is closed under expressibility, scaling by non-negative rational constants, and addition of rational constants. We define wRelClone(Γ) to be the smallest weighted relational clone containing Γ.

Theorem 31 ([19])**.** *A valued constraint language* Γ *is tractable if* wRelClone(Γ) *is tractable and intractable if* wRelClone(Γ) *is intractable.*

Example 32. By Theorem 31, and Examples 5 and 6, in order to show that Γ is an intractable language it is sufficient to show that ϕ_{nae} or ϕ_{xor} is in wRelClone(Γ). We discuss general reasons for intractability of constraint languages in Section 5.

We now develop tools that will allow an alternative characterisation of any weighted relational clone.

Definition 33. We define a k-ary *weighting* of a clone C to be a function $\omega : C^{(k)} \to \mathbb{Q}$ such that $\omega(f) < 0$ only if f is a projection and

$$\sum_{f \in C^{(k)}} \omega(f) = 0\,.$$

We denote by $\mathbf{W}_C$ the set of all possible weightings of C and by $\mathbf{W}_C^{(k)}$ the set of k-ary weightings of C.

Since a weighting is simply a rational-valued function satisfying certain linear inequalities it can be scaled by any non-negative rational to obtain a new weighting. Similarly, any two weightings of the same clone of the same arity can be added to obtain a new weighting of that clone.

The notion of superposition can also be extended to weightings in a natural way, by forming a superposition with each argument of the weighting, as follows.

Definition 34. For any clone C, any $\omega \in \mathbf{W}_C^{(k)}$ and any $g_1, g_2, \ldots, g_k \in C^{(\ell)}$, we define the *superposition* of ω and $g_1, \ldots, g_k$, to be the function $\omega[g_1, \ldots, g_k] : C^{(\ell)} \to \mathbb{Q}$ defined by

$$\omega[g_1, \ldots, g_k](f') \stackrel{\text{def}}{=} \sum_{\substack{f \in C^{(k)} \\ f[g_1,\ldots,g_k]=f'}} \omega(f)\,. \tag{6}$$

It follows immediately from the definition of superposition that the sum of the weights in any superposition $\omega[g_1, \ldots, g_k]$ is equal to the sum of the weights in ω, which is zero, by Definition 33. However, it is not always the case that an arbitrary superposition satisfies the other condition in Definition 33, that negative weights are only assigned to projections. Hence we make the following definition:

Definition 35. If the result of a superposition is a valid weighting, then that superposition will be called a *proper* superposition.

Definition 36. A *weighted clone*, W, is a non-empty set of weightings of some fixed clone C which is closed under non-negative scaling, addition of weightings of equal arity, and proper superposition with operations from C. The clone C is called the *support* of W.

Example 37. For any clone, C, the set $\mathbf{W}_C$ containing all possible weightings of C is a weighted clone with support C.

Example 38. For any clone, C, the set $\mathbf{W}_C^0$ containing all *zero-valued* weightings of C is a weighted clone with support C. $\mathbf{W}_C^0$ contains exactly one weighting of each possible arity, which assigns the value 0 to all operations in C of that arity.

Weighted clones were introduced only very recently and not much is known about them (in comparison with ordinary clones). Some initial study of weighted clones can be found in [19, 24].

Given a cost function ϕ, some weightings will satisfy the conditions of Definition 17, and hence be weighted polymorphisms of ϕ.

Definition 39. For any $\Gamma \subseteq \mathbf{\Phi}_D$, we denote by wPol($\Gamma$) the set of all weightings of Pol(Γ) which are weighted polymorphisms of all cost functions $\phi \in \Gamma$.

To define a mapping in the other direction, we need to consider the union of the sets $\mathbf{W}_C$ over all clones C on some fixed set D, which will be denoted $\mathbf{W}_D$. If we have a set $W \subseteq \mathbf{W}_D$ which may contain weightings of *different* clones over D, then we can extend each of these weightings with zeros, as necessary, so that they are weightings of the same clone C, where C is the smallest clone containing all the clones that are supports of weightings in W. For any set $W \subseteq \mathbf{W}_D$, we define wClone(W) to be the smallest weighted clone containing this set of extended weightings obtained from W.

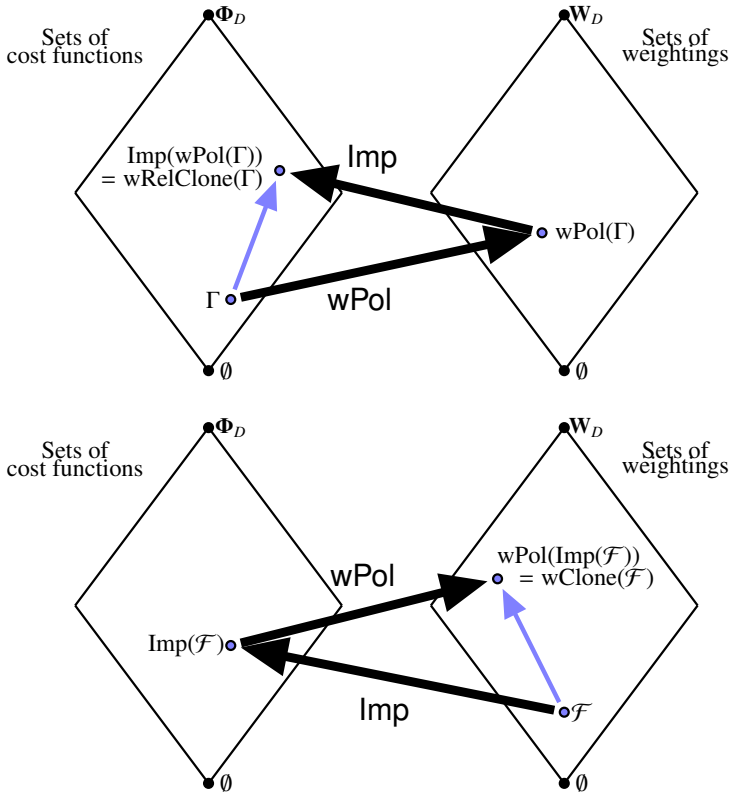


Figure 2: Galois connection between $\mathbf{\Phi}_D$ and $\mathbf{W}_D$.

Definition 40. For any $W \subseteq \mathbf{W}_D$, we denote by $\mathrm{Imp}(W)$ the set of all cost functions in $\mathbf{\Phi}_D$ which admit all weightings $\omega \in W$ as weighted polymorphisms [1].

It follows immediately from the definition of a Galois connection [5] that, for any set D, the mappings wPol and Imp form a Galois connection between $\mathbf{W}_D$ and $\mathbf{\Phi}_D$, as illustrated in Figure 2. A characterisation of this Galois connection for finite sets D is given by the following theorem from [19]:

Theorem 41 (Galois Connection for Valued Constraint Languages [19])**.**

1. *For any finite D, and any finite $\Gamma \subseteq \mathbf{\Phi}_D$,* $\mathrm{Imp}(\mathrm{wPol}(\Gamma)) = \mathrm{wRelClone}(\Gamma)$.
2. *For any finite D and any finite $W \subseteq \mathbf{W}_D$,* $\mathrm{wPol}(\mathrm{Imp}(W)) = \mathrm{wClone}(W)$.

[1] The name Imp is chosen to suggest that such cost functions are *improved* by weightings in W.

It follows that to identify all tractable valued constraint languages on a finite set D it is sufficient to study the possible weighted clones on D. This provides a new approach to the identification of tractable cases, which we hope will prove to be as successful as the algebraic approach has been in the study of the CSP.

The Galois connection described in Theorem 41 can be used to derive necessary conditions for tractability. It is shown in [19] that every tractable valued constraint language must have a weighted polymorphism that assigns positive weight to certain specific kinds of operations.

The algebraic theory of the CSP extends beyond clones to finite algebras and varieties of algebras (see [10, 11, 67], see also the surveys in [26]). This extension explains why the complexity of a (crisp) language is determined by the identities satisfied by its polymorphisms, which is why we usually define the relevant operations by identities. This extension was instrumental in obtaining most state-of-the-art results in this area (e.g. [2, 3, 4, 7, 8, 9, 12, 49, 67]). An initial study of a similar extension of the algebraic theory for the VCSP can be found in [73].

A valued constraint language Γ is called a *core* if every unary weighted polymorphism ω of Γ has the property that every operation $f \in \mathrm{supp}(\omega)$ is surjective. Intuitively, a valued constraint language Γ defined on D is a core if no label $x \in D$ can be removed without losing solutions. In other words, for every $a \in D$ there is an instance $\Phi_a \in \mathrm{VCSP}(\Gamma)$ such that a appears in every optimal solution to Φ_a [48]. Furthermore, a language Γ is called a *rigid core* if $\mathrm{Pol}^{(1)}(\Gamma)$ contains only the unary projection $e_1^{(1)}$. In this case, all operations in $\mathrm{Pol}(\Gamma)$ must be *idempotent*, i.e., satisfy the identity $f(x, \ldots, x) = x$.

Generalising the arguments used for the CSP [11] and finite-valued languages [48, 83], one can show that the search for tractable valued constraint languages can be restricted to languages that are rigid cores, see [73]. This technical restriction has very important implications because the structural theory of finite algebras works much better for idempotent operations (and idempotent algebras and varieties), see, e.g. [2, 3, 4, 7, 8, 12, 49, 67, 69]

5 Algorithms and complexity classifications

A curious feature of research into the tractability of constraint languages is that all languages known to be tractable have been shown tractable by using very few algorithmic techniques.

Despite many tractability results concerning crisp languages (i.e., the CSP), only two algorithmic techniques seem to be sufficient, and the applicability of each of them individually has been characterised by specific algebraic conditions.

The first technique is based on enforcing local consistency, which is a natural algorithm for dealing with (crisp) constraints. Roughly, this algorithm, for a given CSP instance, starts by adding a new constraint for each subset of variables of bounded size, the new constraints initially allowing all tuples. Then the algorithm repeatedly discards (i.e., disallows) tuples of labels in the new constraints that are inconsistent with at least one constraint in the instance. Eventually, either all assignments are discarded or else local consistency is established; this procedure takes polynomial time for any fixed D and any fixed bound on the size of subsets. The former case implies no feasible assignments. One says that a CSP is solved by local consistency if the latter case implies the existence of a feasible assignment. The power of local consistency (i.e., a precise characterisation of crisp languages that give rise to VCSP instances solvable by some form of local consistency) has recently been established [4, 8]. A k-ary ($k \geq 2$) idempotent operation $f : D^k \to D$ is called a *weak near-unanimity* operation if, for all $x, y \in D$,

$$f(y, x, x, \ldots, x) = f(x, y, x, x, \ldots, x) = f(x, x, \ldots, x, y).$$

Theorem 42 (Bounded Width [4, 8])**.** *Let* Γ *be a crisp language that is a rigid core.* VCSP(Γ) *is solvable by local consistency if and only if* Pol(Γ) *contains weak near-unanimity operations of all but finitely many arities.*

Remark. One of many equivalent forms of the Algebraic Dichotomy conjecture [11] mentioned in Example 10 is the following: A crisp language Γ that is a rigid core is tractable if and only if Pol(Γ) contains a weak near-unanimity operation. Crisp rigid cores Γ that do not satisfy this condition are known to be NP-complete [11]. This reformulation of the conjecture follows from [69] via [10] (see also [3]).

The second standard algorithmic technique for the CSP is based on the property of having a polynomial-sized representation (a generating set) for the solution set of any instance [9, 49]. Roughly, the algorithm works by starting from the empty set and adding constraints in an instance one by one while maintaining (in polynomial time) a small enough representation of the current solution set (of feasible assignments). At the end (i.e., after all constraints have been added), either this representation is non-empty and contains a solution to the instance or else there is no solution. In a way, this technique is a generalisation of Gaussian elimination. This algorithm is often called "few subpowers" because it is related to a certain algebraic property to do with the number of of subalgebras in powers of an algebra. The power of this algorithm was established in [49]. A k-ary ($k \geq 3$) operation $f : D^k \to D$ is called an *edge* operation if, for all $x, y \in D$,

$$f(y, y, x, x, \ldots, x) = f(y, x, y, x, x, \ldots, x) = x$$

and

$$f(x, x, x, y, x, \ldots, x) = f(x, x, x, x, y, x, \ldots, x) = f(x, \ldots, x, y) = x.$$

Theorem 43 (Few Subpowers [49])**.** *Let* Γ *be a crisp language. Then* VCSP(Γ) *is solvable by the few subpowers algorithm if* Pol(Γ) *contains an edge operation.*

The converse to this theorem is true in the following sense: the absence of edge operations from Pol(Γ) implies that the presence of small enough representations is not guaranteed, see [49] for details. Interestingly, the few subpowers algorithm makes use of the actual edge operations in its work (in contrast with bounded width, where the weak near-unanimity operations only guarantee correctness).

It is natural to try to extend the conditions characterising the applicability of these two algorithms to the VCSP, and to investigate whether valued constraint languages satisfying these algebraic conditions are also tractable. However, so far this approach is largely unexplored. Some forms of local consistency techniques have been generalised to the VCSP [20], but their power is not fully understood.

For the general VCSP another algorithm, based on linear programming, has been the most thoroughly investigated. Every VCSP instance has a natural linear programming relaxation called the *basic LP relaxation* (BLP). For an instance Φ defined by $\Phi(\mathbf{x}) = \sum_{i=1}^{q} \phi_i(\mathbf{x}_i)$, with set of variables V, the associated LP instance BLP(Φ) is defined as follows:

$$BLP(\Phi) \stackrel{\text{def}}{=} \min \quad \sum_{i=1}^{q} \sum_{\mathbf{s_i} \in D^{\mathbf{x}_i}} \phi_i(\mathbf{s_i})\, \lambda_{i,\mathbf{s_i}} \tag{7a}$$

$$\text{s.t.} \quad \sum_{\mathbf{s_i} \in D^{\mathbf{x_i}} \mid \mathbf{s_i}(x)=a} \lambda_{i,\mathbf{s_i}} = \mu_x(a), \quad 1 \le i \le q,\ x \in \mathbf{x_i},\ a \in D \tag{7b}$$

$$\sum_{a \in D} \mu_x(a) = 1, \quad x \in V \tag{7c}$$

$$\lambda_{i,\mathbf{s_i}} = 0, \quad 1 \le i \le q,\ \phi_i(\mathbf{s_i}) = \infty \tag{7d}$$

We minimise over the variables $\mu_x(a)$, where $x \in V$ and $a \in D$, and $\lambda_{i,\mathbf{s_i}}$, where $1 \le i \le q$ and $\mathbf{s_i} \in D^{\mathbf{x}_i}$, that take on real values in the interval $[0, 1]$. These variables can be seen as probability distributions on D and $D^{\mathbf{x}_i}$, respectively. The marginalization constraints (7b) impose that μ_x is the marginal of $\lambda_{i,\mathbf{s_i}}$, for each constraint and each variable x in the scope of that constraint. Note that terms in (7a) corresponding to (7d) are assumed to be equal to 0.

We remark that an LP relaxation of the VCSP, similar or closely related to (7), has been proposed independently by many authors; we refer the reader to [62] and the references therein.

Given a VCSP instance Φ, we say that BLP *solves* Φ if the optimal value of BLP(Φ) is equal to the optimal value of Φ. Moreover, we say that BLP solves a valued constraint language Γ if BLP solves every instance $\Phi \in$ VCSP(Γ). It is shown in [62] that in all cases where BLP solves Γ, a standard self-reduction

technique can be used to obtain an assignment that minimises any Φ in VCSP(Γ) in polynomial time. Hence if BLP solves Γ, then Γ is tractable.

The power of BLP for valued constraint languages was fully characterised in [82]. To state this result, we first introduce some further terminology about operations. A k-ary operation $f : D^k \to D$ is called *symmetric* if for every permutation π on $\{1, \ldots, k\}$, $f(x_1, \ldots, x_k) = f(x_{\pi(1)}, \ldots, x_{\pi(k)})$. A weighted polymorphism ω is called symmetric if supp(ω) is non-empty and contains symmetric operations only. Finally, we say that an operation f is *generated* from a set of operations $F \subseteq \mathbf{O}_D$ if $f \in \text{Clone}(F)$.

Theorem 44 (Power of BLP for Arbitrary Languages [82])**.** *Let Γ be a valued constraint language. Then the following are equivalent:*

1. *BLP solves Γ;*
2. *For every $k \geq 2$, Γ admits a k-ary symmetric weighted polymorphism;*
3. *For every $k \geq 2$, Γ admits a weighted polymorphism (not necessarily k-ary) ω_k such that $supp(\omega_k)$ generates a symmetric k-ary operation.*

It is unknown whether the conditions in Theorem 44 are decidable. Nevertheless, condition (3) has turned out to be very useful for proving the tractability of many valued constraint languages. A binary operation $f : D^2 \to D$ is called a *semilattice* operation if f is associative, commutative, and idempotent. Since any semilattice operation trivially generates symmetric operations of all arities, Theorem 44 shows that any valued constraint language with a binary weighted polymorphism whose positive support includes a semilattice operation is solvable using the BLP. This immediately implies that all of the following cases are solvable using the BLP, and hence tractable: languages with a (generalised) submodular multimorphism (Example 19), a bisubmodular multimorphism (Example 22), a symmetric tournament pair multimorphism (Example 24), or a skew bisubmodular weighted polymorphism (Example 23), or the weighted polymorphisms describing submodularity on modular semilattices [45]. Moreover, a not very difficult argument can be used to show that languages with a 1-defect multimorphism (Example 25) also satisfy condition (3) of Theorem 44 [82], and thus are tractable.

For valued constraint languages where the cost functions take only finite values, this result has been strengthened even further [82, 61], see also [62].

Theorem 45 (Power of BLP for Finite-Valued Languages [82, 61])**.** *Let Γ be a valued constraint language where every cost function takes only finite values. Then the following are equivalent:*

1. *BLP solves Γ;*

2. *For every* $k \geq 2$, Γ *admits a* k*-ary symmetric weighted polymorphism;*

3. *For some* $k \geq 2$, Γ *admits a* k*-ary symmetric weighted polymorphism;*

4. Γ *admits a binary symmetric weighted polymorphism;*

5. Γ *admits a weighted polymorphism* ω *such that* $\mathrm{supp}(\omega)$ *generates a symmetric operation.*

We mentioned above that the tractability of constraint languages seems to come from very few techniques. Interestingly, the hardness of constraint languages also seems to come from very few specific hard problems! Recall the functions ϕ_{nae} and ϕ_{xor} on $\{0,1\}$, from Examples 5 and 6, corresponding to the NP-hard problems NAE-SAT and Max-Cut.

The hardness of VCSP($\{\phi_{\mathsf{nae}}\}$) generalises in an obvious way to any problem VCSP($\{\phi\}$) over any set D, where ϕ is defined as follows: choose a subset $X \subseteq D$ with $|X| > 1$ and a surjective function $h : X \to \{0,1\}$, and let $\phi(x,y,z) = \phi_{\mathsf{nae}}(h(x),h(y),h(z))$ if $(x,y,z) \in X^3$ and $\phi(x,y,z) = \infty$ otherwise. Call such functions NAE-like. By Theorem 31, every language Γ such that wRelClone (Γ) contains a NAE-like function is intractable. Moreover, *every* crisp core language Γ known to be NP-complete satisfies this condition [11]. In other words, the ability to express ϕ_{nae} is the only known reason for a crisp core language to be NP-hard, and the only reason for this if the Algebraic Dichotomy conjecture holds.

Now let ϕ be a binary cost function over D such that, for some distinct $a, b \in D$, $\mathrm{argmin}(\phi) = \{(a,b),(b,a)\}$ and $\phi(a,a), \phi(b,b)$ are both finite. The hardness of VCSP($\{\phi_{\mathsf{xor}}\}$) on $\{0,1\}$ generalises in an obvious way to VCSP($\{\phi\}$) for such functions ϕ (see [48, 83]). Call such a function XOR-like. By Theorem 31, every Γ such that wRelClone (Γ) contains a XOR-like function is intractable. Moreover, the converse is known to be true, that is, for *every* NP-hard finite-valued core language Γ, wRelClone (Γ) contains a XOR-like function [48, 83] (see Theorem 46).

In fact, *most* languages (not necessarily crisp or finite-valued) known to be NP-hard are known to satisfy the condition that wRelClone (Γ) contains a function that is NAE-like or XOR-like. It is an open question whether there exist intractable languages Γ that do not satisfy this condition. Some NP-hard languages, e.g. those from [81], are not known to satisfy it.

We now focus on complexity classifications. For crisp languages (i.e. pure feasibility problems), complexity classifications have been established for languages over two-element sets [77] and three-element sets [7] and for languages containing all unary relations [12, 2]. For finite-valued languages (i.e. pure optimisation

problems), it has been shown that BLP solves *all* tractable cases [83].

Theorem 46 (Classification of Finite-Valued Languages [83])**.** *Let* Γ *be a finite-valued constraint language that is a core. Either* Γ *has a binary symmetric weighted polymorphism (and hence is solvable by BLP), or else* wRelClone(Γ) *contains a XOR-like function, and hence* Γ *is intractable.*

Theorem 46 generalises several previous classification results for finite-valued languages. Tractability in these earlier results was often characterised by (more) specific binary symmetric weighted polymorphisms:

- A core $\{0, 1\}$-valued language[2] over a two-element set [58, 25], or over a three-element set [55], or including all unary $\{0, 1\}$-valued functions [28] is tractable if it is submodular on a chain (cf. Examples 18 and 19), and intractable otherwise.
- A core $\{0, 1\}$-valued language over a four-element set [57] is tractable if it is submodular on some lattice (cf. Example 19) or 1-defect (cf. Example 25) and intractable otherwise.
- A core finite-valued language over a two-element set [17] is tractable if it is submodular (cf. Example 18) and intractable otherwise.
- A core finite-valued language over a three-element set [48] is intractable if it is submodular on a chain (cf. Example 19) or skew bisubmodular (cf. Example 23) and intractable otherwise.
- A finite-valued language containing all $\{0, 1\}$-valued unary cost functions [63] is tractable if it is submodular on a chain (cf. Example 24) and intractable otherwise.

Theorem 46 also implies a classification of the so-called Min-0-Ext problems [45].

For languages where the cost functions can take infinite values, no general complexity classification is known. In fact, even the special case of $\{0, \infty\}$-valued languages is a challenging open problem over sets with four or more elements as it corresponds to the complexity classification of the CSP (cf. Example 10). For the general VCSP, unlike the CSP, there is not even a well-established conjecture.

Nevertheless, some interesting and nontrivial partial results are known. For example, a complete complexity classification for valued constraint languages over a two-element set was established in [17]. Note that on a two-element set there is precisely one majority operation, as defined in Example 26, which we will denote by Mjrty, and precisely one minority operation, as defined in Example 27, which we will denote by Mnrty. There are also precisely two constant operations, which will be denoted Const_0 and Const_1.

[2] $\{0, 1\}$-valued languages correspond to Max-CSPs, cf. Example 12.

Theorem 47 (Classification of Boolean Languages [17])**.** *A valued constraint language* Γ *on* $D = \{0, 1\}$ *is tractable if it admits at least one of the following eight multimorphisms. Otherwise* wRelClone (Γ) *contains* ϕ_{nae} *or* ϕ_{xor} *and* Γ *is intractable.*

1. $\langle \mathrm{Const}_0 \rangle$
2. $\langle \mathrm{Const}_1 \rangle$
3. $\langle \mathrm{Min}, \mathrm{Min} \rangle$,
4. $\langle \mathrm{Max}, \mathrm{Max} \rangle$,
5. $\langle \mathrm{Min}, \mathrm{Max} \rangle$,
6. $\langle \mathrm{Mjrty}, \mathrm{Mjrty}, \mathrm{Mjrty} \rangle$,
7. $\langle \mathrm{Mnrty}, \mathrm{Mnrty}, \mathrm{Mnrty} \rangle$,
8. $\langle \mathrm{Mjrty}, \mathrm{Mjrty}, \mathrm{Mnrty} \rangle$.

Let us compare Theorem 47 with a classification of crisp Boolean languages, originally established by Schaefer in [77] and restated here using polymorphisms (see, e.g. [14]): A crisp constraint language on $D = \{0, 1\}$ is tractable if it admits one of the following six polymorphisms: Const_0, Const_1, Min, Max, Mjrty, Mnrty; otherwise it is intractable. These six tractable cases are covered by cases (1-4), (6), and (7) in Theorem 47. The six cases correspond to sets of Boolean relations that are 0-valid, or 1-valid, or expressible by Horn clauses, dual Horn clauses, 2-clauses, or linear equations over the field with 2 elements, respectively.

The hardness part of Theorem 47 can be rederived using the algebraic theory described in Section 4; see [24, 19] for details. We remark that if we restrict to core Boolean valued constraint languages, the first two cases in Theorem 47 disappear as those languages are not cores (and in fact are solvable trivially).

Another general complexity classification result concerns languages that contain all $\{0, 1\}$-valued unary cost functions. Note that a weighted polymorphism ω is called *conservative* if $f(x_1, \ldots, x_k) \in \{x_1, \ldots, x_k\}$ for all $f \in \mathrm{supp}(\omega)$.

Theorem 48 (Classification of Conservative Languages [63])**.** *Let* Γ *be a valued constraint language on a set* D *such that* Γ *contains all* $\{0, 1\}$*-valued unary cost functions on* D*. Then either* Γ *admits a conservative binary multimorphism* $\langle f_1, f_2 \rangle$ *and a conservative ternary multimorphism* $\left\langle f'_1, f'_2, f'_3 \right\rangle$ *and there is a family* M *of 2-element subsets of* D*, such that:*

- *for every* $\{a, b\} \in M$, $\langle f_1, f_2 \rangle$ *restricted to* $\{a, b\}$ *is a symmetric tournament pair (see Example 24), and*
- *for every* $\{a, b\} \notin M$, $\left\langle f'_1, f'_2, f'_3 \right\rangle$ *restricted to* $\{a, b\}$ *is an MJN multimorphism (see Example 28),*

in which case Γ *is tractable, or else* Γ *is intractable.*

The algorithm for solving the tractable case identified in Theorem 48 first enforces local consistency (see the discussion of bounded width at the beginning of this section). After this preprocessing step, any instance admits a symmetric tournament pair multimorphism [63] and is thus solvable using BLP.

We now briefly describe the partial classification results so far obtained for the Min-Cost-Hom and Min-Sol problems discussed in Examples 13 and 15 respectively. Recall that a Min-Cost-Hom problem corresponds to VCSP(Γ) for some language Γ containing only crisp cost functions and unary cost functions. Min-Sol problems are Min-Cost-Hom problems where the only unary cost function in Γ is a specific injective and finite-valued cost function.

The complexity classification for Min-Cost-Hom for languages containing all unary cost functions was established in [81]. The tractable case can be reduced, after a preprocessing step using local consistency techniques, to a certain problem on perfect graphs known to be solvable in polynomial time using linear programming [38]. For the special case of digraphs (i.e., when the only non-unary cost function allowed is a single binary crisp cost function), a complexity classification was obtained in [43].

The classification of Min-Cost-Hom for languages containing all unary crisp cost functions was initially studied in [80] and fully established in [85].

Finally, using the techniques from Section 4 and from [83], a very recent result has established the computational complexity of Min-Cost-Hom for all languages over a three-element set [86]. The only tractable cases either admit a weighted polymorphism with a semilattice operation in its positive support or a certain type of tournament pair. The former case is tractable using BLP by Theorem 44 and the latter case is tractable using a reduction to the result in [81] discussed above.

The classification of Min-Sol problems was established in [56] for maximal languages over a four-element set and for homogenenous languages. The classification of Min-Sol has recently also been obtained for all languages over a three-element set [85]. Using the notion of cores and the algebraic techniques from Section 4 and from [82, 83], three tractable cases have been identified: bisubmodular languages (Example 22), generalised min-closed languages (generalising Example 21), and generalised weak-tournament pair languages (generalising Example 24); the first two are solvable using BLP, by Theorem 44, while the last is solvable by a method similar to the tractable case from [81] discussed above.

Adapting the main result of [13] on CSPs, Powell and Krokhin have recently shown [74] that for every problem VCSP(Γ), where Γ is finite, there is a polynomial-time equivalent Min-Cost-Hom problem, VCSP(Γ'), where Γ' contains only a single crisp binary function and a single finite-valued unary function. Moreover, the equivalence also preserves (in both directions) many useful weighted polymorphisms of Γ, such as symmetric and weak near-unanimity polymorphisms [4]. Thus, in order to classify the computational complexity of *any* valued constraint

language it suffices to classify Min-Cost-Hom problems of this restricted form. This mirrors a similar reduction from the general CSP to the binary case which was first established in [29].

6 Approximation

Since many forms of valued constraint satisfaction problem are NP-hard, it is natural to study approximation algorithms for these problems, and their limits. Recall that a polynomial-time algorithm for an optimisation problem Π is called an r-approximation algorithm if, for each instance I of Π, the algorithm returns a solution S for I whose measure $m(S)$ satisfies the inequality

$$\max\left(\frac{m(S)}{OPT(S)}, \frac{OPT(S)}{m(S)}\right) \leq r.$$

The bound r is called the approximation ratio of the algorithm. Note that in general r can be a function of the size of I.

There has been major progress in the last 20 years in designing approximation algorithms and understanding the (in)approximability of combinatorial optimisation problems. The former direction was boosted by the application of techniques based on semidefinite programming (SDP) [34] whilst the latter was powered to a large extent by the theory of probabilistically checkable proofs, or PCPs, see [1]. A notable early source of inapproximability results is [40], where it is shown that certain problems (such as Max-3-Sat) can be approximated within a (problem-specific) constant r, but, unless P=NP, not within $r-\epsilon$ for any $\epsilon > 0$. There is now a large body of such optimal inapproximability results, including those for Minimum Vertex Cover and Max Cut, whose validity depends on the *Unique Games Conjecture*, or UGC (see survey [59]). This conjecture states that, for any $\epsilon > 0$, there is a large enough integer $k = k(\epsilon)$ such that it is NP-hard to distinguish two types of systems of linear equations of the form $x_i + x_j \equiv a_{ij} \pmod{k}$: those where at least a $(1-\epsilon)$-fraction of the equations can be satisfied and those where any assignment satisfies at most an ϵ-fraction of the equations. Despite the fact that the UGC has been used as a basis for many results, it is still open and the approximation community seems to be evenly divided as to which way it will eventually be resolved.

Semidefinite programming is an extension of linear programming where the variables are vectors in a high-dimensional space and the constraints, as well as the objective function, are linear in the inner products of these vectors. Any VCSP instance has a basic semidefinite programming relaxation similar to the BLP relaxation defined in Section 5. A breakthrough result of Raghavendra [75, 76] shows how to use the basic SDP relaxation to design, for *any* given finite and

finite-valued language Γ, an approximation algorithm for VCSP(Γ) that achieves some constant approximation ratio; moreover, this ratio cannot be improved unless the UGC is false. This ratio is not explicit, but there is an algorithm that can compute it with any given accuracy in doubly exponential time. It is interesting that this (conditionally) optimal ratio is related to a parameter of some objects similar to weighted polymorphisms. For more details, consult Raghavendra's paper and thesis [75, 76]; note that the (finite-valued) VCSP is referred to there as the generalized CSP or GCSP.

The class of all optimisation problems having a (polynomial-time) constant-factor approximation algorithm is denoted by APX. From the approximation point of view, the best type of algorithm is a PTAS (polynomial-time approximation scheme) which is actually a series of algorithms A_ϵ, $\epsilon > 0$, such that A_ϵ gives a $(1 + \epsilon)$-approximation and runs in time that is polynomial in the size of the instance (but not necessarily in $1/\epsilon$). One way to rule out the existence of a PTAS for a specific optimisation problem Π (unless P=NP) is to show that this problem is APX-hard, i.e., that every problem in APX has an approximation-preserving reduction to Π.

The classification results from Section 5 distinguish between (exact) polynomial solvability and NP-hardness. Some of these results can be strengthened to become dichotomies between polynomial solvability and APX-hardness For example, as discussed in Example 12, Max-CSP is equivalent to VCSP(Γ_{Max}) where Γ_{Max} consists of all cost functions taking only the values 0 and 1. For approximation results it is convenient to replace these with values with -1 and 0 respectively. Then the intractable cases of VCSP(Γ) with $\Gamma \subseteq \Gamma_{\mathsf{Max}}$ can be shown to be APX-hard (in fact, APX-complete, as each Max-CSP problem with a finite language belongs to APX) when Γ contains all unary $\{-1, 0\}$-valued functions [28] and when $|D| = 3$ [55].

There are only a few results concerning the approximability of VCSP(Γ) for languages Γ containing cost functions that can take infinite values. For example, it is shown in [44] that the problem VCSP($\{\phi_H\} \cup \Gamma^+_{\mathsf{unary}}$), a special case of Min-Cost-Hom (see Example 13) where $H = (V, E)$ is an undirected graph without loops and $\Gamma^+_{\mathsf{unary}}$ contains all unary functions with non-negative values, is not approximable within any factor if the List H-Colouring problem (cf. Example 11) is NP-complete and it has a $|V|$-approximation algorithm otherwise. As another example, the APX-hardness of some Min-Sol problems (Example 15) is established in [53].

7 The oracle model

In this paper we have assumed that the objective function in our problem is represented as a sum of functions each defined on some subset of the variables. There is a rich tradition in combinatorial optimisation of studying problems where the objective function is represented instead by a value-giving *oracle*. In this model a problem is tractable if it can be solved in polynomial time using only polynomially many queries to the oracle (where the polynomial is in the number of variables). Note that any query to the oracle can be simulated in linear time in the VCSP model. Hence, a tractability result (for a class of functions) in the oracle model automatically transfers to the VCSP model, while hardness results automatically transfer in the opposite direction.

One class of functions that has received particular attention in the oracle model is the class of submodular functions (cf. Example 18). There are several known algorithms for minimising a (finite-valued) submodular function using only a polynomial number of calls to a value-giving oracle (see [50, 51, 78]).

However, for some submodular valued constraint languages Γ, VCSP(Γ) can be solved much more efficiently than by using these general approaches. For example, the language Γ_{cut} defined in Example 8 can be solved in cubic time using the Min-Cut-based algorithm described in Example 8. A similar efficient approach can be used for all languages that are expressible over Γ_{cut}. However, it was shown in [88, 90] that not all submodular functions are expressible over Γ_{cut}, so this approach cannot be directly extended to solve arbitrary submodular VCSP instances. It is currently an open question whether the minimisation problem for submodular functions defined by sums of bounded arity submodular functions in the VCSP model is easier than general submodular function minimisation in the oracle model.

Other classes of finite-valued functions that can be efficiently minimised in the oracle model include bisubmodular and α-bisubmodular functions (Examples 22 and 23) [31, 71, 32, 47], functions with a 1-defect multimorphism (Example 25) [57], and functions that are submodular on certain lattices (Example 19) [64, 65]. The complexity of submodular function minimisation in the oracle model over arbitrary non-distributive lattices is still unknown (in the VCSP model, all such language are tractable, by Theorem 44).

The following general problem was mentioned in [48, 57, 82]: which weighted polymorphisms ω are sufficient to guarantee an efficient minimization algorithm, in the value-oracle model, for valued constraint languages Γ with $\omega \in \mathrm{wPol}(\Gamma)$? Natural candidates for which the question is open include the k-submodularity multimorphism for $k \geq 3$ from Example 22 and submodularity multimorphisms on many lattices from Example 19.

8 Conclusions and future directions

We have shown that the valued constraint satisfaction problem is a powerful general framework that can be used to express many standard combinatorial optimisation problems. The general problem is NP-hard, but there are many special cases that have been shown to be tractable. In particular, by considering restrictions on the cost functions we allow in problem instances, we have identified a range of different sets of cost functions that ensure tractability.

These restricted sets of cost functions are referred to as valued constraint languages, and we have described in Section 4 the very general algebraic techniques now being developed to classify the complexity of these languages.

This classification is still far from complete. In fact, even in the special case of the CSP (Example 10), where all cost functions take only the values 0 or ∞, there is still no complete classification of complexity for the corresponding constraint languages. This problem has been studied for many years, beginning with the seminal work of Feder and Vardi who conjectured that any such language will be either tractable or NP-complete [29]. This conjecture is still unresolved. However, the Algebraic Dichotomy conjecture [11] specifies the boundary between tractable and intractable languages, and it has been proved in many important cases. Naturally, it is desirable to develop the algebraic theory of VCSPs to the point where one could make a credible algebraic dichotomy conjecture for the VCSP, in order to have a specific target to aim at.

For finite-valued languages, the complexity classification is complete, see Theorem 46. One could ask, however, whether the tractability condition can be made tighter by being more specific about which binary symmetric weighted polymorphisms need to be present there. For $|D| = 2, 3$, tight descriptions are given in [17, 48].

The algebraic theory of the VCSP presented in Section 4 is based on the new notion of a weighted clone. Very little is known about weighted clones, and this direction is wide open for purely algebraic investigation. Some specific open problems include the (possible) description of weighted clones for $D = \{0, 1\}$, the identification of minimal weighted clones, and the investigation of classes of weighted clones supported by a given ordinary clone.

Further developing the algebraic theory of the VCSP using algebras and varieties [73] is a very promising direction of research because this theory works with a more general notion of expressibility. Possible algebraic dichotomy results from this theory would state that either a language expresses, in this more general way, a given function (usually with undesirable algorithmic properties of the corresponding VCSP) or else it has a "nice" weighted polymorphism. Such results [11, 67] have been fundamental to the success of the algebraic approach to complexity for the CSP.

It is natural to investigate how the operations that play a role in the algebraic theory for the CSP can be adapted to the VCSP setting. Examples of such conditions that we discussed earlier are weak near-unanimity and edge operations; there are several others. What can be said about valued constraint languages with weighted polymorphisms whose positive support includes such operations?

As we discussed in Section 5, only three algorithmic techniques seem to be sufficient to solve tractable crisp and finite-valued VCSPs (Bounded Width, Few Subpowers, and Basic LP relaxation). There also seem to be essentially only two seeds of hardness that cause intractability (NAE-like and XOR-like functions). Are there tractable general-valued VCSPs that require different techniques? Are there intractable general-valued VCSPs that can express neither NAE-like nor XOR-like functions?

The notion of weighted polymorphism works well for studying the exact solvability of the VCSP. It would be natural to explore its applicability to approximability questions for the VCSP and to oracle-tractability for classes of functions, as we discussed in Sections 6 and 7.

In this survey we have focused on the complexity of valued constraint satisfaction problems with restricted constraint languages. It is also possible to ensure tractability by restricting the structure of the constraint scopes - so-called *structural* restrictions [36, 37, 70]. Combining structural restrictions with language restrictions leads to so-called *hybrid* restrictions, and these provide a promising source of new tractable cases [21, 22] which has so far been very little explored.

References

[1] S. Arora and B. Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009.

[2] L. Barto. The dichotomy for conservative constraint satisfaction problems revisited. In *LICS'11*, pages 301–310. IEEE Computer Society, 2011.

[3] L. Barto and M. Kozik. Absorbing subalgebras, cyclic terms and the constraint satisfaction problem. *Logical Methods in Computer Science*, 8, 2012.

[4] L. Barto and M. Kozik. Constraint Satisfaction Problems Solvable by Local Consistency Methods. *Journal of the ACM*, 61(1), 2014. Article No. 3.

[5] F. Börner. Basics of Galois connections. In *Complexity of Constraints*, volume 5250 of *LNCS*, pages 38–67. Springer, 2008.

[6] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *Computer Vision and Pattern Recognition*, pages 648–655. IEEE Computer Society, 1998.

[7] A. Bulatov. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *Journal of the ACM*, 53(1):66–120, 2006.

[8] A. Bulatov. Bounded relational width. Manuscript, 2009.

[9] A. Bulatov and V. Dalmau. A simple algorithm for Mal'tsev constraints. *SIAM Journal on Computing*, 36(1):16–27, 2006.

[10] A. Bulatov and P. Jeavons. Algebraic structures in combinatorial problems. Technical Report MATH-AL-4-2001, Technische Universität Dresden, 2001.

[11] A. Bulatov, A. Krokhin, and P. Jeavons. Classifying the Complexity of Constraints using Finite Algebras. *SIAM Journal on Computing*, 34(3):720–742, 2005.

[12] A. A. Bulatov. Complexity of conservative constraint satisfaction problems. *ACM Transactions on Computational Logic*, 12(4), 2011. Article 24.

[13] J. Bulín, D. Delic, M. Jackson, and T. Niven. On the Reduction of the CSP Dichotomy Conjecture to Digraphs. In *CP'13*, volume 8124 of *LNCS*, pages 184–199. Springer, 2013.

[14] D. Cohen and P. Jeavons. The complexity of constraint languages. In F. Rossi, P. van Beek, and T. Walsh, editors, *The Handbook of Constraint Programming*. Elsevier, 2006.

[15] D. Cohen, M. Cooper, P. Jeavons, and A. Krokhin. Supermodular Functions and the Complexity of MAX-CSP. *Discrete Applied Mathematics*, 149(1-3):53–72, 2005.

[16] D. A. Cohen, M. C. Cooper, and P. G. Jeavons. An Algebraic Characterisation of Complexity for Valued Constraints. In *CP'06*, volume 4204 of *LNCS*, pages 107–121. Springer, 2006a.

[17] D. A. Cohen, M. C. Cooper, P. G. Jeavons, and A. A. Krokhin. The Complexity of Soft Constraint Satisfaction. *Artificial Intelligence*, 170(11):983–1016, 2006b.

[18] D. A. Cohen, M. C. Cooper, and P. G. Jeavons. Generalising submodularity and Horn clauses: Tractable optimization problems defined by tournament pair multimorphisms. *Theoretical Computer Science*, 401(1-3):36–51, 2008.

[19] D. A. Cohen, M. C. Cooper, P. Creed, P. Jeavons, and S. Živný. An algebraic theory of complexity for discrete optimisation. *SIAM Journal on Computing*, 42(5):915–1939, 2013.

[20] M. C. Cooper, S. de Givry, M. Sánchez, T. Schiex, M. Zytnicki, and T. Werner. Soft arc consistency revisited. *Artificial Intelligence*, 174(7–8):449–478, 2010.

[21] M. C. Cooper and S. Živný. Hybrid tractability of valued constraint problems. *Artificial Intelligence*, 175(9-10):1555–1569, 2011.

[22] M. C. Cooper and S. Živný. Tractable triangles and cross-free convexity in discrete optimisation. *Journal of Artificial Intelligence Research*, 44:455–490, 2012.

[23] Y. Crama and P. L. Hammer. *Boolean Functions - Theory, Algorithms, and Applications*. Cambridge University Press, 2011.

[24] P. Creed and S. Živný. On minimal weighted clones. In *CP'11*, volume 6876 of *LNCS*, pages 210–224. Springer, 2011.

[25] N. Creignou, S. Khanna, and M. Sudan. *Complexity Classification of Boolean Constraint Satisfaction Problems*, volume 7 of *SIAM Monographs on Discrete Mathematics and Applications*. SIAM, 2001.

[26] N. Creignou, P. G. Kolaitis, and H. Vollmer, editors. *Complexity of Constraints: An Overview of Current Research Themes*, volume 5250 of *LNCS*, 2008. Springer.

[27] E. Dahlhaus, D. Johnson, C. Papadimitriou, P. Seymour, and M. Yannakakis. The Complexity of Multiterminal Cuts. *SIAM Journal on Computing*, 23(4):864–894, 1994.

[28] V. Deineko, P. Jonsson, M. Klasson, and A. Krokhin. The approximability of Max CSP with fixed-value constraints. *Journal of the ACM*, 55(4), 2008. Article 16.

[29] T. Feder and M. Y. Vardi. The Computational Structure of Monotone Monadic SNP and Constraint Satisfaction: A Study through Datalog and Group Theory. *SIAM Journal on Computing*, 28(1):57–104, 1998.

[30] S. Fujishige. *Submodular Functions and Optimization*, volume 58 of *Annals of Discrete Mathematics*. North-Holland, Amsterdam, 2nd edition, 2005.

[31] S. Fujishige and S. Iwata. Bisubmodular Function Minimization. *SIAM Journal on Discrete Mathematics*, 19(4):1065–1073, 2005.

[32] S. Fujishige, S. Tanigawa, and Y. Yoshida. Generalized skew bisubmodularity: A characterization and a min-max theorem. Technical Report RIMS-1781, Kyoto University, 2013.

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

[34] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995.

[35] A. V. Goldberg and R. E. Tarjan. A New Approach to the Maximum Flow Problem. *Journal of the ACM*, 35(4):921–940, 1988.

[36] G. Gottlob, G. Greco, and F. Scarcello. Tractable Optimization Problems through Hypergraph-Based Structural Restrictions. In *ICALP'09*, volume 5556 of *LNCS*, pages 16–30. Springer, 2009.

[37] M. Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *Journal of the ACM*, 54(1):1–24, 2007.

[38] M. Grötschel, L. Lovasz, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988.

[39] G. Gutin, P. Hell, A. Rafiey, and A. Yeo. A dichotomy for minimum cost graph homomorphisms. *European Journal of Combinatorics*, 29(4):900–911, 2008.

[40] J. Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4): 798–859, 2001.

[41] P. Hell and J. Nešetřil. *Graphs and Homomorphisms*. Oxford University Press, 2004.

[42] P. Hell and J. Nešetřil. Colouring, constraint satisfaction, and complexity. *Computer Science Review*, 2(3):143–163, 2008.

[43] P. Hell and A. Rafiey. The Dichotomy of Minimum Cost Homomorphism Problems for Digraphs. *SIAM Journal on Discrete Mathematics*, 26(4):1597–1608, 2012.

[44] P. Hell, M. Mastrolilli, M. M. Nevisi, and A. Rafiey. Approximation of Minimum Cost Homomorphisms. In *Proceddings of the 20th Annual European Symposium on Algorithms (ESA12)*, pages 587–598, 2012.

[45] H. Hirai. Discrete Convexity and Polynomial Solvability in Minimum 0-Extension Problems. In *SODA'13*, pages 1770–1778. SIAM, 2013.

[46] A. Huber and V. Kolmogorov. Towards minimizing k-submodular functions. Technical Report arXiv:1309.5469, 2013.

[47] A. Huber and A. Krokhin. Oracle tractability of skew bisubmodular functions. Technical Report arXiv:1308.6505, 2013.

[48] A. Huber, A. Krokhin, and R. Powell. Skew bisubmodularity and valued CSPs. *SIAM Journal on Computing*, 2014. To appear.

[49] P. M. Idziak, P. Markovic, R. McKenzie, M. Valeriote, and R. Willard. Tractability and learnability arising from algebras with few subpowers. *SIAM Journal on Computing*, 39(7):3023–3037, 2010.

[50] S. Iwata. Submodular Function Minimization. *Mathematical Programming*, 112(1): 45–64, 2008.

[51] S. Iwata and J. B. Orlin. A Simple Combinatorial Algorithm for Submodular Function Minimization. In *SODA'09*, pages 1230–1237, 2009.

[52] P. G. Jeavons, D. A. Cohen, and M. Gyssens. Closure Properties of Constraints. *Journal of the ACM*, 44(4):527–548, 1997.

[53] P. Jonsson and G. Nordh. Introduction to the MAXIMUM SOLUTION Problem. In *Complexity of Constraints*, volume 5250 of *LNCS*, pages 255–282. Springer, 2008.

[54] P. Jonsson and J. Thapper. Approximability of the maximum solution problem for certain families of algebras. In *CSR'09*, volume 5675 of *LNCS*, pages 215–226. Springer, 2009.

[55] P. Jonsson, M. Klasson, and A. Krokhin. The Approximability of Three-valued MAX CSP. *SIAM Journal on Computing*, 35(6):1329–1349, 2006.

[56] P. Jonsson, F. Kuivinen, and G. Nordh. MAX ONES Generalized to Larger Domains. *SIAM Journal on Computing*, 38(1):329–365, 2008.

[57] P. Jonsson, F. Kuivinen, and J. Thapper. Min CSP on Four Elements: Moving Beyond Submodularity. In *CP'11*, volume 6876 of *LNCS*, pages 438–453. Springer, 2011.

[58] S. Khanna, M. Sudan, L. Trevisan, and D. Williamson. The approximability of constraint satisfaction problems. *SIAM Journal on Computing*, 30(6):1863–1920, 2001.

[59] S. Khot. On the Unique Games Conjecture (Invited Survey). In *CCC'10*, pages 99–121. IEEE Computer Society, 2010.

[60] V. Kolmogorov. Submodularity on a tree: Unifying $L^\natural$-convex and bisubmodular functions. In *MFCS'11*, volume 6907 of *LNCS*, pages 400–411. Springer, 2011.

[61] V. Kolmogorov. The power of linear programming for finite-valued CSPs: a constructive characterization. In *ICALP'13*, volume 7965 of *LNCS*, pages 625–636. Springer, 2013.

[62] V. Kolmogorov, J. Thapper, and S. Živný. The power of linear programming for general-valued CSPs. 2013. arXiv:1311.4219.

[63] V. Kolmogorov and S. Živný. The complexity of conservative valued CSPs. *Journal of the ACM*, 60(2), 2013. Article No. 10.

[64] A. Krokhin and B. Larose. Maximizing Supermodular Functions on Product Lattices, with Application to Maximum Constraint Satisfaction. *SIAM Journal on Discrete Mathematics*, 22(1):312–328, 2008.

[65] F. Kuivinen. On the complexity of submodular function minimisation on diamonds. *Discrete Optimization*, 8(3):459–477, 2011.

[66] R. Ladner. On the Structure of Polynomial Time Reducibility. *Journal of the ACM*, 22:155–171, 1975.

[67] B. Larose and P. Tesson. Universal algebra and hardness results for constraint satisfaction problems. *Theoretical Computer Science*, 410(18):1629–1647, 2009.

[68] S. L. Lauritzen. *Graphical Models*. Oxford University Press, 1996.

[69] M. Maróti and R. McKenzie. Existence theorems for weakly symmetric operations. *Algebra Universalis*, 59(3-4):463–489, 2008.

[70] D. Marx. Tractable hypergraph properties for constraint satisfaction and conjunctive queries. *Journal of the ACM*, 60(6), 2013. Article No. 42.

[71] S. T. McCormick and S. Fujishige. Strongly polynomial and fully combinatorial algorithms for bisubmodular function minimization. *Mathematical Programming*, 122(1):87–120, 2010.

[72] M. Mezard and A. Montanari. *Information, Physics, and Computation*. Oxford University Press, 2009.

[73] J. Ochremiak. Algebraic properties of valued constraint satisfaction problem. Technical report, 2014. arXiv:1403.0476.

[74] R. Powell and A. Krokhin. A reduction of VCSP to digraphs. Manuscript, 2014.

[75] P. Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *STOC'08*, pages 245–254. ACM, 2008.

[76] P. Raghavendra. *Approximating NP-hard problems: Efficient algorithms and their limits*. PhD thesis, University of Washington, 2009.

[77] T. J. Schaefer. The Complexity of Satisfiability Problems. In *STOC'78*, pages 216–226. ACM, 1978.

[78] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume 24 of *Algorithms and Combinatorics*. Springer, 2003.

[79] A. Szendrei. *Clones in Universal Algebra*, volume 99 of *Seminaires de Mathematiques Superieures*. University of Montreal, 1986.

[80] R. Takhanov. Extensions of the Minimum Cost Homomorphism Problem. In *COCOON'10*, volume 6196 of *LNCS*, pages 328–337. Springer, 2010a.

[81] R. Takhanov. A Dichotomy Theorem for the General Minimum Cost Homomorphism Problem. In *STACS'10*, pages 657–668, 2010b.

[82] J. Thapper and S. Živný. The power of linear programming for valued CSPs. In *FOCS'12*, pages 669–678. IEEE, 2012.

[83] J. Thapper and S. Živný. The complexity of finite-valued CSPs. In *STOC'13*, pages 695–704. ACM, 2013.

[84] D. Topkis. *Supermodularity and Complementarity*. Princeton University Press, 1998.

[85] H. Uppman. The Complexity of Three-Element Min-Sol and Conservative Min-Cost-Hom. In *ICALP'13*, volume 7965 of *LNCS*, pages 804–815. Springer, 2013.

[86] H. Uppman. Computational Complexity of the Extended Minimum Cost Homomorphism Problem on Three-Element Domains. In *STACS'14*, volume 25, pages 651–662, 2014.

[87] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.

[88] S. Živný. *The Complexity and Expressive Power of Valued Constraints*. PhD thesis, University of Oxford, 2009.

[89] S. Živný. *The complexity of valued constraint satisfaction problems*. Cognitive Technologies. Springer, 2012. ISBN 978-3-642-33973-8.

[90] S. Živný, D. A. Cohen, and P. G. Jeavons. The Expressive Power of Binary Submodular Functions. *Discrete Applied Mathematics*, 157(15):3347–3358, 2009.

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

A fundamental technique in the design of parameterized algorithms is *kernelization*: Given a problem instance I with parameter k, the basic idea is to try and preprocess the instance I of length n by applying efficient "reduction rules" in order to simplify it and reduce it to a kernel instance of the same problem that is of size a polynomial in k. A brute-force/exponential-time algorithm can then be used to solve the kernel instance. Smaller kernels often lead to faster algorithms. How small, as a function of k, can kernels be made? There is a nice hardness theory, based on the complexity theoretic assumption coNP $\not\subseteq$ NP/poly, which can be used to prove lower bounds for kernel size.

Kernelization is a flourishing area of parameterized complexity with many recent results (both upper and lower bounds). Stefan Kratsch shares with us some of the latest developments in the field. His very readable survey article, with illustrative examples, invites the non-expert to this exciting area of complexity theory.

BEATCS no 113

RECENT DEVELOPMENTS IN KERNELIZATION: A SURVEY

Stefan Kratsch*
Technical University Berlin, Germany
stefan.kratsch@tu-berlin.de

Abstract

Kernelization is a formalization of efficient preprocessing, aimed mainly at combinatorially hard problems. Empirically, preprocessing is highly successful in practice, e.g., in state-of-the-art SAT and ILP solvers. The notion of kernelization from parameterized complexity makes it possible to rigorously prove upper and lower bounds on, e.g., the maximum output size of a preprocessing in terms of one or more problem-specific parameters. This avoids the often-raised issue that we should not expect an efficient algorithm that provably shrinks every instance of any NP-hard problem.

In this survey, we give a general introduction to the area of kernelization and then discuss some recent developments. After the introductory material we attempt a reasonably self-contained update and introduction on the following topics: (1) Lower bounds for kernelization, taking into account the recent progress on the AND-conjecture. (2) The use of matroids and representative sets for kernelization. (3) Turing kernelization, i.e., understanding preprocessing that adaptively or non-adaptively creates a large number of small outputs.

1 Introduction

Kernelization is a theoretical formalization of efficient preprocessing for (NP-) hard problems. By efficient preprocessing we mean any polynomial-time algorithm that given a problem instance outputs an equivalent instance that is, if possible, simpler than the initial one. Mainly, we are interested in data reduction where the obtained instance is as small as possible (but we will avoid the term data reduction for its name clash with reductions). Empirically, preprocessing is

*Supported by the Emmy Noether-program of the German Research Foundation (DFG), KR 4286/1.

very successful in practice, e.g., within the well-known ILP solver CPLEX, which motivates a mathematically rigorous study.

Before giving formal definitions and further background, let us begin with a simple and well-known example. Consider the Vertex Cover problem where we are given as input a graph $G = (V, E)$ and a value $k \in \mathbb{N}$ and we need to determine whether there exists a set S of at most k vertices such that every edge is incident with at least one vertex in S. Due to the NP-hardness of the problem we do not expect that *every instance* can be efficiently reduced in size. Indeed, any polynomial-time algorithm that guarantees a size reduction of at least one bit for *all instances* of Vertex Cover could be iterated to also solve Vertex Cover in polynomial time, implying P = NP. Despite this obstacle to efficient preprocessing there are simple reduction rules that can be seen to yield a provable size bound; how does that fit together?

Rule 1. Delete any isolated vertex v of G, i.e., return $(G - v, k)$. *Correctness: We never need v in any solution since it covers no edges.*

Rule 2. If a vertex v has degree greater than k in G then we (are forced to) select the vertex for the solution, which is expressed by returning $(G - v, k - 1)$. *Correctness: Not selecting v would require selecting the neighborhood $N(v)$ of v which is of size greater than our budget k.*

Rule 3. If Rule 2 does not apply and the graph G has more than k^2 edges then answer no. *Correctness: Covering more than k^2 edges with at most k vertices would require at least one vertex of degree greater than k.*

It is not hard to see that all three rules can be applied in polynomial time and that when no rule is applicable we have an equivalent instance with a graph that has at most k^2 edges and $2k^2$ vertices; this instance can be encoded in $O(k^2 \log k)$ bits. (By more sophisticated arguments this can be improved to at most $2k$ vertices and $O(k^2)$ total size [15].)

We see that by relating the output guarantee of our preprocessing to the value k, we avoided the issue of not being able to shrink every instance. Intuitively, the solution size k in a vertex cover instance is a good measure of its complexity, since it is not hard to find, e.g., a $O(2^k nm)$ time branching algorithm for it; if k is constant or at least $k \in O(\log n)$ then this runtime is even polynomial in the input size. Similarly, our simple preprocessing has showed us that a comparatively small value of k implies that the size of our instance can be reduced. If, otherwise, k is large (compared to n) then the bound of $n \leq 2k^2$ does not guarantee any simplification, which is consistent with the observed obstacle to general efficient size reductions.

Generally, the field of *parameterized complexity* studies the influence of so-called *parameters*, like k for Vertex Cover, on problem complexity. We will

adopt the naming convention of including the parameter choice into the problem name, e.g., Vertex Cover(k) stands for Vertex Cover with parameter k and Vertex Cover(Δ) stands for parameterization by maximum degree. A kernelization for a parameterized problem can then be simply formalized as any efficient algorithm that gives an equivalent instance of size (and parameter value) bounded by a function in the input parameter (see Section 3 for formal definitions). It should come as no surprise that the achievable output guarantees depend greatly on the choice of parameter.

2 A brief history and overview of kernelization

The use of reduction rules to simplify problems is often traced back to the work of Quine [66] from 1952 on simplifying truth functions, e.g., by unit-clause propagation and elimination of pure literals. It was recognized early that efficient reduction rules are not only empirically useful but could also be used to improve theoretical performance guarantees of exhaustive search algorithms by ensuring structural restrictions (like degree-bounds); see, e.g., [68]. The study of provable performance guarantees for preprocessing by reduction rules (or any other means) regarding the achievable output size, rather than achievable structure, took much longer to develop.

Kernelization originated as one of many techniques in the toolbox of parameterized complexity (see [24, 25]) and is a successful theoretical formalization of efficient preprocessing with provable performance guarantees. In its early stages kernelization was mostly about coming up with clever reduction rules and combining them with combinatorial arguments to prove that exhaustively reduced instances (to which no more rule could be applied) have size bounded by some function in the initial parameter value. A 2007 survey of Guo and Niedermeier [40] nowadays provides a nice overview on these "early days of kernelization"[1] and in particular asked to develop techniques for kernelization lower bounds. Two other influential works from that time are the linear kernel for Planar Dominating Set by Alber et al. [3] and a programmatic paper of Estivill-Castro et al. [29] that amongst others was perhaps the first to explicitly ask for Turing kernelizations.

The field of kernelization matured, in a sense, when in 2008 Bodlaender et al. [9] came up with a framework for ruling out polynomial sized kernels for many parameterized problems, and, shortly afterwards, this was followed by the first paper on meta kernelization by Bodlaender et al. [10] that gave general kernelization results for a wealth of problems on planar and bounded genus graphs (see also the 2009 survey of Bodlaender [7]). Since then, the field of kernelization

[1] The field of kernelization is still in its twenties.

has been growing rapidly and many new techniques for upper and lower bounds were invented in short succession, apart, of course, from a wealth of results for concrete problems. The survey of Lokshtanov et al. [58] on the occasion of Mike Fellows' 60th birthday in 2012 (see also [8]) gives an excellent account of these developments.

In the present survey we want to focus mainly on recent developments that have taken place since 2012, but also provide a fair introduction for readers new to the field. To this end, the core part of the survey singles out three topics and attempts a (as far as possible) self-contained and detailed presentation. Concretely, we will discuss the use of matroids and representative sets for kernelization (based on [56, 57]), and review the current knowledge about Turing kernelization (motivated by recent progress [69, 49]). Furthermore, since the lower bound framework initiated by Bodlaender et al. [9] holds a central place in kernelization, we explain one complete set of tools for proving such lower bounds. This is, of course, also motivated by the breakthrough work of Drucker [26] that (among other results) settled the so-called AND-distillation conjecture.[2] But, first things first, let us begin by giving an overview of all the interesting things that could not be fitted into this survey for the sake of length and focus.[3]

Overview. The "bread and butter", so to speak, in the kernelization business lies in studying a given parameterized problem, deriving efficient reduction rules for it, and analyzing the obtained rules, that is, analyzing the structure and size of reduced instances. Unfortunately, such rules are of course problem dependent and there does not appear to be *the single general recipe* for them. That said, two frequently used approaches are the following: (1) Begin with an approximation of the desired object or a dual structure. If this is sufficiently large then the instance is trivially YES or trivially NO. If not then there must be large parts that do not contribute to the solution (or do not incur any cost), or that are obstructed by a small set of objects/vertices/etc. Often, a careful analysis can devise "high-degree rules" (as for the simple example of VERTEX COVER(k)) that resolve or simplify these cases. (2) Another frequently used tool is the Sunflower Lemma of Erdős and Rado [28], particularly for covering or packing objects or sets of bounded size. Effectively, the Sunflower Lemma states that a sufficiently large family of bounded size objects either involves a large packing (giving trivial YES for packing and trivial NO for covering) or it contains a so-called sunflower formed by objects that are pairwise obstructing in the same way; often, we can safely delete on of these obstructing objects (and repeat).

[2] Very recently, Dell [20] announced a simpler proof for the AND-distillation conjecture.

[3] Conveniently, and not entirely by chance, these topics are covered in detail by Lokshtanov et al. [58].

To get a more detailed understanding of reduction rule based kernelization results it is probably best to read some of them in detail; see, e.g., [50, 11, 52].

Above-guarantee parameterization. Many maximization problems have the property that, perhaps after some simple reduction rules, the optimum value OPT for an instance x is at least $\frac{1}{c} \cdot |x|$. This entails that, if $|x| \geq ck$ then the question whether $OPT \geq k$ is trivially YES, and otherwise we have $|x| < ck$; this is a (trivial) kernelization for the problem. As an example, consider the MAX CUT(k) problem where given a graph $G = (V, E)$ and $k \in \mathbb{N}$ we ask whether there is a bipartition of the vertex set such that at least k edges have endpoints on both sides. It is well known that OPT equals at least half the number m of the edges. Thus, $m \geq 2k$ gives an immediate YES and $m < 2k$ gives a linear kernelization (after discarding isolated vertices). More generally, if we know that $OPT \in \Omega(|x|^{-c})$ then we get a trivial kernelization to size $O(k^c)$.

Motivated by these trivial kernelizations and the fact that the parameter needs to be large to have a nontrivial instance, Mahajan and Raman [61] initiated the study of problems parameterized *above lower bounds*. For example, they considered the MAX CUT($k - \frac{m}{2}$) problem asking whether there is a cut with at least $k = \frac{m}{2} + k'$ edges, parameterized by $k' = k - \frac{m}{2}$, and showed that this problem remains fixed-parameter tractable. Gutin et al. [43] (and follow-up work of Alon et al. [4]) made an important contribution to this direction by introducing the use of the probabilistic method. At high level, they prove that a random solution will exceed the lower bound by at least k with nonzero probability, provided that the instance is sufficiently large compared to k; again (though no longer trivial) this yields either a direct YES or the instance is sufficiently small. Among the further results in this direction let us point out Crowston et al. [17, 16] who obtain further kernelization results.

Meta kernelization. The term *meta kernelization* refers to a series of (positive) kernelization results that apply to a large variety of graph problems when the input graphs are restricted to (in most cases) sparse graph classes such as planar, bounded genus, or H-minor-free graphs [10, 32, 36, 37, 51, 38]. "Meta" here means that the results apply assuming that the problem in question fulfills an appropriate set of technical but rather general properties, obviating the need for any problem-specific reduction rules. A key necessity (but far from sufficient) is, thus, that the problem in question can be formalized in some general language, e.g., monadic second order logic. The first result of this type was obtained by Bodlaender et al. [10], namely linear and polynomial kernelizations for a wealth of problems when restricted to planar or bounded genus graphs. Important predecessors of this work are the linear kernelization for DOMINATING SET in planar graphs by Alber et al. [3] and a more general planar kernelization result, still using problem-specific rules, by Guo and Niedermeier [41].

Most meta kernelization results are based on the following intuition: The cen-

tral notion is that of a *protrusion*, which refers to a subgraph (of the input graph) that is structurally simple and has a limited interaction with the rest of the graph. More concretely, a protrusion has a constant size *boundary* of vertices that are adjacent to the rest of the graph. Furthermore, it has bounded treewidth, which, for the considered problems, implies that we have an efficient dynamic programming routine to solve the problem on the protrusion subgraph (or any other graph of bounded treewidth). The outcome of this dynamic programming is a set of partial solutions relative to the boundary vertices alone. Intuitively, if the problem in question has a bounded number of partial solutions relative to any constant-size boundary, then many protrusions must give rise to the same partial solutions; this is, roughly, captured by the notion of the problem being *finite integer index*. Thus, if we can manage to compute a smaller protrusion with the same partial solutions then this can replace the original protrusion, shrinking the overall instance size. Thus, modulo a significant amount of technical heavy lifting (which we omit), this yields a *protrusion replacement rule* that can be used to replace large protrusions by smaller ones. Apart from this well-behaved interaction with dynamic programming it is required that YES- or NO-instances of the problem in question admit a small set of vertices whose deletion leaves a graph of bounded treewidth. (This holds trivially, for example, for VERTEX COVER(k) or for the FEEDBACK VERTEX SET(k) problem of deleting at most k vertices to obtain a forest.) This can be combined with the topological properties of the input graph class under consideration to prove that the graph can be decomposed into a small number of protrusions, the so-called *protrusion decomposition*.

Let us conclude this part by highlighting recent papers on meta kernelization: Kim et al. [51] recently extended the range of applicable sparse graph classes to classes excluding any fixed graph H as a *topological* minor. Gajarský et al. [36] extended this even further to the larger classes of graphs of bounded expansion, locally bounded expansion, and nowhere dense graphs. This, however, comes at the price that the kernelization bounds are no longer (implicitly) in terms of vertex-deletion distance to bounded treewidth, but instead by distance to bounded *treedepth* (which cannot be avoided [36]). Note also, that, unlike previous work where a low vertex-deletion distance to bounded treewidth is a consequence of other problem properties, Gajarský et al. [36] directly consider the deletion distance to bounded treedepth as the parameter. Independently, Ganian et al. [37] also initiated a study of meta kernelization with respect to structural parameters. Their results apply to problems on *general graphs* and do not require finite integer index. Very recently, Garnero et al. [38] revisited the meta kernelization framework and initiated research into making the obtained kernelization results more explicit. At high level, this is achieved by working more closely on the intuitive connection between meta kernelization and dynamic programming. For an overview on earlier meta kernelization results and a more detailed explanation

thereof we refer to the survey of Lokshtanov et al. [58].

Further new results. Last year, Wahlström [70] came up with an intriguing polynomial compression for the Steiner Cycle(k) problem of finding a cycle (of unbounded length) through a given set of k terminals in a graph. Crucially, the result makes use of the Tutte matrix (and randomization) and, while it obtains an equivalent instance of bounded size, it is not known whether this can be turned into a polynomial kernelization because the output language is not known to be in NP (the connection between compressions and kernelizations will be discussed later).

Fomin et al. [34] proved that Dominating Set(k) and Connected Dominating Set(k) admit linear kernels when restricted to input graphs excluding any fixed graph H as a topological minor. This continues a sequence of results [44, 65, 59, 63, 33, 34] on kernels for (Connected) Dominating Set(k) in restricted graph classes. Note that both problems are W[2]-hard on general graphs and thus do not even admit exponential kernels unless FPT = W[2].

A recent work of Kratsch et al. [54] settled the question of whether the so-called Point Line Cover(k) problem of covering a point set in the plane by at most k lines admits an efficient reduction to significantly less than $O(k^2)$ *points*. (The reader is invited to rediscover a simple reduction to k^2 points that is in the spirit of the Vertex Cover(k) example.) Crucially, the result that no reduction to $O(k^{2-\varepsilon})$ points is possible unless the polynomial hierarchy collapses used the full generality of Dell and van Melkebeek's [22] lower bound framework that applies also to oracle communication protocols. While we will discuss at length the existing lower bound techniques (see Section 4), a discussion of the latter is beyond the scope of this survey.

3 Formal definitions

Formally, a parameterized problem is any language $Q \subseteq \Sigma^* \times \mathbb{N}$, where Σ is any finite alphabet and $\mathbb{N}$ denotes the non-negative integers. The second component k of any instance $(x, k) \in \Sigma^* \times \mathbb{N}$ is called the *parameter*. The problem Q is *fixed-parameter tractable* (FPT) if there is an algorithm A, a computable function $f\colon \mathbb{N} \to \mathbb{N}$, and a constant c such that A correctly decides $(x, k) \in Q$ for all $(x, k) \in \Sigma^* \times \mathbb{N}$ in time $f(k) \cdot |x|^c$. We omit in this survey a detailed discussion of *fixed-parameter intractability*, e.g., regarding fpt-reductions and the W-hierarchy. It suffices to know that intractability is typically established by proving W[1]- or W[2]-hardness;[4] note that FPT $\subseteq$ W[1] $\subseteq$ W[2] and it is believed that the inclusions are strict.

[4] E.g., Clique(k) is W[1]-complete and Hitting Set(k) is W[2]-complete.

A kernelization for a parameterized problem Q is a polynomial-time algorithm K that given any instance $(x,k) \in \Sigma^* \times \mathbb{N}$ returns an instance (x',k') such that $(x,k) \in Q$ if and only if $(x',k') \in Q$ and with $|x'|, k' \leq f(k)$ for some computable function $f\colon \mathbb{N} \to \mathbb{N}$. The function f is called the *size of the kernelization* K and K is a *polynomial (linear) kernelization* if $f(k)$ is polynomially (linearly) bounded in k. For simplicity, we allow a kernelization to outright answer YES or NO, understanding that it could instead return any hard-wired YES- or NO-instance of Q (of constant size). It is known that a parameterized problem is fixed-parameter tractable if and only if it is decidable and admits a kernelization (see Theorem 1 below).

In the literature there exist two relaxed variants of kernelization: A *generalized kernelization* (or *bikernel*) returns an output instance (x',k') that is with respect to a, possibly different, parameterized problem Q'. More general, a *compression* may return an instance with respect to any (also unparameterized) language $L \subseteq \Sigma^*$. All kernelization lower bound tools in this survey, and almost all lower bounds in the literature, imply also the same lower bounds for compressions. We will see later (in Section 4) that lower bounds for compressions are slightly preferable, due to greater ease of transferring them by appropriate reductions.

Theorem 1. *A parameterized problem Q is fixed-parameter tractable if and only if it is decidable and has a kernelization.*

Proof. Assume that we have a kernelization for Q that reduces any instance (x,k) to an equivalent instance (x',k') of size at most $f(k)$. We can then apply an arbitrary algorithm for Q (guaranteed by decidability) to solve (x',k') and thereby also (x,k). If $g\colon \mathbb{N} \to \mathbb{N}$ bounds the runtime of the assumed algorithm then the total time investment is $|x|^{O(1)}$ for the kernelization plus $g(f(k))$ for the algorithm. This is bounded by $f'(k)|x|^{O(1)}$ where $f'(k) := g(f(k))$, implying fixed-parameter tractability.

For the converse, assume that we have an algorithm that solves all instances (x,k) of Q in time $f(k)|x|^c$. Now run this assumed algorithm for $|x|^{c+1}$ steps. If it finishes then we have the correct YES or NO answer. Otherwise, it did not finish cause $f(k)|x|^c > |x|^{c+1}$. This, however, implies $|x| < f(k)$. Thus, either way, in polynomial time $O(|x|^{c+1})$ we get an equivalent instance of size at most $f(k)$. □

Note that the kernelizations implied by this theorem are not very useful cause the size bound $f(k)$ is the same $f(k)$ as in the FPT runtime, which is usually exponential in k. Nevertheless, the existence of exponential kernelizations for many problems further motivates the question which of them also have polynomial kernelizations. Conversely, if a problem is W[1]-hard and thus not FPT unless FPT = W[1] then we also expect no kernelization.

4 Lower bounds for kernelization

The goal of this section is to explain the basic intuition underlying known techniques for lower bounds for kernelization and to give *one* complete set of tools for proving them. To this end, we will formally define so-called *cross-compositions* and *polynomial parameter transformations* as these appear very convenient to use. Cross-composition is a unifying front end to various insightful tools, and complexity theorists might prefer to directly employ these underlying results of, e.g., Dell and van Melkebeek [22] and Drucker [26].

At high level, there are two prevalent forms of kernelization lower bounds known so far: First, and dominantly, for a wealth of problems it has been shown that they admit no polynomial kernelization unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. Second, for a smaller list of problems that do have polynomial kernels, it is known that no kernels of size $O(k^{c-\varepsilon})$ are possible, where k is the parameter and c is some constant, unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. The assumption that $\mathsf{NP} \not\subseteq \mathsf{coNP/poly}$ (or, equivalently, $\mathsf{coNP} \not\subseteq \mathsf{NP/poly}$) is clearly stronger than $\mathsf{P} \neq \mathsf{NP}$ and $\mathsf{NP} \not\subseteq \mathsf{coNP}$ but, since its failure would imply a collapse of the polynomial hierarchy [71, 14], it is still widely believed.

Intuition for ruling out polynomial kernels. Let us consider the NP-hard Path(k) problem where we are given a graph $G = (V, E)$ and $k \in \mathbb{N}$ with the question of whether G contains a simple path on at least k vertices. If we combine t instances $(G_1, k), \ldots, (G_t, k)$ into a single one (G', k) by letting G' be the disjoint union of the graphs G_i then, clearly, (G', k) is yes if and only if at least one (G_i, k) is yes. Intuitively, for t large but polynomial in k, a kernelization applied to (G', k) would have to determine some graphs G_i that are less likely to be yes and remove the corresponding components from G'. More concretely, if we assume a kernelization to size k^c and take $t = k^{c+1}$ then the output of the kernelization applied to G' has less than one bit per instance (G_i, k). On the other hand, the total input size is polynomial in the largest instance (G_i, k) and, hence, we do not expect that (in general) the time would suffice to solve any of the instances.

More generally, we do not expect an efficient algorithm that for $s \in \mathbb{N}$ takes t instances of any NP-hard problem, each of size at most s, and returns a single instance of size polynomial in s that is yes if at least one of the inputs is yes. Such an algorithm is called an or-*distillation* in the breakthrough lower bound framework of Bodlaender et al. [9]; and they conjectured that no NP-hard problem admits an or-distillation. The conjecture was proved shortly after by Fortnow and Santhanam [35] modulo the assumption that $\mathsf{NP} \not\subseteq \mathsf{coNP/poly}$. The analogous conjecture for the natural variant called and-*distillation* was made as well, but it remained an open problem for five years until it was settled by an impressive work of Drucker [26]; amongst a wide range of results on both deterministic and

probabilistic compression (in fact also for quantum compression) Drucker proved that the AND-distillation conjecture holds under NP $\not\subseteq$ coNP/poly as well.

The framework of Bodlaender et al. [9] introduced so-called OR- *and* AND-*composition* algorithms that, essentially, generalize the above example for PATH(k) to any efficient mapping (not just disjoint union and not just for graph problems) that encodes the OR or AND of t instances with parameter value k into a single instance of the same problem with parameter value k' polynomially bounded in k. I.e., given t instances the obtained instance is YES if and only if at *least one* respectively *all* given instances are YES. Similarly to the example, such a composition together with a polynomial kernelization gives an OR- or AND-distillation. Since proving existence of a particular algorithm (the composition) is typically easier than ruling out an algorithm (the polynomial kernelization) proving compositions became a very successful way of ruling out polynomial kernels. Curiously, even before Drucker's result [26], most lower bounds used OR-compositions and only very few proofs had to rely on the then unproven AND-distillation conjecture.

Cross-composition. We will now review an extension to the composition-based framework that was introduced by Bodlaender at al. [12]. In a so-called OR- resp. AND-cross-composition the input consists of instances of *any* NP-*hard* problem, while the output is an instance of the target parameterized problem for which we desire a lower bound. Essentially, the parameter of the output instance must be polynomially bounded in the largest size among input instances, which often makes the proofs easier. In addition, there is the straightforward notion of a so-called *polynomial equivalence relation* that simplifies arguments for why inputs to a (cross-)composition may be assumed to be fairly similar (e.g., you may have wondered why we tacitly assumed that all PATH(k) inputs have the same parameter).

Despite these extensions to the composition-based framework [9, 35, 26] the underlying ideas go through in the same way. Nevertheless, several fairly ad-hoc tricks needed for compositions are no longer required for cross-compositions and this front end has seen wide adoption.

Definition 1 (polynomial equivalence relation [12])**.** An equivalence relation $\mathcal{R}$ on Σ^* is called a *polynomial equivalence relation* if the following two conditions hold:

1. There is an algorithm that given two strings $x, y \in \Sigma^*$ takes time polynomial in $|x|+|y|$ and decides whether x and y belong to the same equivalence class.

2. For any finite set $S \subseteq \Sigma^*$ the equivalence relation $\mathcal{R}$ partitions the elements of S into a number of classes that is polynomially bounded in the size of the largest element of S.

A simple example usage of a polynomial equivalence relation for PATH(k) instances (G_i, k_i) would be to declare instances (G_i, k_i) and (G_j, k_j) equivalent if $k_i = k_j$. (As a technical remark, if k is given in binary then this would formally allow an exponential number of equivalence classes. Thus, one usually resorts to a dummy class containing "ill-posed" or otherwise infeasible inputs. E.g., for PATH(k) we can make one class for all instances where k exceeds the number of vertices since these are trivially NO.)

Definition 2 (AND/OR-cross-composition [12])**.** Let $L \subseteq \Sigma^*$ be a language, let $\mathcal{R}$ be a polynomial equivalence relation on Σ^*, and let $Q \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized problem. An OR-*cross-composition of L into Q* (with respect to $\mathcal{R}$) is an algorithm that, given t instances $x_1, x_2, \ldots, x_t \in \Sigma^*$ of L belonging to the same equivalence class of $\mathcal{R}$, takes time polynomial in $\sum_{i=1}^{t} |x_i|$ and outputs an instance $(y, k) \in \Sigma^* \times \mathbb{N}$ such that:

"PB": The parameter value k is polynomially bounded in $\max_i |x_i| + \log t$.

"OR": The instance (y, k) is YES for Q if and only if *at least one* instance x_i is YES for L.

An AND-*cross-composition of L into Q* (with respect to $\mathcal{R}$) is an algorithm that, instead, fulfills Properties "PB" and "AND".

"AND": The instance (y, k) is YES for Q if and only if *all* instances x_i are YES for L.

We say that L OR-cross-composes, respectively AND-cross-composes, into Q if a cross-composition algorithm of the relevant type exists for a suitable relation $\mathcal{R}$.

Note that the use of a polynomial equivalence relation in the definition is, effectively, optional since $\mathcal{R} = \Sigma^* \times \Sigma^*$ is a valid choice and simply makes all inputs equivalent. The *intended use* of polynomial equivalence relations, however, is to group inputs for a cross-composition such that it need only be applied to groups of instances that are somewhat similar, thereby simplifying the necessary constructions and gadgets.

Similar to compositions, any AND- or OR-cross-composition combined with a polynomial kernelization creates an AND- or OR-distillation. Thus, using the results of Fortnow and Santhanam [35] and Drucker [26] we can use them to rule out polynomial kernelizations.

Theorem 2 ([12])**.** *If an* NP*-hard language L* AND/OR*-cross-composes into the parameterized problem Q, then Q does not admit a polynomial kernelization or polynomial compression unless* NP $\subseteq$ coNP/poly *and the polynomial hierarchy collapses.*

Note that the theorem also rules out polynomial compressions, which relax polynomial kernelizations by allowing the output to be an instance (a string) with respect to *any language*; in the same way this holds also for lower bounds via AND- and OR-compositions. This simplifies transferring lower bounds via appropriate reductions (as we will see later).

An example for AND-cross-composition. We will now sketch an AND-cross-composition for the EDGE CLIQUE COVER(k) problem. The question about existence of a polynomial kernelization for EDGE CLIQUE COVER(k) was a frequently posed open problem (see, e.g., Guo and Niedermeier [40]) until being settled negatively by Cygan et al. [19].

> EDGE CLIQUE COVER(k)
> **Input:** A graph $G = (V, E)$ and $k \in \mathbb{N}$.
> **Parameter:** k.
> **Question:** Is there a collection of at most k cliques in G such that each edge is contained in at least one of them?

We give an AND-cross-composition from EDGE CLIQUE COVER to EDGE CLIQUE COVER(k) following in spirit the construction of Cygan et al. [19]. (Note that EDGE CLIQUE COVER has the same problem definition as EDGE CLIQUE COVER(k), including the value $k \in \mathbb{N}$, except for not specifying k as the parameter.) We begin by choosing a polynomial equivalence relation. We make one equivalence class for all instances that are trivially YES because k exceeds the number of edges. Among the rest, let any two instances (G_i, k_i) and (G_j, k_j) be equivalent if G_i and G_j have the same number of vertices and furthermore $k_i = k_j$. Finally, since we are careful theoreticians, we devote one class to all inputs that are not valid encodings of a graph and integer k (and which are thus NO instances). Of course, in the following it suffices to discuss the interesting case of inputs that are not trivially YES or NO.

Let t instances from the same (nontrivial) equivalence class be given, e.g., $(G_1, k), \ldots, (G_t, k)$. Let n be the number of vertices in each graph and, for convenience, assume that the vertices of each graph G_i are numbered arbitrarily, say $V_i = \{v_{i,1}, \ldots, v_{i,n}\}$.

The basic idea is to start with a disjoint union of the graphs and add all edges between different graphs (i.e., we take the join of the graphs). Then, if all instances are YES, we may combine the t times k cliques used for the graphs into k cliques that cover all edges in graphs G_i. Concretely, say that for $i \in \{1, \ldots, t\}$ the edges of G_i can be covered by cliques $C_{i,1}, \ldots, C_{i,k}$. Then for $j \in \{1, \ldots, k\}$ each set $\widehat{C}_j := \bigcup_i C_{i,j}$ induces a clique (using join edges), and together these k cliques cover all edges *inside* each graph G_i.

The caveat, however, is that the combination of the cliques does not necessarily cover *all* join edges that we introduced between different graphs G_i. We

handle this situation by increasing the budget and forcing inclusion of additional $O(n \log t)$ cliques that cover all join edges *but do not contain any edge in any graph* G_i. If we can ensure this, then the remaining budget of k will allow only k further cliques, like, e.g., $\widehat{C}_1, \ldots, \widehat{C}_k$, that must induce a k-clique cover in each graph G_i.

The idea is to add auxiliary vertices that will each be adjacent to exactly one vertex $v_{i,\ell}$ per graph G_i. To ensure that we cover all edges between any graphs G_i and G_j the exact choice for each auxiliary vertex depends on the binary expansion of i and j (using that different numbers differ in at least one position, but avoiding the use of $O(t)$, or worse, many extra vertices/cliques).

We introduce auxiliary vertices $w_{a,b,p}$ for all $a, b \in \{1, \ldots, n\}$ and $p \in \{1, \ldots, \log t\}$. We connect a vertex $w_{a,b,p}$ to vertex $v_{i,a}$ of graph G_i if the pth bit in the binary expansion of i is even, and to $v_{i,b}$ otherwise (if the bit is odd). We call the obtained graph (of G_i's and auxiliary vertices) G' and let the budget be $k' := k + n^2 \cdot \log t$. Since we already excluded instances with k exceeding the number of edges, which is less than n^2, the value k' is indeed polynomially bounded in the largest input instance plus $\log t$.

Let us briefly check that the obtained instance behaves as intended. Crucially, the auxiliary vertices form an independent set and none of them is isolated. Thus, we need to include at least one separate clique for each of them. Clearly, the closed neighborhood of any $w_{a,b,p}$ is a clique since all neighbors are adjacent by join edges. Thus, a single clique per $w_{a,b,p}$ is necessary and sufficient. For any join edge from, say, $v_{i,a}$ to $v_{j,b}$, we find that both vertices are contained in the neighborhood of $w_{a,b,p}$ or $w_{b,a,p}$ for all positions p where the binary expansions of i and j differ (the choice of $w_{a,b,p}$ or $w_{b,a,p}$ depends on the respective parities in position p). At this point, all join edges are covered and all edges inside graphs G_i still need to be covered by the remaining k cliques (which can be combined over all t graphs). Thus, the instance (G', k') correctly encodes the AND and by Theorem 2 this rules out polynomial kernels and compressions for EDGE CLIQUE COVER(k).

Polynomial parameter transformations. Before the framework of Bodlaender et al. [9] the question for lower bounds for kernelization was frequently posed as an open problem. It is surprising, in hindsight, that this never led to a reduction-based study of polynomial kernels akin to the collective evidence created by NP-complete problems. In contrast, shortly after the framework was published, it was recognized that compositions are by no means always as easy as for PATH(k) and may sometimes be outright impossible.[5]

[5]This problem was mainly with the original notion of compositions, where source and target problem needed to be the same.

It was soon recognized that having a Karp reduction from one parameterized problem to another with the additional restriction that the output parameter is polynomially bounded in the input parameter essentially preserves kernelization properties (we will formalize this in a moment). This was first, implicitly, used by Binkele-Raible [6], first made formal by Bodlaender et al. [13], and first heavily used by Dom et al. [23]. We introduce these reductions under the widely adopted name of *polynomial parameter transformations*.

Definition 3 (polynomial parameter transformation)**.** Let $Q, Q' \subseteq \Sigma^* \times \mathbb{N}$ be parameterized problems. A *polynomial parameter transformation* (PPT) from Q to Q' is a polynomial-time computable mapping $\pi\colon \Sigma^*\times\mathbb{N} \to \Sigma^*\times\mathbb{N}\colon (x,k) \mapsto (x',k')$ such that $(x,k) \in Q$ if and only if $(x',k') \in Q'$ and $k' \leq p(k)$ for all $(x,k) \in \Sigma^* \times \mathbb{N}$, where $p\colon \mathbb{N} \to \mathbb{N}$ is some fixed polynomial. If there is such a reduction from Q to Q' then we write $Q \leq_{ppt} Q'$.

If $Q \leq_{ppt} Q'$ and Q' has a polynomial kernelization (or compression) then we can take any instance (x,k) for Q, compute an equivalent instance (x',k') of Q' with k' polynomially bounded in k, and then apply the kernelization/compression of Q'. The obtained instance, say (x'',k'') of Q' is YES if and only if (x,k) is YES for Q and its size is polynomially bounded in k. Thus, the combined algorithm of PPT plus polynomial kernelization/compression constitutes a polynomial compression for Q. This yields the following simple but useful lemma for proving lower bounds.

Lemma 1. *If $Q \leq_{ppt} Q'$ and Q admits no polynomial compression (possibly modulo some complexity assumption) then Q' admits no polynomial kernel or compression (under the same assumption).*

Note that to combine a PPT from Q to Q' and a polynomial kernelization for Q' into a polynomial *kernelization* for Q we still need to convert the output, which is a $poly(k)$-sized instance for Q', into an instance for Q without blowing up size and parameter more than polynomially. If Q is NP-hard and $Q' \in \mathsf{NP}$ then we can use the implied Karp reduction from Q' to Q; a technicality, however, is that we need NP-hardness of Q for polynomially bounded value of its parameter (or, equivalently, with parameter value encoded in unary) to ensure that there is a Karp reduction that also implies a polynomial bound for the parameter (see Bodlaender et al. [13]).

We will make further use of PPTs in Section 6. Let us anyway copy a nice example from [58]: In the 2-PATH(k) problem, given (G,k) we need to find two vertex-disjoint simple paths of length k each. The disjoint union composition fails, since we might have two input graphs *with only one k-path each*. There is, however, a simple PPT from PATH(k) to 2-PATH(k): Given a PATH(k) instance

(G, k), simply return (G', k) where G' is obtained from the disjoint union of G and a k-path. Clearly, G has a k-path if and only if G' contains two vertex-disjoint k-paths.

Let us add to the example that there is also a simple OR-cross-composition from PATH(k) to 2-PATH(k), either by disjoint union with *two copies* of each input graph or by similarly adding one additional disjoint k-path.

Polynomial lower bounds for kernelization. So far we have discussed how to rule out polynomial kernels for certain parameterized problems. An insightful work of Dell and van Melkebeek [22] was the first to open up the possibility of proving polynomial lower bounds for problems that do admit some polynomial kernelization. E.g., they showed that d-HITTING SET(k) admits no kernelization to size $O(k^{d-\varepsilon})$ for any fixed $\varepsilon > 0$ unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. In fact, their bounds are more general and apply also to compressions and, interestingly, to a form of oracle communication protocol. For reasons of space (and focus) we restrict ourselves to the goal of discussing polynomial lower bounds, but strongly suggest a follow-up reading of [22].

The key step for getting to polynomial lower bounds was a closer inspection of Fortnow and Santhanam's [35] proof of the OR-distillation conjecture [22]. This revealed that, roughly speaking, an efficient algorithm that encodes the OR of any t instances for L into an equivalent instance of L' of length $O(t \log t)$ implies $L \in \mathsf{coNP/poly}$. More concretely, we need such an algorithm that works when given $t := t(n)$ instances of size at most n each for any value of n, where t is any polynomially bounded function. A similar statement follows for encoding the AND of t instances of L (see Theorem 4) as one of many consequences of Drucker's work [26].

To sketch how this gives polynomial lower bounds let us first see how it works for ruling out all polynomial kernels. If we have an OR-cross-composition of some L into a parameterized problem that yields parameter $k \in O(n^c)$ then applying *any polynomial kernelization* yields a total size of $O(k^d) \subseteq O(n^{cd})$. If we apply the combined algorithm to $t = n^{cd}$ instances then this makes the total size $O(n^{cd}) \subseteq O(t)$. Hence, for any assumed polynomial kernelization we can choose $t\colon \mathbb{N} \to \mathbb{N}$ such that we get "OR of t instances into $O(t)$ bits", implying $L \in \mathsf{coNP/poly}$.

Now, assume instead that we can encode the OR of t instances of L of size n each into one instance with parameter $k \in O(t^{1/2}n^c)$. Using *any kernelization with size guarantee* $O(k^{2-\varepsilon})$ would now give total size $O(t^{1-\varepsilon'}n^{c'})$. This again, for an appropriate function $t\colon \mathbb{N} \to \mathbb{N}$, suffices to get "OR of t instances into $O(t)$ bits" and, hence, $L \in \mathsf{coNP/poly}$.

We will next define an extension of AND/OR-cross-composition that allows for such larger contributions of the number t of instances in the parameter obtained

by the compositions. Again, this is a front end to very insightful works [22, 26], and, hopefully, motivates more applications of their results.

Definition 4 (AND/OR-cross-composition of bounded cost [12])**.** An AND/ OR-*cross-composition of L into Q* (with respect to $\mathcal{R}$) *of cost $f(t)$ for t instances* is an AND/OR-cross-composition algorithm as described in Definition 2 that satisfies "CB" instead of "PB".

"CB": The parameter k is bounded by $O(f(t) \cdot (\max_i |x_i|)^c)$, where c is some constant independent of t.

The following theorem formalizes the intuition of how the dependence on t in an AND/OR-cross-composition relates to polynomial lower bounds.

Theorem 3 ([12])**.** *Let $L \subseteq \Sigma^*$ be a language, let $Q \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized problem, and let d, ε be positive reals. If L has an* AND/OR*-cross-composition into Q with cost $f(t) = t^{1/d+o(1)}$, where t denotes the number of instances, and Q has a polynomial compression into an arbitrary language L' with size bound $O(k^{d-\varepsilon})$, then $L \in \mathsf{coNP/poly}$. If, additionally, L is* NP*-hard, then $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.*

The statement for OR-cross-composition was proved in [12] building on [22]. The analogous proof for AND-cross-compositions is given here for the first time. Modulo swapping of AND and OR and avoiding the use of the oracle communication protocol this proof is fully analogous to the OR-cross-composition case. Crucially, however, the proof depends on having a proven consequence of encoding the AND of t instances of any L into $O(t \log t)$ bits, which follows as a consequence of a more powerful result of Drucker [26, 27].[6]

Theorem 4 (Consequence of [27, Theorem 7.1])**.** *Let L, L' be any languages, let $d > 0$, and let $t \colon \mathbb{N} \to \mathbb{N}$ be polynomially bounded. Suppose that there exists a polynomial-time mapping that on input of $t := t(n)$ instances $x_1, \ldots, x_t$ for L each of size n computes a single instance x of size at most $d \cdot t \log t$ such that $x \in L'$ if and only if $x_i \in L$ for all i. Then $L \in \mathsf{coNP/poly}$.*

Proof. This follows as an application of the more general [27, Theorem 7.1]. First, we need to swap the role of AND and OR by complementation to match [27, Theorem 7.1]: Assume a mapping that given $x_1, \ldots, x_t$ returns x with $x \in L'$ if and only if $x_i \in L$ for all i. If we consider $\overline{L}$ and $\overline{L'}$ instead then we get $x \in \overline{L'}$ if and only $x_i \in \overline{L}$ for at least one i. Once we have chosen all other parameters we can thus apply our mapping as an OR for $\overline{L}$ in [27, Theorem 7.1] which implies $\overline{L} \in \mathsf{NP/poly}$ and $L \in \mathsf{coNP/poly}$.

[6] The author is indebted to Andrew Drucker for clarifying how this follows from his work.

We use the following choices for $t_1(n)$, $t_2(n)$, $\widehat{\delta}$, and $\xi(n)$: We have an error-free mapping and, thus, use error bound $\xi(n) = 0$. We set $t_1(n) := t(n)$ and $t_2(n) := d \cdot t(n) \log t(n)$. Using the definition of $\widehat{\delta}$ in [27, Theorem 7.1], this yields $\widehat{\delta} \leq 1 - \frac{1}{8}(t(n))^{-d}$. Since t is polynomially bounded, there are constants a, b such that $t(n) \leq a \cdot n^b$ for sufficiently large n. Our parameters fulfill the requirement of $1 - 2\xi(n) - \widehat{\delta} \geq \frac{1}{n^c}$ in [27, Theorem 7.1] for $c = bd + 1$:

$$1 - 2\xi(n) - \widehat{\delta} \geq \frac{1}{8 \cdot (t(n))^d} \geq \frac{1}{8 \cdot a \cdot n^{bd}} \geq \frac{1}{n^c},$$

for sufficiently large n. □

Now we can explain the proof of Theorem 3. It follows the basic intuition given earlier and is analogous to the OR case in Bodlaender et al. [12].

Proof of Theorem 3 for AND*-cross-compositions.* Let $\mathcal{R}$ denote a polynomial equivalence relation on Σ^* which partitions any set of strings of length at most s into at most $O(s^b)$ equivalence classes. Let $f(t) = t^{1/d+o(1)}$ for some constant d. Let C be an AND-cross-composition from L into Q, which maps t instances of size at most s and from the same $\mathcal{R}$-equivalence class to an output instance with parameter value bounded by $O(f(t)s^c)$. Finally, let K be a polynomial compression for Q into some language L' that given an instance with parameter k outputs an equivalent string (with respect to L') of size bounded by $h(k) = O(k^{d-\varepsilon})$.

We define a polynomially bounded function t by $t(s) := s^{(b+cd)\cdot\frac{d}{\varepsilon}}$. By Theorem 4 it suffices to provide an appropriate encoding of the AND of t instances of L. As the target language we will use $\text{AND}(L') := \{(x_1, \ldots, x_r) \mid r \in \mathbb{N} \wedge x_1, \ldots, x_r \in L'\}$. Fixing s and $t := t(s)$, let t instances $x_1, \ldots, x_t$ of L each of length at most s be given.

As a first step, we partition the strings x_i according to equivalence under $\mathcal{R}$, obtaining $r \leq O(s^b)$ groups. Then we apply the AND-cross-composition C to each group, obtaining r instances $(y_1, k_1), \ldots, (y_r, k_r)$. The parameter values k_i are bounded by $O(f(t)s^c)$. Now we apply the assumed polynomial compression K to each instance (y_i, k_i), obtaining instances $z_1, \ldots, z_r$ of the language L'. We return the instance $(z_1, \ldots, z_r)$.

Each compressed instance z_i has size at most

$$h(k_i) = O((k_i)^{d-\varepsilon}) = O((f(t)s^c)^{d-\varepsilon}).$$

Thus we can bound the output size, i.e., the size of $(z_1, \ldots, z_r)$, as follows:

$$O\left(r\,(f(t)s^c)^{d-\varepsilon}\right) = O\left(s^b\left(t^{\frac{1}{d}+o(1)}s^c\right)^{d-\varepsilon}\right) = O\left(s^{b+c(d-\varepsilon)}t^{1-\frac{\varepsilon}{d}+o(1)}\right) = O(t),$$

using that $r \leq O(s^b)$ and the following bound for $s^{b+c(d-\varepsilon)}$:

$$s^{b+c(d-\varepsilon)} = s^{b+cd} \cdot s^{-c\varepsilon} = t^{\frac{\varepsilon}{d}} \cdot s^{-c\varepsilon} = t^{\frac{\varepsilon}{d}-\delta},$$

where $\delta = \frac{c\varepsilon^2}{(b+cd)d} > 0$. (Note that $t^{1-\delta+o(1)} = O(t)$, for any $\delta > 0$.)

Correctness. It remains to show that the returned instance $(z_1, \ldots, z_r)$ is indeed an encoding of the AND of the instances $x_1, \ldots, x_t$. Assume first that at least one input instance x_i is a NO-instance (requiring the output to be NO for AND(L')). It follows that the corresponding instance (y_j, k_j) that is created by C from all instances $\mathcal{R}$-equivalent to x_i must be NO for Q. Accordingly, the polynomial compression K transforms (y_j, k_j) to a NO-instance z_j for the language L'. Hence, the output instance $(z_1, \ldots, z_r)$ is NO for AND(L').

In the remaining case all input instances $x_1, \ldots, x_t$ are YES for L. The AND-cross-composition C will therefore create r YES-instances (y_i, k_i) for Q. These are converted to r YES-instances z_i for L'. Hence, the returned instance $(z_1, \ldots, z_r)$ is YES for AND(L'). Thus, we get a polynomial-time mapping fulfilling the requirement of Theorem 4. It follows that $L \in \mathsf{coNP/poly}$, as claimed. If L is NP-hard then $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. □

To conclude the section on lower bounds for kernelization, let us illustrate a successful "design-paradigm" for proving polynomial lower bounds that has been identified through results of Dell and van Melkebeek [22] and Dell and Marx [21]. The idea is to use a source problem that is d-partite in a sense. More strongly, similar to, for example, problems on bipartite graphs, all the relevant information needs to be encoded in the adjacency (or other structure) between the partite sets; the partite sets themselves should be isomorphic over all input instances (here polynomial equivalence relations can be of help). Thus, one can tightly encode t instances of a bipartite problem by using only $\sqrt{t}$ copies each of both partite sets and choosing a different pair for each instance. Let us perhaps make this more concrete in the following example.

Example of a polynomial lower bound. As an illustration let us sketch an $O(n^{d-\varepsilon})$ lower bound for the d-HITTING SET(n) problem for any fixed $d \geq 3$. We give an OR-cross-composition from HITTING SET restricted to d-partite d-uniform hypergraphs, which is NP-hard for $d \geq 3$ (cf. [42]). In that problem we have a given partition of the ground set U into d color classes, say $U = C_1 \cup \ldots \cup C_d$ with each hyperedge containing exactly one vertex from each set C_i, and the task is to find k elements of U that intersect all edges (if possible).

Let t instances $(U_i, \mathcal{F}_i, k)$ of HITTING SET on d-partite d-uniform hypergraphs be given. For simplicity, skipping over padding arguments and choice of polynomial equivalence relation, assume that the ground set U_i of each instance is partitioned

into d color classes, each containing exactly n vertices. As a first step, rename the instances from $i \in \{1, \ldots, t\}$ to $\mathbf{i} \in \{(i_1, \ldots, i_d) \mid i_j \in \{1, \ldots, t^{1/d}\}$; a simple counting argument shows that this allows an injective renaming.

Now, rather than taking simply the disjoint union of the instances we carefully identify the color classes of different instances. Concretely, for $p \in \{1, \ldots, d\}$ and $q \in \{1, \ldots, t^{1/d}\}$ identify, vertex by vertex, the pth color class of all instances with number $\mathbf{i} = (i_1, \ldots, i_d)$ with $i_p = q$. In this way, for each color $p \in \{1, \ldots, d\}$ we end up with $t^{1/d}$ color classes (each with n vertices) that are shared by several instances. Let $C_{p,q}$ for $p \in \{1, \ldots, d\}$ and $q \in \{1, \ldots, t^{1/d}\}$ denote the obtained color classes.

Now, for all colors p and any two vertices u and v in different color classes $C_{p,q}$ (i.e., with different values of q) we add a new edge $\{u, v\}$. Thus, any hitting set for the instance has to *completely contain all but one color class $C_{p,q}$ for each color p*. Let us see what happens if, taking this into account, we ask for a hitting set of total size at most $k' = d(t^{1/d} - 1)n + k$ for the combined instance of d-Hitting Set(n).

As just observed any k'-hitting set, say S, must contain all but one color class $C_{p,q}$ for each color p. Let $q_1, \ldots, q_d \in \{1, \ldots, t^{1/d}\}$ such that $C_{p,q_p} \not\subseteq S$ for all p, i.e., each q_i corresponds to the color class that is not fully contained in S. Since $|S| \leq k'$ we find that the intersection of S with $C_{1,q_1} \cup \ldots \cup C_{d,q_d}$ is of size at most k; let S' denote the intersection. It follows that S' is a k-hitting set for *all edges that are fully contained in* $C_{1,q_1} \cup \ldots \cup C_{d,q_d}$. Note that, during our identification process, all color classes of instance $\mathbf{i}$ with $\mathbf{i} = (q_1, \ldots, q_d)$ have been identified with $C_{1,q_1}, \ldots, C_{d,q_d}$ and all its hyperedges are, therefore, contained in $C_{1,q_1} \cup \ldots \cup C_{d,q_d}$. Thus, S' is a k-hitting set for instance $\mathbf{i}$, proving that at least one input is yes.

For the converse, if some instance $(U_i, \mathcal{F}_i, k)$ is yes then begin by letting S' a k-hitting set for that instance. Let $\mathbf{i} = (i_1, \ldots, i_d)$ be the assigned renaming of i. Now, let S contain S' as well as all color classes $C_{p,q}$ with $q \neq i_p$, i.e., all color classes not used for instance $\mathbf{i}$. Clearly, this covers all additional edges between color classes $C_{p,q}$ and $C_{p,q'}$ with $q \neq q'$. Furthermore, for every instance $\mathbf{i}' = (i'_1, \ldots, i'_r) \neq (i_1, \ldots, i_r) = \mathbf{i}$ at least one position must differ, e.g., $i'_p \neq i_p$. But then S already includes all vertices of C_{p,i'_p} covering all hyperedges of instance $\mathbf{i}'$. Thus, the constructed instance is yes.

To wrap up, note that the combined instance has exactly $n' = d \cdot t^{1/d} \cdot n$ vertices, which is bounded by $t^{1/d}$ times a polynomial in the largest instance size. Thus, we have an or-cross-composition with cost $t^{1/d}$ implying that d-Hitting Set(n) has no kernelization with size $O(n^{d-\varepsilon})$ for any $\varepsilon > 0$ unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. As in [22] the analogous bound for d-Hitting Set(k) follows immediately by noting that all nontrivial instances have $k \leq n$.

Further reading. We point out some more results regarding polynomial lower bounds for concrete problems since, unlike ruling out polynomial kernels altogether, this is not yet in common use. Independently from Dell and Marx [21], Hermelin and Wu [47] formalized a form of composition algorithms with larger dependence on the number t of composed instances, which they called *weak compositions*. Both papers prove polynomial lower bounds for several standard problems when restricted to families of sets of bounded size or graphs of bounded degree, respectively. A recent work of Cygan et al. [18] obtains kernelization lower bounds for several problems when restricted to graphs of bounded degeneracy that almost exactly match known upper bounds. Jansen [48] used the polynomial lower bound framework to rule out sparsification for computing the treewidth of a graph by proving that the problem admits no polynomial compression to size $O(n^{2-\varepsilon})$, which would, for example, be implied by any nontrivial reduction to the number of edges. Generally, also the initial results of Dell and van Melkebeek [22] had sparsification lower bounds as one of their goals.

5 Representative sets and matroids

In this section we give an introduction to using representative sets and matroids for kernelization. As a warm-up, we will begin by introducing representative sets for set families and using them to reproduce two "classic" kernelization results, namely polynomial kernels for d-Hitting Set(k) and d-Set Packing(k). (See below for problem definitions.) It is known that kernels for these two problems can also be obtained via the Sunflower Lemma of Erdős and Rado [28]; see, e.g., [30, 21]. The best known kernelizations for both problems are due to Abu-Khzam [2, 1], with a slightly smaller ground set of $O(k^{d-1})$ but same asymptotic total size of $O(k^d \log k)$. It is known, by work of Dell and van Melkebeek [22] and Dell and Marx [21], that neither result can be improved to size $O(k^{d-\varepsilon})$ unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

In the second part we move on to using representative sets on families of independent sets of a given matroid. A 1977 result of Lovász [60] states that such sets, of modest size, exist for every linear matroid, i.e., for every matroid that can be represented as the column matroid of a matrix. Marx [62] observed that Lovász' proof in fact also gives rise to an efficient algorithm. Since then, representative sets, both for set families (or, equivalently, uniform matroids) but also for gammoids and graphic matroids, have found various applications in parameterized complexity for kernelization [57] and faster algorithms [31]. In particular, Fomin et al. [31] also gave faster algorithms for finding representative sets for both linear matroids and the special case of uniform matroids. To illustrate the use for kernelization, we will give a fairly detailed description of the polynomial

kernelization for Deletable Terminal Multiway Cut(k) obtained in [57].

Representative sets for set families. Let us jump right in and give a definition of q-representativeness for the case of set families.

Definition 5 (q-representative set family)**.** Let $\mathcal{A}$ be a family sets and let $q \in \mathbb{N}$. A subset $\mathcal{A}' \subseteq \mathcal{A}$ is *q-representative for $\mathcal{A}$* if for every set B of size at most q there is a set $A \in \mathcal{A}$ with $A \cap B = \emptyset$ if and only if there is a set $A' \in \mathcal{A}'$ with $A' \cap B = \emptyset$.

We will later give a similar definition for representative independent sets in a specified matroid (see Definition 6) that additionally requires $A \cup B$ and $A' \cup B$ to be independent sets of the matroid. The present definition can then be seen as a special case by using so-called uniform matroids where all sets up to some prescribed size are independent, but this is not at all required for understanding. Nevertheless, the general efficient algorithm of Lovász [60] and Marx [62] (see also Theorem 5 below) implies the following lemma.

Lemma 2. *Let $\mathcal{A}$ be a family of sets of size p each and let $q \in \mathbb{N}$. In time polynomial in $\binom{p+q}{p} + |\mathcal{A}|$ one can compute a q-representative subset $\mathcal{A}' \subseteq \mathcal{A}$ of size at most $\binom{p+q}{p}$.*

While the guaranteed size bound of $\binom{p+q}{p}$ might seem somewhat arbitrary at first, it is in fact tight: Consider the family $\mathcal{A}$ containing all $\binom{p+q}{p}$ subsets of size p of the set $\{1, \ldots, p+q\}$. Then, going over all sets B that are size q subsets of $\{1, \ldots, p+q\}$, we always find a unique set $A \in \mathcal{A}$ that is disjoint from B, namely $A = \{1, \ldots, p+q\} \setminus B$. Thus, all sets in $\mathcal{A}$ must be included and the lemma is tight. We will later make more use of the implicit observation that sets A that are unique "partners" for some set B must be included in any q-representative subset.

Let us now see that even this simple form of using representative sets, i.e., without the full power of specialized matroids, already suffices to reproduce "classic" kernelization results. We begin with the d-Hitting Set(k) problem, defined as follows.

> d-Hitting Set(k)
> **Input:** A universe U, a family $\mathcal{A}$ of subsets of U each of size at most d, and $k \in \mathbb{N}$.
> **Parameter:** k.
> **Question:** Is there a set of at most k elements of U that intersects all sets in $\mathcal{A}$?

We sketch a kernelization; let an instance $(U, \mathcal{A}, k)$ be given. Using Lemma 2 with $p = d$ and $q = k$ compute a k-representative subset $\mathcal{A}' \subseteq \mathcal{A}$ of size at

most $\binom{k+d}{d} \in O(k^d)$. If $(U, \mathcal{A}, k)$ is YES then also $(U, \mathcal{A}', k)$ must be YES since $\mathcal{A}' \subseteq \mathcal{A}$. If, however, $(U, \mathcal{A}, k)$ is NO then, in particular, no set $B \subseteq U$ of size at most k can be a solution for $(U, \mathcal{A}, k)$. In other words, for each such set B there is at least one set $A \in \mathcal{A}$ that avoids B, i.e., $A \cap B = \emptyset$. Since $\mathcal{A}'$ is k-representative for $\mathcal{A}$, for each choice of B we also find a set $A' \in \mathcal{A}'$ with $A' \cap B = \emptyset$, implying that $(U, \mathcal{A}', k)$ is NO, too.

We remark that the reduction to $|\mathcal{A}'| \in O(k^d)$ allows an encoding in $O(k^d \log d)$ bits, which is essentially optimal due to the mentioned result of Dell and van Melkebeek [22] that rules out efficient reduction to bit size $O(k^{d-\varepsilon})$ unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. It is possible, however, to improve the size of the ground set to $O(k^{d-1})$, rather than the implicit $O(d \cdot k^d) = O(k^d)$, using the kernelization of Abu-Khzam [2]. (It is an interesting problem to close the wide gap between this result and the trivial lower bound of $\Omega(k)$ for the ground set size.)

Let us now consider d-Set Packing(k) where the argument is slightly more involved, though certainly comparable to the less obvious application of the Sunflower Lemma as compared to d-Hitting Set(k) (cf. [21]).

> d-Set Packing(k)
> **Input:** A universe U, a family $\mathcal{A}$ of subsets of U each of size at most d, and $k \in \mathbb{N}$.
> **Parameter:** k.
> **Question:** Is there a selection of k sets in $\mathcal{A}$ that are pairwise disjoint?

Again, representative sets can be used to obtain a polynomial kernelization whose size is essentially optimal. This time, given an instance $(U, \mathcal{A}, k)$ of d-Set Packing(k) we compute a $d(k-1)$-representative subset $\mathcal{A}'$ of $\mathcal{A}$. Let us see that this works correctly. Clearly, if $(U, \mathcal{A}, k)$ was NO in the first place then the obtained instance $(U, \mathcal{A}', k)$ will be NO too. Assume now that $(U, \mathcal{A}, k)$ is YES. Let $A_1, \ldots, A_k \in \mathcal{A}$ be a selection of k pairwise disjoint sets such that as many sets A_i as possible are also contained in $\mathcal{A}'$. If $A_1, \ldots, A_k \in \mathcal{A}'$ then we are done, so assume w.l.o.g. that $A_1 \notin \mathcal{A}'$. Then, letting $B := A_2 \cup \ldots \cup A_k$ we note that $A_1 \cap B = \emptyset$ and that $|B| \leq d(k-1)$. It follows, since $\mathcal{A}'$ is $d(k-1)$-representative for $\mathcal{A}$, that there exists $A_1' \in \mathcal{A}'$ with $A_1' \cap B = \emptyset$. Then, however, we immediately see that $A_1' \in \mathcal{A}$ and $A_1', A_2, \ldots, A_k$ is also a selection of k pairwise disjoint sets but with more sets also contained in $\mathcal{A}'$; a contradiction. Thus, we must have $A_1, \ldots, A_k \in \mathcal{A}'$, and, therefore, the obtained instance $(U, \mathcal{A}', k)$ is indeed equivalent to $(U, \mathcal{A}, k)$.

Representative sets for matroids. We will now introduce representative sets for families of independent sets of a given matroid. Since all further known kernelizations via representative sets [57] make use of a particular type of matroid called

gammoid we will mainly focus on those. Let us recall that a matroid $M = (U, \mathcal{I})$ consists of a finite set U and a family $\mathcal{I}$ of subsets of U, called *independent sets*, fulfilling the following properties:

1. $\emptyset \in \mathcal{I}$.
2. If $X \subseteq Y$ and $Y \in \mathcal{I}$ then also $X \in \mathcal{I}$.
3. If $X, Y \in \mathcal{I}$ with $|X| < |Y|$ then there exists $y \in Y \setminus X$ such that $X \cup \{y\} \in \mathcal{I}$.

We can now give the full definition of q-representative sets for families of independent sets in a matroid. For ease of writing, let us say that an *independent set A extends an independent set B* if $A \cap B = \emptyset$ and $A \cup B$ is independent. Note that independence of $A \cup B$ requires independence of both A and B due to the second matroid property.

Definition 6 (q-representativeness for families of independent sets)**.** Let $M = (U, \mathcal{I})$ be a matroid. Let $\mathcal{A} \subseteq \mathcal{I}$ be a collection of independent sets of M and let $q \in \mathbb{N}$. We call a set $\mathcal{A}' \subseteq \mathcal{A}$ *q-representative for $\mathcal{A}$* if for every independent set B of size at most q there is an $A \in \mathcal{A}$ that extends B if and only if there is also an $A' \in \mathcal{A}'$ that extends B.

It should not come as a surprise that with the addition of matroid independence this opens up a much bigger world of applications. The, so far, most interesting matroids regarding kernelization applications are the gammoids (defined below). Their independence notion is strongly related to Menger's Theorem, and the proof that they are indeed matroids is due to Perfect [64].

Let $G = (V, E)$ be a graph that may have both directed and undirected edges, and let $S \subseteq V$. Say that a set $T \subseteq V$ is *linked to S* if there exist $|T|$ vertex-disjoint paths from S to T, i.e., each vertex in T is endpoint of a different path from S. Then the set system $M = (V, \mathcal{I})$ where $\mathcal{I}$ contains all sets T that are linked to S is a matroid. We say that M is the *gammoid on G with sources S*. (We note that often the roles of S and T are switched, which makes no difference regarding what matroids are gammoids. Furthermore, restricting $\mathcal{I}$ to any subset $V' \subseteq V$ still yields a gammoid, and the case of $V' = V$ is also called a *strict gammoid*.)

It is known that every gammoid can be represented as the (linear) independence of column vectors of a matrix, making them *linear matroids* (cf. [62]). The construction of the matrix over an appropriately large field can be made constructive by an efficient, randomized algorithm but it is a big open problem whether a deterministic construction exists. For simplicity, we hide these details in the following theorem, noting that the general version [60, 62] holds for any linear matroid when given the matrix representation (and without further use of randomization).

Theorem 5 (simplified version of result by Lovász [60] and Marx [62])**.** *Let M be a gammoid and let $\mathcal{A} = \{A_1, \ldots, A_m\}$ be a collection of independent sets, each of size p. We can find in randomized polynomial time a set $\mathcal{A}' \subseteq \mathcal{A}$ of size at most $\binom{p+q}{p}$ that is q-representative for $\mathcal{A}$.*

A highly useful property of representative sets is that they can be employed for actually finding particular objects (e.g., vertices) rather than just "blindly" discarding sets (or other objects) as we did for d-Hitting Set(k) and d-Set Packing(k). For $\mathcal{A}'$ to be q-representative for $\mathcal{A}$ it is required that *every set* B that can be extended by some $A \in \mathcal{A}$ can also be extended by some $A' \in \mathcal{A}'$. This entails, however, that if a given $A \in \mathcal{A}$ is *unique in extending some given* B then this enforces that $A \in \mathcal{A}'$; else, no set in $\mathcal{A}'$ could extend B. We will return to this trick soon.

Example application. Let us now discuss an application of Theorem 5, namely a polynomial kernelization for the following variant of Multiway Cut(k), called Deletable Terminal Multiway Cut(k):

> Deletable Terminal Multiway Cut(k)
> **Input:** A graph $G = (V, E)$, a set of terminals $S \subseteq V$, and $k \in \mathbb{N}$.
> **Parameter:** k.
> **Question:** Is there a set X of at most k vertices such that in $G - X$ no two terminals $t_1, t_2 \in S \setminus X$ are in the same connected component?

The problem can be easily seen to be NP-hard, since using terminal set $S = V$ requires finding a vertex cover of size at most k. Note also, that all instances with $|S| \leq k + 1$ are trivial since this would allow deletion of all but one terminal. Finally, unlike Multiway Cut, which is hard already for three terminals, for any *fixed size* of S we have a trivial solution if $k \geq |S| - 1$ or else can enumerate and test all $O(|V|^k) \subseteq O(|V|^{|S|-1})$ solution candidates in polynomial time.

The kernelization proceeds as follows: (1) We show that if an instance is yes then there is always a solution X that allows a certain path packing from S to X. (2) We set up a gammoid based on a graph G' derived from G, and with sources S. (3) We use Theorem 5 to find a superset of X of size $O(k^3)$, using the path packing to distinguish vertices in V. (4) We briefly explain how to use this superset to shrink the input graph G to $O(k^3)$ vertices.

Analyzing solutions. Let an instance (G, S, k) of Deletable Terminal Multiway Cut(k) be given. Assume that the instance is yes and, for analysis, let X denote a solution for (G, S, k) that contains the maximum number of terminals from S (among solutions of size at most k). Clearly, vertices in $X \cap S$ correspond to outright deletions of terminals, whereas $X_0 := X \setminus S$ separates the remaining

terminals $S_0 := S \setminus X$ from one another. We want to establish that X_0 is linked to S_0 in a strong sense, by using Hall's theorem.

Note that each connected component of $G - X$ contains at most one terminal from S; for brevity, we will call C containing a terminal from $S_0 = S \setminus X$ a *terminal component*. Let us say that a vertex $x \in X_0$ *sees a terminal component* C if in G the vertex x is adjacent to a vertex of C. We extend this to sets $Y \subseteq X_0$ by saying that Y sees a terminal component C if at least one $x \in Y$ sees C. Intuitively, if a vertex of X_0 sees some terminal components, then "putting that vertex back" into $G - X$ reconnects those components and terminals; ditto for $Y \subseteq X_0$.

We set up for using Hall's Theorem: Assume that any nonempty set $Y \subseteq X_0$ sees at most $|Y| + 1$ terminal components. It follows that in $G - (X \setminus Y)$ the set Y together with these terminal components (and possibly terminal-free components) forms a larger component with up to $|Y| + 1$ terminals. All other terminal components not seen by Y are unaffected. Observe that this allows an alternative solution by deleting any $|Y|$ of the $|Y|+1$ terminals, say a set $Y' \subseteq S_0$. This, however, contradicts our choice of X since $(X \setminus Y) \cup Y'$ would be a solution with larger intersection with S. Thus, every $Y \subseteq X_0$ sees at least $|Y| + 2$ terminal components C.

Using Hall's Theorem it can now be checked that we can find a matching of $|X_0| + 2$ terminal components to vertices in X_0 such that:

- Each component is matched to a vertex $x \in X_0$ that sees it.
- For any fixed vertex $x \in X_0$ we get three components matched to x.

Now, we "trade" matched components for disjoint paths from S_0 to X_0: Notice that in each component with a terminal t that is seen by some $x \in X_0$ we can freely choose a path from t to x with all vertices but x contained in the component. Thus, for all $|X_0|+2$ components we can find disjoint paths to the matched vertices in X_0. Hence, we get a path packing with $|X_0| + 2$ paths from S_0 to X_0 with three paths ending in any chosen vertex $x \in X_0$.

Setting up the gammoid. For the gammoid M we use a graph G' that is obtained from $G = (V, E)$ by adding two so-called *sink-only copies* v', v'' for each vertex $v \in V$. A sink-only copy v' (or v'') for v shares all in-neighbors with v but has no out-neighbors (i.e., if $\{u, v\}$ is an edge then we only add a directed edge (u, v')). Thus, adding such vertices does not affect, e.g., the existence of paths between any terminals, since they can only act as endpoints (sinks) of paths. Using the sink-only copies, we can formalize the informal statement of three paths ending in any $x \in X_0$ to three paths ending in $\{x, x', x''\}$. Let us also point out that the gammoid setting allows trivial paths consisting of just one vertex, e.g., we have such paths from $S \cap X$ to $S \cap X$. Overall, together with the above path packing we get that in G' there must exist a path packing of $|X|+2$ paths from S to $X \cup \{x', x''\}$ for every choice of $x \in X_0$.

Applying representative sets. Now we will apply the idea that representative sets can be used to identify particular objects. We will use Theorem 5 to compute a $k-1$ representative subset $\mathcal{T}'$ of $\mathcal{T}$ where $\mathcal{T} := \{\{v, v', v''\} \mid v \in V\}$. Our goal is to show that for all $x \in X_0$ we must have $\{x, x', x''\} \in \mathcal{T}'$. Note that the theorem guarantees $|\mathcal{T}'| \in O(k^3)$.

Our argument now depends crucially on the trick that we outlined previously: If there exists an independent set I of M of size/rank at most $k-1$ such that $\{x, x', x''\}$ uniquely extends I then this directly implies that $\{x, x', x''\}$ is contained in every $k-1$-representative subset $\mathcal{T}'$ of $\mathcal{T}$. Recall that we already know that $X \cup \{x', x''\}$ is linked to S in G' and thus it is independent, for all $x \in X_0$. It follows directly that $\{x, x', x''\}$ extends the independent set $X - x$ for all $x \in X_0$. It remains to prove that no other set $\{v, v', v''\} \in \mathcal{T}$ extends $X - x$.

Consider first any $v \in X - x$. In this case we have $\{v, v', v''\} \cap (X - x) = \{v\} \neq \emptyset$, implying that the set $\{v, v', v''\}$ does not extend $X - x$. The more interesting case is for $\{v, v', v''\}$ with $v \in V \setminus X$. First, note that for $\{v, v', v''\}$ to extend $X - x$ requires for $(X - x) \cup \{v, v', v''\}$ to be linked to S in G'. A (weaker) requirement is that $\{v, v', v''\}$ is linked to S in $G' - (X - x)$, since any paths from S to $X - x$ definitely block at least $X - x$ from being used in paths from S to $\{v, v', v''\}$.

Let us see that there cannot be three disjoint paths from S to $\{v, v', v''\}$ in $G' - (X - x)$: Recall that paths cannot have sink-only copies as interior vertices, so apart from v' and v'' we can use that X is a solution in graph G. At most one of the paths can come from a terminal in the terminal component of v, and one more path can include the vertex x. No third path is possible. Thus, we find that no other set $\{v, v', v''\}$ can extend $X - x$.

Since for each $x \in X_0$ the set $\{x, x', x''\}$ uniquely extends $X - x$ we get that for all vertices $x \in X_0$ we must have $\{x, x', x''\} \in \mathcal{T}'$. Hence, letting $V(\mathcal{T}')$ stand for $\{v \mid \{v, v', v''\} \in \mathcal{T}'\}$, it is guaranteed that $X_0 \subseteq V(\mathcal{T}')$. In extension this implies $X = X_0 \cup (X \cap S) \subseteq V(\mathcal{T}') \cup S$. There is a reduction rule that ensures $|S| = O(k)$ (see [39]), but let us omit this detail and directly assume that we have a set of $O(k^3)$ vertices containing all terminals S as well as at least one solution X (if one exists).

Shrinking the input graph to $O(k^3)$ vertices. We can now complete the kernelization. Let W denote the established set of $O(k^3)$ vertices that is guaranteed to completely contain at least one solution (as well as all terminals). Using this guarantee, there is no harm in making all vertices of $V \setminus W$ *undeletable*: For any vertex $v \in V \setminus W$ simply make the neighbors of v a clique and remove v from the graph; this captures the intention that deleting v does not remove any connectivity while also shrinking the graph. (Note that doing this for all vertices of $V \setminus W$ at once corresponds to the so-called *torso* operation applied to W.) We obtain an equivalent instance $(\hat{G}, S, k)$ where $\hat{G}$ is a graph on vertex set W of size at most $O(k^3)$.

Further results kernelization results based on matroids. Prior to the application of representative sets for kernelization [57], the fact that gammoids admit an efficient representation as column matroids of matrices over (sufficiently large) finite fields (cf. [62]) was used to find a (randomized) polynomial kernelization for Odd Cycle Transversal(k) [56], settling a well-known problem in kernelization. At high level, a represented gammoid is used to fairly succinctly encode a family of two-way cut queries that are sufficient to determine the status of the input instance. In the follow-up work [57] representative set tools were used, amongst others, to obtain somewhat more combinatorial[7] kernel results based on irrelevant vertex arguments.

Theorem 6 ([57])**.** *The following kernelizations are possible:* Almost 2-SAT(k)*, with $\mathcal{O}(k^6)$ variables;* s-Multiway Cut(k)*, with $\mathcal{O}(k^{s+1})$ vertices;* s-Multicut(k)*, with $\mathcal{O}(k^{\lceil\sqrt{2}s\rceil+1})$ vertices;* Group Feedback Vertex Set(k)*, for a group of s elements, with $\mathcal{O}(k^{2s+2})$ vertices. All results are randomized, with failure probability exponentially small in n.*

Note that, Almost 2-SAT(k), i.e., the task of making a 2-CNF formula satisfiable by deleting at most k variables, is a pivotal problem since several other problems have PPTs to it, e.g., e.g., Vertex Cover Above Matching, Vertex Cover Above LP, and RHorn-Backdoor Deletion Set. It also directly generalizes Odd Cycle Transversal(k). All these problems have polynomial kernelizations due to this connection.

Furthermore, the techniques were also used to obtain results called *cut covering sets*, which guarantee to include an optimal cut for each one of a (possibly exponentially large) set of cut queries. We recall the statement for the two-way cut setting and direct the reader to [57] for an s-multiway cut variant of the theorem.

Theorem 7 ([57])**.** *Let $G = (V, E)$ be a digraph and let $S, T \subseteq V$. Let r denote the size of a minimum (S, T)-vertex cut (which may intersect S and T). There exists a set $Z \subseteq V$, $|Z| = \mathcal{O}(|S| \cdot |T| \cdot r)$, such that for any $A \subseteq S$ and $B \subseteq T$, it holds that Z contains a minimum (A, B)-vertex cut. We can find such a set in randomized polynomial time with failure probability $\mathcal{O}(2^{-n})$.*

Further reading. The already mentioned recent paper of Fomin et al. [31] is a recommended follow-up read. Fomin et al. obtain faster algorithms for finding representative sets for linear matroids and for the special case of uniform matroids; in particular the second does not require a matrix representation. Furthermore, they explain several algorithmic applications and obtain, amongst others, the so far fastest deterministic algorithm for Path(k), running in time $\mathcal{O}(2.851^k m \log^2 n)$.

[7] The underlying result of Lovász [60] is proved via exterior algebra, and derived algorithms [62, 31] still use linear algebra tools.

6 Turing kernelization

Already before the kernelization lower bound framework [9] several authors had suggested the possibility of preprocessing into many independent small instances rather than just one [29, 40]. After the framework appeared, it was noted that the obtained lower bounds do not apply to this relaxed form of kernelization, which makes it a possible option for avoiding lower bounds.

A Turing kernel for Leaf Out-Tree(k). A first example was soon discovered by Binkele-Raible et al. [6]: Say that an *out-tree* is any directed tree with a unique vertex of in-degree zero, called the *root*, and with vertices of out-degree zero called the *leaves*. The Leaf Out-Tree(k) problem asks whether a given digraph $D = (V, A)$ contains an out-tree with at least k leaves. Binkele-Raible et al. [6] showed that this problem admits no polynomial kernelization unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ (using the then new framework of Bodlaender et al. [9]). In contrast, they proved that a variant called Rooted Leaf Out-Tree(k), where in addition to $D = (V, A)$ and k we are given a fixed vertex $v \in V$ to use as the root of the out-tree, does admit a kernelization to $\mathcal{O}(k^3)$ vertices (and, hence, polynomial total size). They concluded that, since a given instance $(D = (V, A), k)$ of Leaf Out-Tree(k) has only $|V|$ choices for a root v, one may preprocess the instance by returning $|V|$ instances (D, v, k) of Rooted Leaf Out-Tree(k), one for each choice of $v \in V$. Since the latter admits a polynomial kernelization, this yields $|V|$ instances on $\mathcal{O}(k^3)$ vertices each. Furthermore, (D, k) is yes for Leaf Out-Tree(k) if and only if at least one instance (D, v, k) is yes for Rooted Leaf Out-Tree(k). Altogether, the reduction of one instance of Leaf Out-Tree(k) to $|V|$ instances of Rooted Leaf Out-Tree(k) combined with a polynomial kernelization for the latter gave the first example[8] of what is now called a (polynomial) Turing kernelization. More specifically, it is a polynomial disjunctive kernelization since the status of the input instance is equivalent to the disjunction (or) of the outcomes of the $|V|$ reduced instances.

Turing kernelization and other variants. Given the success of the lower bound framework and the wealth of obtained results, a notion of preprocessing that avoids these lower bounds is of course highly interesting. Note that, from a practical perspective, a sequence of small, independent instances might also be easier to handle (e.g., by parallelization) than a single large instance. This aspect applies of course only to the case that the reduced instances are created in parallel, rather than adaptively. Theoretically, also an adaptive creation of inputs is interesting;

[8] Binkele-Raible et al. [6] also proved analogous results for Rooted Leaf Out-Branching(k) and Leaf Out-Branching(k) where the out-tree is required to span the input graph D.

in particular, lower bounds against adaptive (i.e., Turing) kernelization would be very powerful. Note that this necessitates a slightly more involved definition, since the "kernelization" needs to know the answers to already created instances before outputting the next one. It is thus natural to formalize a Turing kernelization for $Q \subseteq \Sigma^* \times \mathbb{N}$ as an efficient algorithm that given $(x, k) \in \Sigma^* \times \mathbb{N}$ *correctly decides whether* $(x, k) \in Q$ provided that it gets the answers to all (adaptively) created small instances. The traditional way in computer science to formalize this is by means of an oracle; we recall the definition given by Binkele-Raible et al. [6].

Definition 7 ([6])**.** A *t-oracle* for a parameterized problem Q is an oracle that takes as input (x, k) with $|x|, k \leq t$ and decides whether $(x, k) \in Q$ in constant time.

Definition 8 ([6])**.** A parameterized problem Q is said to have a $g(k)$*-sized Turing kernelization* if there is an algorithm which given an input (x, k) together with a $g(k)$-oracle for Q decides whether $(x, k) \in Q$ in time polynomial in $|x| + k$.

Naturally, by letting the oracle queries be to any other parameterized problem Q' or to any (classical) language L we could define variants such as generalized Turing kernelization or Turing compression. Note, however, that using Karp reductions we can easily translate oracle questions, which probably makes the distinction meaningless. In the following we will not insist on a concrete definition and simply allow the most relaxed variant of t-sized queries to any language L.

Let us informally state also the following restricted variants of Turing kernelization:

Disjunctive kernels: Like the example for Leaf Out-Tree(k), given an input (x, k), create $|x|^{O(1)}$ instances of size bounded in k such that (x, k) is yes if and only if *at least one output instance is* yes.

Conjunctive kernels: Given an input (x, k), create $|x|^{O(1)}$ instances of size bounded in k such that (x, k) is yes if and only if *all output instances are* yes. Surprisingly perhaps, we are already able to rule out polynomial conjunctive kernels for most problems with lower bounds against polynomial kernelization. We will recall this briefly later in this section.

Truth-table kernels: Generalizing conjunctive and disjunctive kernels one may simply define any Boolean function (or a family thereof, one for each arity) and demand that the input is yes if and only if the function applied to the outcomes for all output instances (treating yes as true and no as false) evaluates to true.

Initially, only few examples of polynomial Turing kernels were found for problems without polynomial kernels and all of them are in fact disjunctive kernels [6, 5, 67]. A few more simple examples have been observed throughout the

community. As an example, the reader is invited to consider the CLIQUE(Δ) problem where we seek a k-clique in a given graph G, parameterized by the maximum degree of G. It is not hard to give both an OR-(cross-)composition and a disjunctive polynomial kernelization.

Recently discovered Turing kernels. Last year, Thomassé et al. [69] found a polynomial Turing kernelization for INDEPENDENT SET on bull-free graphs[9], where the oracle questions are used in a dynamic programming fashion on a decomposition of the bull-free input graphs. In this case, the full power of Turing kernelizations as opposed to truth-table kernelization (or others) seems required. A similar form of Turing kernelization was independently found by Jansen [49] more recently for the PATH(k) problem restricted to planar graphs (and related cases). We describe a simplified version of the approach taken by Jansen [49], since this requires less preliminaries.

1. We are given a planar graph $G = (V, E)$ and an integer k, and want to find out whether G contains a simple path on at least k vertices. We will efficiently solve the instance by making a polynomial in $|V|$ number of oracle queries of size polynomial in k each.

2. We apply a tree-like decomposition of the graph into its three-connected components (attributed to Tutte). Any two incident components overlap in at most two vertices. Roughly, this can be obtained by recursing on vertex-separators of size at most two, until reaching a three-connected component.

3. Any three-connected component of a planar graph on at least $\Omega(k^c)$ vertices must contain a path of length at least k, for some known constant c (cf. [49]). Thus, if the graph has a three-connected component that has size $\Omega(k^c)$, then we can safely answer YES. Otherwise, and henceforth, all three-connected components have size $O(k^c)$.

4. If we take a leaf component then this is of size $O(k^c)$ and we can afford an oracle question for the longest path therein. If this returns a path of length at least k then we can answer YES and stop. Else, we ask for the longest paths ending in the component or passing through it. Concretely, if, e.g., the component has vertices p and q shared with its parent component, then we also perform oracle questions for (1) the longest p,q-path; (2) the maximum total length of two disjoint paths starting in p and q; (3) the longest path

[9]The so-called bull graph is obtained from a triangle by attaching a leaf each to two of its vertices. Bull-free graphs are exactly those graphs that contain no induced subgraph (on five vertices) that is isomorphic to the bull.

starting in p; (4) the longest path starting in p and avoiding q; (5) the longest path starting in q; (6) the longest path starting in q and avoiding p.

5. If the computation on a component does not lead to an immediate YES answer, then we encode the gained information from questions (1-6) using annotations in the parent component, delete the present component, and continue. Note that, in this simplified version, we tacitly used oracle questions for finding longest paths in some form of annotated graph. With a bit more work (cf. [49]), we can avoid annotations and employ self-reduction to find longest paths.

Jansen [49] also proved a polynomial Turing kernelization for CYCLE(k) on planar graphs, and generalized his ideas to work also on bounded degree graphs, claw-free graphs, and $K_{3,t}$ minor graphs (for both problems). Note also that all mentioned cases of PATH(k) and CYCLE(k) remain NP-hard and have trivial OR-(cross-)compositions by disjoint union that rule out polynomial kernels (cf. [49]). While the Tutte decomposition works on general graphs, it is crucial that the considered graph class has an inverse polynomial lower bound on the length of simple paths inside three-connected components (i.e., a component of size ℓ must be known to contain a path of length at least ℓ^{-c}).

Ruling out polynomial conjunctive kernels. Consider a polynomial conjunctive kernelization for a problem Q. On input (x,k) it will create $|x|^{O(1)}$ instances of size polynomial in k such that the input is YES if and only if all output instances are YES. (Note that, again, this will work just fine independently of whether the outputs are for Q, another problem Q', or any classical language L.) Let us modify the kernelization to arbitrarily (i.e., nondeterministically) output *only one of its created instances*. Clearly, if the input is YES then all outputs are YES and it returns any one of them. If the input is NO then at least one created output is NO. Thus, by nondeterministically selecting one output, it may falsely return a YES instance but at least one possible computation leads to the output of a NO instance. Generally, such kernelizations have been called *co-nondeterministic kernelizations* [53] for their similarity to Turing machines for coNP. (Note that those are in general more powerful because they are not restricted to "just" $|x|^{O(1)}$ instances but may in fact have $2^{|x|^{O(1)}}$ computation paths, each with different output.)

It has been observed[10] that the proof of Fortnow and Santhanam [35] for the OR-distillation conjecture applies also if the OR-distillation behaves, similarly to above, in a co-nondeterministic fashion. In the work of Dell and van Melkebeek [22] the so-called "complementary witness lemma" holds explicitly also for the co-nondeterministic setting. Long story short, both OR-(cross-)compositions

[10] This is attributed to Chen and Müller by Harnik and Naor [45].

and polynomial kernelizations/compressions may behave co-nondeterministically without any harm to the lower bound implications. Thus, any (possibly co-nondeterministic) OR-(cross-)composition rules out co-nondeterministic polynomial kernelizations and compressions; in particular, this rules out the more restricted case of polynomial conjunctive kernels for the problem in question [53]. (For more applications of co-nondeterminism we refer to [53, 55].)

Lower bounds for Turing kernels. Unlike for normal (many-one) kernelization, there is yet no technique for ruling out polynomial Turing kernels for any FPT problem (modulo any reasonable complexity hypothesis). The observation applied for polynomial conjunctive kernelizations should not be expected to generalize, in particular not to the seemingly powerful adaptive setting of Turing kernels. (Note that having any Turing kernelization again also implies fixed-parameter tractability, and thus W[1]-hardness rules out such kernels, assuming FPT $\neq$ W[1].)

Motivated by this state of the art, Hermelin et al. [46] initiated a completeness program centered around a newly introduced WK/MK-hierarchy of parameterized problems.[11] The starting point is the fact that results for polynomial kernelizations transfer, modulo technical details, by polynomial parameter transformations (see Bodlaender et al. [13]). If we relax to using generalized kernelizations or compressions then results transfer directly (see, e.g., Lemma 1). In the same way, this applies to the existence and non-existence of polynomial disjunctive, conjunctive, truth-table, and Turing kernelizations.

Arguably the most important class in [46] is WK[1]; it is the lowest hardness class in the hierarchy. Since a variety of problems were shown to be complete for WK[1] we will simply list some complete problems for WK[1], MK[2], and WK[2] below rather than giving formal definitions (and will not discuss further classes). At high level, all WK[i] and MK[i] classes are defined as closures of certain parameterized satisfiability-related problems under PPTs. These defining problems are reparameterizations of problems used to define the W[i] and M[i] classes from the parameterized hierarchy of intractability (see, e.g., [30]). Motivated by the variety of problems that could be classified as WK[1]-complete, Hermelin et al. [46] conjectured that no WK[1]-hard problem admits a polynomial Turing kernelization. Similarly to an efficient algorithm for any NP-hard problem (but maybe not as surprising) a polynomial Turing kernelization for any WK[1]-hard problem would be a breakthrough since none of the known hard problems (see below) seem

[11] The hierarchy is, in a sense, a reparameterization of the W[i]- and M[i]-hierarchies in parameterized intractability. It subsumes a strongly related hierarchy of Harnik and Naor [45] aimed at classical problems in relation to their witness size. A detailed discussion of the relation is given in Hermelin et al. [46].

particularly amenable to this (see also the discussion in [46]).

The HITTING SET problem (note the unrestricted set size) nicely showcases several levels of the hierarchy when taken under different parameterizations.

> HITTING SET
> **Input:** A universe U, a set family $\mathcal{F} \subseteq 2^U$, and $k \in \mathbb{N}$.
> **Question:** Is there a set of at most k elements of U that intersects every set in $\mathcal{F}$?

Under its standard parameter k the problem is complete for W[2] under parameterized reductions and, thus, not even FPT unless FPT = W[2]. Using, however, parameters $n := |U|$, $m := |\mathcal{F}|$, or $k \log n$ it can be easily seen to be FPT. Nevertheless, for all three parameters it is possible to rule out polynomial kernelizations; for the first two results this follows from work of Dom et al. [23]. Curiously, all three parameterizations give problems that are complete for different levels of the WK- and MK-hierarchies.

- HITTING SET(m) is complete for WK[1] and equivalent (also under PPTs) to problems such as CAPACITATED VERTEX COVER(k), CONNECTED VERTEX COVER(k), STEINER TREE($k + t$), MIN ONES d-SAT(k), CLIQUE($k \log n$), SET COVER(n), MULTICOLORED PATH(k), and BINARY NDTM HALTING(k). The latter problem asks whether a given nondeterministic Turing machine with binary alphabet stops within k steps.

  DISJOINT PATHS(k) and DISJOINT CYCLES(k) are WK[1]-hard.

- HITTING SET(n) is complete for MK[2] and equivalent to problems such as SET COVER(m) and CNF-SAT(n).

  Among hard problems for MK[2] there are, e.g., several structural parameterizations of DOMINATING SET(k).

- HITTING SET($k \log n$) is complete for WK[2] and equivalent to SET COVER($k \log m$), and DOMINATING SET($k \log n$).

We refer to Hermelin et al. [46] for a more extensive list of hard and complete problems, in particular also for MK[2] and WK[2]. The most interesting feature, perhaps, is the richness of complete problems for WK[1]. The fact that all these fairly different problems are equivalent for existence of polynomial Turing kernelizations supports the conjecture that no WK[1]-hard problem has such a kernelization. We also refer to Hermelin et al. [46] for a discussion of why these problems seem hard to Turing-kernelize.

A particular problem that has so far resisted a classification is PATH(k), for which neither a polynomial Turing kernelization nor WK[1]-hardness are known.

If we make the problem slightly richer by taking the input graph to be k-colored and asking for a k-path containing all k colors then it becomes $\mathsf{WK}[1]$-complete [46]; Jansen [49] extended this to the special case of planar inputs, motivated by his Turing kernelization for the un-colored version. Apart from this it would, obviously, be of high interest to have any complexity-theoretic evidence for the correctness of the conjecture that $\mathsf{WK}[1]$-hard problems have no polynomial Turing kernels.

7 Open problems

In this section we conclude the survey with some open problems. One of the central problems in kernelization research is certainly the understanding of possibilities and limitations of Turing kernelization. Furthermore, the Turing kernelization status of the Path(k) problem is of particular interest since it is not known to be hard for $\mathsf{WK}[1]$.

Open problem 1. Devise general upper and lower bound tools for Turing kernelization.

Open problem 2. Prove or disprove the conjecture that no $\mathsf{WK}[1]$-hard problem admits a polynomial Turing kernelization.

Open problem 3. Prove or disprove the existence of a polynomial Turing kernelization for Path(k).

The randomized polynomial kernelizations for, e.g., Deletable Terminal Multiway Cut(k) and Odd Cycle Transversal(k) [57], bring up the question of whether there are also deterministic polynomial kernels for these problems. This could be either by a derandomization of the existing approach or by completely new methods. Note that the exponentially small error in the kernelizations makes a lower bound against deterministic kernelizations unlikely (at least within the current framework).

Open problem 4. Are there deterministic polynomial kernelizations for the problems covered by the matroid-based kernelization results in [57]?

Finally, we mention (and recall) two concrete parameterized problems that have so far resisted classification into admitting or not admitting (e.g., modulo $\mathsf{NP} \not\subseteq \mathsf{coNP/poly}$) a polynomial kernelization.

Open problem 5. In the Multiway Cut(k) problem we are given an undirected graph $G = (V, E)$, a set of terminal vertices T, and $k \in \mathbb{N}$ with the task of deleting at most k non-terminal vertices to disconnect all terminals. Does this problem have a polynomial kernelization?

Recall that the restricted variant with only a fixed number s of terminals has a kernelization to an equivalent instance with $O(k^{s+1})$ vertices [57]. It is interesting whether the occurrence of s in the exponent is necessary and, if so, whether it is asymptotically optimal.

Open problem 6. In the Directed Feedback Vertex Set(k) problem we are given a directed graph $G = (V, A)$ and $k \in \mathbb{N}$ with the task to delete at most k vertices to make the graph acyclic (if possible). Does this problem have a polynomial kernelization?

This problem has survived, so far, the development of various upper and lower bound techniques, and is probably the longest-standing open problem in kernelization (and holding a solid place among established open problems in parameterized complexity overall).

Acknowledgements

The author is indebted to Andrew Drucker and Magnus Wahlström for several discussions and comments that greatly improved this survey. Furthermore, detailed comments of Bart Jansen and Somnath Sikdar on their work are gratefully acknowledged.

References

[1] Faisal N. Abu-Khzam. An improved kernelization algorithm for r-Set Packing. *Inf. Process. Lett.*, 110(16):621–624, 2010.

[2] Faisal N. Abu-Khzam. A kernelization algorithm for d-Hitting Set. *J. Comput. Syst. Sci.*, 76(7):524–531, 2010.

[3] Jochen Alber, Michael R. Fellows, and Rolf Niedermeier. Polynomial-time data reduction for dominating set. *J. ACM*, 51(3):363–384, 2004.

[4] Noga Alon, Gregory Gutin, Eun Jung Kim, Stefan Szeider, and Anders Yeo. Solving MAX-r-SAT above a tight lower bound. *Algorithmica*, 61(3):638–655, 2011.

[5] Abhimanyu M. Ambalath, Radheshyam Balasundaram, Chintan Rao H., Venkata Koppula, Neeldhara Misra, Geevarghese Philip, and M. S. Ramanujan. On the kernelization complexity of colorful motifs. In *IPEC*, volume 6478 of *LNCS*, pages 14–25. Springer, 2010.

[6] Daniel Binkele-Raible, Henning Fernau, Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Yngve Villanger. Kernel(s) for problems with no kernel: On out-trees with many leaves. *ACM Transactions on Algorithms*, 8(4):38, 2012.

[7] Hans L. Bodlaender. Kernelization: New upper and lower bound techniques. In *IWPEC*, volume 5917 of *LNCS*, pages 17–37. Springer, 2009.

[8] Hans L. Bodlaender, Rod Downey, Fedor V. Fomin, and Dániel Marx, editors. *The Multivariate Algorithmic Revolution and Beyond - Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday*, volume 7370 of *LNCS*. Springer, 2012.

[9] Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009.

[10] Hans L. Bodlaender, Fedor V. Fomin, Daniel Lokshtanov, Eelko Penninkx, Saket Saurabh, and Dimitrios M. Thilikos. (Meta) Kernelization. In *FOCS*, pages 629–638. IEEE Computer Society, 2009.

[11] Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Preprocessing for treewidth: A combinatorial analysis through kernelization. *SIAM J. Discrete Math.*, 27(4):2108–2142, 2013.

[12] Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Kernelization lower bounds by cross-composition. *SIAM J. Discrete Math.*, 28(1):277–305, 2014.

[13] Hans L. Bodlaender, Stéphan Thomassé, and Anders Yeo. Kernel bounds for disjoint cycles and disjoint paths. *Theor. Comput. Sci.*, 412(35):4570–4578, 2011.

[14] Jin-yi Cai, Venkatesan T. Chakaravarthy, Lane A. Hemaspaandra, and Mitsunori Ogihara. Competing provers yield improved Karp-Lipton collapse results. *Inf. Comput.*, 198(1):1–23, 2005.

[15] Jianer Chen, Iyad A. Kanj, and Weijia Jia. Vertex cover: Further observations and further improvements. *J. Algorithms*, 41(2):280–301, 2001.

[16] Robert Crowston, Michael R. Fellows, Gregory Gutin, Mark Jones, E. J. Kim, Fran Rosamond, Imre Z. Ruzsa, Stéphan Thomassé, and Anders Yeo. Satisfying more than half of a system of linear equations over GF(2): A multivariate approach. *J. Comput. Syst. Sci.*, 80(4):687–696, 2014.

[17] Robert Crowston, Gregory Gutin, Mark Jones, and Gabriele Muciaccia. Maximum balanced subgraph problem parameterized above lower bound. *Theor. Comput. Sci.*, 513:53–64, 2013.

[18] Marek Cygan, Fabrizio Grandoni, and Danny Hermelin. Tight kernel bounds for problems on graphs with small degeneracy - (extended abstract). In *ESA*, volume 8125 of *LNCS*, pages 361–372. Springer, 2013.

[19] Marek Cygan, Stefan Kratsch, Marcin Pilipczuk, Michal Pilipczuk, and Magnus Wahlström. Clique cover and graph separation: New incompressibility results. In *ICALP (1)*, volume 7391 of *LNCS*, pages 254–265. Springer, 2012.

[20] Holger Dell. A simple proof that AND-compression of NP-complete problems is hard. *Electronic Colloquium on Computational Complexity (ECCC)*, 2014. Available at `http://eccc.hpi-web.de/report/2014/075/`.

[21] Holger Dell and Dániel Marx. Kernelization of packing problems. In *SODA*, pages 68–81. SIAM, 2012.

[22] Holger Dell and Dieter van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. In *STOC*, pages 251–260. ACM, 2010.

[23] Michael Dom, Daniel Lokshtanov, and Saket Saurabh. Incompressibility through colors and IDs. In *ICALP (1)*, volume 5555 of *LNCS*, pages 378–389. Springer, 2009.

[24] Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity (Monographs in Computer Science)*. Springer, November 1998.

[25] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.

[26] Andrew Drucker. New limits to classical and quantum instance compression. In *FOCS*, pages 609–618. IEEE Computer Society, 2012.

[27] Andrew Drucker. New limits to classical and quantum instance compression. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:112, 2012.

[28] Paul Erdős and Richard Rado. Intersection theorems for systems of sets. *Journal of the London Mathematical Society*, 35:85–90, 1960.

[29] Vladimir Estivill-Castro, Michael R. Fellows, Michael A. Langston, and Frances A. Rosamond. FPT is P-time extremal structure I. In *ACiD*, volume 4 of *Texts in Algorithmics*, pages 1–41. King's College, London, 2005.

[30] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory (Texts in Theoretical Computer Science. An EATCS Series)*. Springer, March 2006.

[31] Fedor V. Fomin, Daniel Lokshtanov, and Saket Saurabh. Efficient computation of representative sets with applications in parameterized and exact algorithms. In *SODA*, pages 142–151. SIAM, 2014.

[32] Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Bidimensionality and kernels. In *SODA*, pages 503–510. SIAM, 2010.

[33] Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Linear kernels for (connected) dominating set on H-minor-free graphs. In *SODA*, pages 82–93. SIAM, 2012.

[34] Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Linear kernels for (connected) dominating set on graphs with excluded topological subgraphs. In *STACS*, volume 20 of *LIPIcs*, pages 92–103. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013.

[35] Lance Fortnow and Rahul Santhanam. Infeasibility of instance compression and succinct PCPs for NP. *J. Comput. Syst. Sci.*, 77(1):91–106, 2011.

[36] Jakub Gajarský, Petr Hlinený, Jan Obdrzálek, Sebastian Ordyniak, Felix Reidl, Peter Rossmanith, Fernando Sanchez Villaamil, and Somnath Sikdar. Kernelization using structural parameters on sparse graph classes. In *ESA*, volume 8125 of *LNCS*, pages 529–540. Springer, 2013.

[37] Robert Ganian, Friedrich Slivovsky, and Stefan Szeider. Meta-kernelization with structural parameters. In *MFCS*, volume 8087 of *LNCS*, pages 457–468. Springer, 2013.

[38] Valentin Garnero, Christophe Paul, Ignasi Sau, and Dimitrios M. Thilikos. Explicit linear kernels via dynamic programming. In *STACS*, volume 25 of *LIPIcs*, pages 312–324. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2014.

[39] Sylvain Guillemot. FPT algorithms for path-transversal and cycle-transversal problems. *Discrete Optimization*, 8(1):61–71, 2011.

[40] Jiong Guo and Rolf Niedermeier. Invitation to data reduction and problem kernelization. *SIGACT News*, 38(1):31–45, 2007.

[41] Jiong Guo and Rolf Niedermeier. Linear problem kernels for NP-hard problems on planar graphs. In *ICALP*, volume 4596 of *LNCS*, pages 375–386. Springer, 2007.

[42] Venkatesan Guruswami and Rishi Saket. On the inapproximability of vertex cover on k-partite k-uniform hypergraphs. In *ICALP (1)*, volume 6198 of *LNCS*, pages 360–371. Springer, 2010.

[43] Gregory Gutin, Eun Jung Kim, Stefan Szeider, and Anders Yeo. A probabilistic approach to problems parameterized above or below tight bounds. *J. Comput. Syst. Sci.*, 77(2):422–429, 2011.

[44] Shai Gutner. Polynomial kernels and faster algorithms for the dominating set problem on graphs with an excluded minor. In *IWPEC*, volume 5917 of *LNCS*, pages 246–257. Springer, 2009.

[45] Danny Harnik and Moni Naor. On the compressibility of $\mathcal{NP}$ instances and cryptographic applications. *SIAM J. Comput.*, 39(5):1667–1713, 2010.

[46] Danny Hermelin, Stefan Kratsch, Karolina Soltys, Magnus Wahlström, and Xi Wu. A completeness theory for polynomial (Turing) kernelization. In *IPEC*, volume 8246 of *LNCS*, pages 202–215. Springer, 2013.

[47] Danny Hermelin and Xi Wu. Weak compositions and their applications to polynomial lower bounds for kernelization. In *SODA*, pages 104–113. SIAM, 2012.

[48] Bart M. P. Jansen. On sparsification for computing treewidth. In *IPEC*, volume 8246 of *LNCS*, pages 216–229. Springer, 2013.

[49] Bart M. P. Jansen. Turing kernelization for finding long paths and cycles in restricted graph classes. *CoRR*, abs/1402.4718, 2014.

[50] Bart M. P. Jansen and Hans L. Bodlaender. Vertex cover kernelization revisited - upper and lower bounds for a refined parameter. *Theory Comput. Syst.*, 53(2):263–299, 2013.

[51] Eun Jung Kim, Alexander Langer, Christophe Paul, Felix Reidl, Peter Rossmanith, Ignasi Sau, and Somnath Sikdar. Linear kernels and single-exponential algorithms via protrusion decompositions. In *ICALP (1)*, volume 7965 of *LNCS*, pages 613–624. Springer, 2013.

[52] Lukasz Kowalik, Marcin Pilipczuk, and Karol Suchan. Towards optimal kernel for connected vertex cover in planar graphs. *Discrete Applied Mathematics*, 161(7-8):1154–1161, 2013.

[53] Stefan Kratsch. Co-nondeterminism in compositions: a kernelization lower bound for a ramsey-type problem. In *SODA*, pages 114–122. SIAM, 2012.

[54] Stefan Kratsch, Geevarghese Philip, and Saurabh Ray. Point line cover: The easy kernel is essentially tight. In *SODA*, pages 1596–1606. SIAM, 2014.

[55] Stefan Kratsch, Marcin Pilipczuk, Ashutosh Rai, and Venkatesh Raman. Kernel lower bounds using co-nondeterminism: Finding induced hereditary subgraphs. In *SWAT*, volume 7357 of *LNCS*, pages 364–375. Springer, 2012.

[56] Stefan Kratsch and Magnus Wahlström. Compression via matroids: a randomized polynomial kernel for odd cycle transversal. In *SODA*, pages 94–103. SIAM, 2012.

[57] Stefan Kratsch and Magnus Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. In *FOCS*, pages 450–459. IEEE Computer Society, 2012.

[58] Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. Kernelization - preprocessing with a guarantee. In Bodlaender et al. [8], pages 129–161.

[59] Daniel Lokshtanov, Matthias Mnich, and Saket Saurabh. A linear kernel for a planar connected dominating set. *Theor. Comput. Sci.*, 412(23):2536–2543, 2011.

[60] László Lovász. Flats in matroids and geometric graphs. In *Proc. Sixth British Combinatorial Conf.*, Combinatorial Surveys, pages 45–86, 1977.

[61] Meena Mahajan and Venkatesh Raman. Parameterizing above guaranteed values: MaxSat and MaxCut. *J. Algorithms*, 31(2):335–354, 1999.

[62] Dániel Marx. A parameterized view on matroid optimization problems. *Theor. Comput. Sci.*, 410(44):4471–4479, 2009.

[63] Neeldhara Misra, Geevarghese Philip, Venkatesh Raman, and Saket Saurabh. The kernelization complexity of connected domination in graphs with (no) small cycles. *Algorithmica*, 68(2):504–530, 2014.

[64] Hazel Perfect. Applications of Menger's graph theorem. *J. Math. Anal. Appl.*, 22:96–111, 1968.

[65] Geevarghese Philip, Venkatesh Raman, and Somnath Sikdar. Polynomial kernels for dominating set in graphs of bounded degeneracy and beyond. *ACM Transactions on Algorithms*, 9(1):11, 2012.

[66] Willard Van Orman Quine. The problem of simplifying truth functions. *The American Mathematical Monthly*, 59(8):521–531, 1952.

[67] Alexander Schäfer, Christian Komusiewicz, Hannes Moser, and Rolf Niedermeier. Parameterized computational complexity of finding small-diameter subgraphs. *Optimization Letters*, 6(5):883–891, 2012.

[68] Robert Endre Tarjan and Anthony E. Trojanowski. Finding a maximum independent set. *SIAM J. Comput.*, 6(3):537–546, 1977.

[69] Stéphan Thomassé, Nicolas Trotignon, and Kristina Vuskovic. Parameterized algorithm for weighted independent set problem in bull-free graphs. *CoRR*, abs/1310.6205, 2013.

[70] Magnus Wahlström. Abusing the Tutte matrix: An algebraic instance compression for the K-set-cycle problem. In *STACS*, volume 20 of *LIPIcs*, pages 341–352. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013.

[71] Chee-Keng Yap. Some consequences of non-uniform conditions on uniform classes. *Theor. Comput. Sci.*, 26:287–300, 1983.

THE CONCURRENCY COLUMN

BY

NOBUKO YOSHIDA

Department of Computing
Imperial College London
180 Queen's Gate, London, SW7 2AZ
n.yoshida@imperial.ac.uk, http://mrg.doc.ic.ac.uk/

BEATCS no 113

Recreational Formal Methods: Designing Vacuum Cleaning Trajectories

Frits Vaandrager
Institute for Computing and Information Sciences
Radboud University Nijmegen
F.Vaandrager@cs.ru.nl

Freek Verbeek
Department of Computer Science
Open University of The Netherlands
Freek.Verbeek@ou.nl

Abstract

We study an example due to Wooldridge of a small robotic agent that will vacuum clean a room. The room is an $n \times n$ grid and at any point the robot can move forward one step or turn right 90 degrees. The problem is to find a deterministic strategy for the robot in which (1) its next action only depends on its current square and orientation (one of north, west, south, east), and (2) all squares are visited infinitely often. We use a model checker and a SAT solver to find such strategies, and a proof assistant to exhibit certain symmetries in the problem.

1 Introduction

In his textbook on multiagent systems, Wooldridge [6] describes an example of a small robotic agent that will clean up a room. Figure 1 illustrates the vacuum world in which this robot operates. It is assumed that the room is a 3×3 grid, and that the robot always starts in square $(0,0)$ facing north. The agent can suck up dirt, move forward to the next square, or turn right 90°. The goal is to traverse the room continuously searching for dirt and removing dirt. Wooldridge asks for the construction of a deterministic, memoryless strategy which, given the current square and orientation (one of north, west, south, east), and given whether the robot observes dirt, specifies the next action of the agent (one of suck, forward,

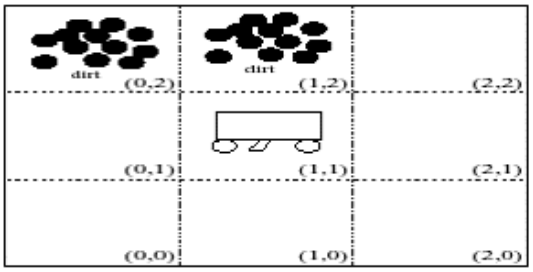


Figure 1: Vacuum world

turn). Assuming that all actions of the robot have their intended effect, this strategy should ensure that the robot will visit all squares infinitely often. Wooldridge gives a partial specification of such a strategy using a number of rules. The first rule states that if the agent is at location (x, y) and it perceives dirt, then the prescribed action is to suck up dirt.

$$In(x, y) \wedge Dirt(x, y) \quad \longrightarrow \quad Do(suck)$$

This rule takes priority over all other possible behaviors of the agent. Next four rules are listed which state that the robot will move from $(0, 0)$ to $(0, 1)$ to $(0, 2)$ and then to $(1, 2)$:

$$\begin{aligned} In(0,0) \wedge Facing(north) \wedge \neg Dirt(0,0) &\longrightarrow Do(forward) \\ In(0,1) \wedge Facing(north) \wedge \neg Dirt(0,1) &\longrightarrow Do(forward) \\ In(0,2) \wedge Facing(north) \wedge \neg Dirt(0,2) &\longrightarrow Do(turn) \\ In(0,2) \wedge Facing(east) &\longrightarrow Do(forward) \end{aligned}$$

According to Wooldridge, "similar rules can easily be generated that will get the agent to $(2, 2)$, and once at $(2, 2)$ back to $(0, 0)$." The first author, however, while diligently preparing a lecture on robotics for a freshman class, failed to find these rules. The problem is how to return to $(0, 0)$ after $(2, 2)$ has been reached. While on the way back, the robot may not revisit any square and orientation where it has been before: in such a case, since the robot is memoryless, it will continue forever on a loop that does not contain square $(0, 0)$. It appears that, after the robot has followed the initial rules specified by Wooldridge, it has painted itself in a corner and can never return to $(0, 0)$. It is not even obvious that there exists a deterministic, memoryless strategy for the robot that visits all squares infinitely often.

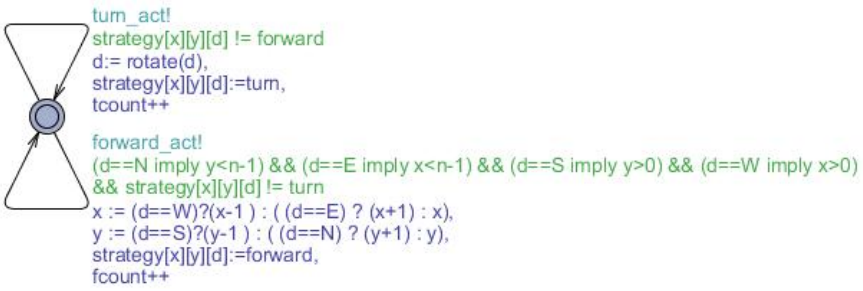


Figure 2: Uppaal model

This note describes how we tackled this problem using a model checker, a SAT solver, and even a proof assistant. The models and logical theories that we describe are available at the URL `http://www.mbsd.cs.ru.nl/publications/papers/fvaan/vacuumworld/`.

2 Model Checking

The problem of finding strategies for the vacuum cleaning robot can easily be encoded in a model checker. We constructed a model using the Uppaal tool [2]. Figure 2 displays the main template of our model. The model is parametrized by a constant `n`, which specifies the size of the grid. We use variables `x` and `y`, which range over type `pos` $= \{0, \ldots, n-1\}$, to store the current position of the robot, and a variable `d`, which ranges over type `dir` $= \{N, W, S, E\}$, to store the current orientation. Initially, `x` and `y` equal `0`, and `d` equals `N`. There are two transitions in the model, `turn_act!` and `forward_act!`. In the turn transition, the orientation `d` is updated using the function rotate, given by rotate(`N`) = `E`, rotate(`E`) = `S`, rotate(`S`) = `W` and rotate(`W`) = `N`. A forward transition is only enabled when there is a square in front of the robot, to prevent that the robot will hit the wall. In the model we abstract away from the dirt sucking as this is irrelevant for our problem.

An auxiliary array variable `strategy` records, for each position (`i`, `j`) and orientation `k`, the current strategy value, which is either `undefined`, `forward` or `turn`. Initially, `strategy[i][j][k]` is undefined for all `i`, `j` and `k`. Once `strategy[i][j][k]` is set to either `turn` or `forward`, it can never be changed again. We also use auxiliary variables `tcount` and `fcount` to count the total number of turns and forward moves, respectively.

Using the Uppaal verifier, we established that if the robot follows the rules specified by Wooldridge, it indeed paints itself in a corner. In fact, since the following Uppaal query does not hold for our model, there does not even exist a

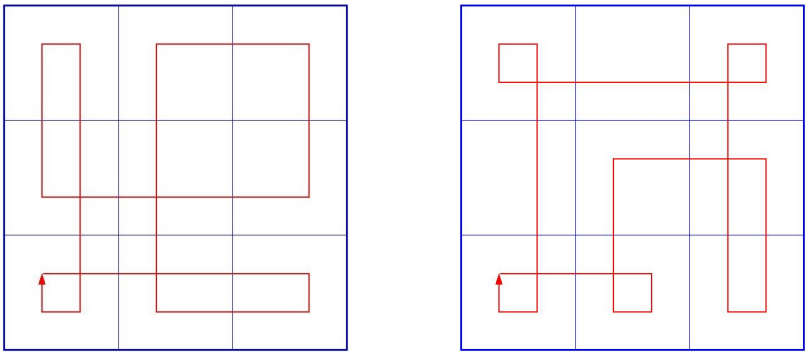


Figure 3: Two strategies with 12 (left) and 16 turns (right)

strategy that follows the rules for $(0,0)$ and $(0,1)$:

```
E<> (x==0 && y==0 && d==N &&
     forall (i:pos) forall (j:pos) visited(i,j) &&
     strategy[0][0][N]==forward && strategy[0][1][N]==forward)
```

Here `E <>` is Uppaal notation for the temporal operator $\exists\diamond$ and means "there exists a run leading to a state satisfying". Predicate `visited(i,j)` evaluates to true if the robot has visited square `(i,j)`, that is, `strategy[i][j][k]` is defined for some orientation `k`. By omitting the last two conjuncts in the above query, we can instruct the Uppaal verifier to search for strategies that visit all squares infinitely often. Figure 3 shows two strategies found by Uppaal. The strategy on the right was (independently) also discovered by Bart van Thiel, one of the students from the robotics class. The two strategies of Figure 3 differ since the left one makes 12 turns whereas the right one makes 16 turns. Clearly, the number of turns in any strategy must be a multiple of 4. Using Uppaal we found that in fact all strategies contain either 12, 16 or 20 turns. Figure 4 shows two strategies, found by Uppaal, which both make 20 turns. These strategies differ since the left one contains 12 forward moves whereas the right one has 14 forward moves. It is easy to see that the number of forward moves in any strategy must be an even number. Using Uppaal we found that all strategies contain either 10, 12 or 14 forward moves.

In theory it is easy to enumerate all strategies using Uppaal: one repeatedly asks Uppaal whether there exists a strategy that is different from all strategies found thus far. In practice, however, this is quite involved, requiring either manual entry of all strategies as part of queries, or a nontrivial script which transforms Uppaal traces into queries. In order to obtain a complete overview of all possible strategies we therefore found it convenient to use a different tool.

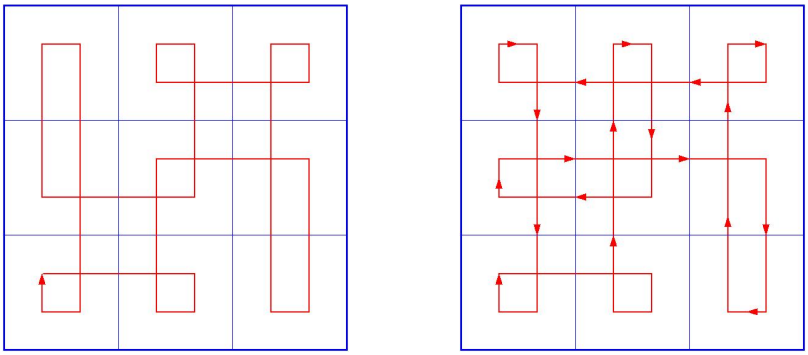


Figure 4: Two strategies with 20 turns and 12 (left) resp. 14 (right) forward moves

3 Constraint Solving

To enumerate vacuum cleaning strategies, we reformulate the problem: we need to find a path of length p that traverses a set of states Q and has to satisfy certain constraints. We then solve this problem for all p such that $l \leq p \leq h$ with l and h some conservatively estimated lower and higher bounds. The constraints can be formulated as a propositional satisfiability (SAT) problem. We therefore use zChaff [4], an automated solver for SAT problems.

For our SAT formulation of finding a vacuum cleaning strategy, we formalize the vacuum world as a labeled transition system.

Definition 3.1. *A* labeled transition system (LTS) *is a triple* $\mathcal{L} = (Q, A, \rightarrow)$*, where* Q *is a set of* states*,* A *is a set of* actions*, and* $\rightarrow \subseteq Q \times A \times Q$ *is a set of* transitions*. We write* $q \xrightarrow{a} q'$ *if* $(q, a, q') \in \rightarrow$*, and* $q \rightarrow q'$ *if there exists an* $a \in A$ *s.t.* $q \xrightarrow{a} q'$*.*

Fix a grid size n. Then our vacuum cleaning world is described by the LTS $\mathcal{V} = (\{0, 1, \ldots, n-1\} \times \{0, 1, \ldots, n-1\} \times \{\mathtt{N}, \mathtt{W}, \mathtt{S}, \mathtt{E}\}, \{\mathit{forward}, \mathit{turn}\}, \rightarrow)$, where relation $\rightarrow$ contains the following transitions, for $x, y \in [0, n-1]$ and $d \in \{\mathtt{N}, \mathtt{W}, \mathtt{S}, \mathtt{E}\}$,

$$
\begin{aligned}
(x, y, d) &\xrightarrow{\mathit{turn}} (x, y, \mathsf{rotate}(d)) \\
(x, y, \mathtt{N}) &\xrightarrow{\mathit{forward}} (x, y+1, \mathtt{N}) \quad \text{if} \quad y < n-1 \\
(x, y, \mathtt{E}) &\xrightarrow{\mathit{forward}} (x+1, y, \mathtt{E}) \quad \text{if} \quad x < n-1 \\
(x, y, \mathtt{S}) &\xrightarrow{\mathit{forward}} (x, y-1, \mathtt{S}) \quad \text{if} \quad y > 0 \\
(x, y, \mathtt{W}) &\xrightarrow{\mathit{forward}} (x-1, y, \mathtt{W}) \quad \text{if} \quad x > 0
\end{aligned}
$$

The SAT formulation of our problem introduces a Boolean variable for each pair (q, t), with q a state and t a natural number such that $0 \leq t < p$. We denote

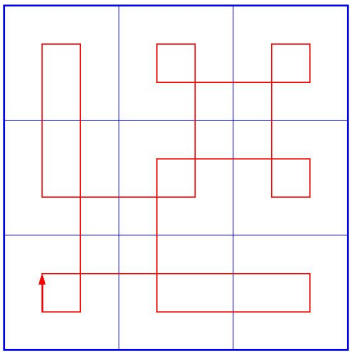


Figure 5: Another strategy with 20 turns and 12 forward moves found by zChaff

the Boolean variable corresponding to (q, t) with $\langle q, t\rangle$. The intended semantics is that Boolean variable $\langle q, t\rangle$ is true iff the vacuum cleaner is in state q at time slot t. All constraints are expressed in terms of these Boolean variables:
The strategy begins in $(0, 0, \mathtt{N})$:

$$\langle((0, 0, \mathtt{N}), 0\rangle$$

The strategy ends in $(0, 0, \mathtt{N})$:

$$\langle((0, 0, \mathtt{N}), p - 1\rangle$$

The strategy is connected:

$$\forall q, t < p - 1 \cdot \langle q, t\rangle \implies \bigvee_{\exists a \cdot q \xrightarrow{a} q'} \langle q', t + 1\rangle$$

The strategy is covering:

$$\forall x, y \cdot \exists d, t \cdot \langle(x, y, d), t\rangle$$

The strategy contains no duplicates other than the starting position:

$$\forall q, t_0, t_1 > t_0 \cdot \langle q, t_0\rangle \wedge q \neq (0, 0, \mathtt{N}) \implies \neg\langle q, t_1\rangle$$

Within one second, zChaff finds 28 solutions, ranging from the smallest (12 turns, 12 forward moves) to the most complex (20 turns, 14 forward moves). However, many solution are symmetric. There are four variants of the left strategy in Figure 3, which can be obtained by rotating the entire strategy 90, 180 and 270 degrees. Rotation and reflection variants of the right strategy of Figure 3 occur

eight times, variants of the left strategy of Figure 4 four times, and variants the right strategy of Figure 4 eight times. The only really new strategy found with zChaff, which has four incarnations, is displayed in Figure 5.

In order to obtain insight in these symmetries, we found it convenient to use a different tool.

4 Symmetries

In this section, we take a closer look at the symmetries that are present in the vacuum cleaning world. The proofs of all the theorems and lemmas in this section have been checked using the proof assistant Isabelle [5]. First we give a slightly more abstract characterization of the strategies Wooldridge asks for.

Definition 4.1. *Let $\mathcal{L} = (Q, A, \rightarrow)$ be an LTS. A* cycle *of $\mathcal{L}$ is a sequence $\sigma = q_1, \ldots, q_k$ of states such that, for all $i < k$, $q_i \rightarrow q_{i+1}$ and $q_k \rightarrow q_1$. A cycle is* minimal *if all states occurring in it are pairwise different.*

Definition 4.2. *Let Q be a a set of states, let σ be a sequence of states from Q, and let $\equiv$ be an equivalence relation on Q. We say that σ* covers *$\equiv$ if each equivalence class C of $\equiv$ contains a state that occurs in σ.*

Let $\approx$ be the equivalence relation that deems two states of the vacuum world LTS $\mathcal{V}$ equivalent if they belong to the same square on the grid:

$$(x, y, d) \approx (x', y', d') \quad \Leftrightarrow \quad x = x' \wedge y = y'.$$

Then the strategies Wooldridge asks for correspond to minimal cycles of the LTS $\mathcal{V}$ that cover $\approx$. Observe that the requirement that a strategy starts with $(0, 0, \mathtt{N})$ is not essential since any minimal cycle of $\mathcal{V}$ that covers $\approx$ contains $(0, 0, \mathtt{N})$, and any state on a cycle can be turned into the initial state by shifting states.

An automorphism is an isomorphism from an object to itself. It preserves the structure and captures a symmetry present in an object.

Definition 4.3. *An* automorphism *for an LTS $\mathcal{L} = (Q, A, \rightarrow)$ is a bijection $f : Q \rightarrow Q$ such that, for all $q, q' \in Q$ and for all $a \in A$, $q \xrightarrow{a} q'$ iff $f(q) \xrightarrow{a} f(q')$.*

The next theorem states that the function R that takes the whole vacuum world LTS and rotates it 90° to the right is an automorphism.

Theorem 4.4. *Let R be the function on states of $\mathcal{V}$ given by*

$$R(x, y, d) \quad = \quad (y, n - 1 - x, \mathsf{rotate}(d)).$$

Then R is an automorphism for vacuum world $\mathcal{V}$.

Theorem 4.5. *Let f be an automorphism for an LTS $\mathcal{L}$ and let σ be a cycle of $\mathcal{L}$. Then $f(\sigma)$ is a cycle of $\mathcal{L}$. Moreover, if σ is minimal then $f(\sigma)$ is also minimal.*

Lemma 4.6. *Let f be a bijection on a set of states Q, let $\equiv$ be an equivalence relation on Q such that $\forall q, q' \in Q : q \equiv q'$ implies $f(q) \equiv f(q')$ ($\equiv$ is a congruence for f), and let σ be a sequence of states in Q that covers $\equiv$. Then $f(\sigma)$ covers $\equiv$.*

Suppose σ is a minimal cycle of $\mathcal{V}$ that covers $\approx$. By Theorems 4.4 and 4.5, $R(\sigma)$ is a minimal cycle of $\mathcal{V}$. Since bijection R trivially is a congruence for $\approx$, it follows by Lemma 4.6 that $R(\sigma)$ covers $\approx$. Thus automorphism R maps strategies to strategies.

With the rotation automorphism R we capture most but not all the symmetries in our vacuum world. Besides rotation of a strategy, we also must consider the reflection of a strategy in the axis $x = 1/2n$. The mirror image of a strategy in which the robot only takes right turns, is a strategy in which the robot only takes left turns. However, in order to obtain a strategy with right turns again we can reverse the direction in which the edges are traversed. Mathematically, we need a notion of "autocontramorphism" to capture these symmetries.

Definition 4.7. *An* autocontramorphism *for an LTS $\mathcal{L} = (Q, A, \rightarrow)$ is a bijection $f : Q \rightarrow Q$ such that, for all $q, q' \in Q$ and for all $a \in A$, $q \xrightarrow{a} q'$ iff $f(q') \xrightarrow{a} f(q)$.*

Theorem 4.8. *Let F be the function on states of $\mathcal{V}$ given by*

$$F(x, y, d) \quad = \quad (n - 1 - x, y, \mathsf{flip}(d)),$$

where function flip *is defined by* $\mathsf{flip}(\mathtt{N}) = \mathtt{S}$, $\mathsf{flip}(\mathtt{E}) = \mathtt{E}$, $\mathsf{flip}(\mathtt{S}) = \mathtt{N}$, *and* $\mathsf{flip}(\mathtt{W}) = \mathtt{W}$, *Then F is an autocontramorphism for vacuum world $\mathcal{V}$.*

Theorem 4.9. *Let f be an autocontramorphism for an LTS $\mathcal{L}$ and let σ be a cycle of $\mathcal{L}$. Then $f(\sigma)$ is a cycle of $\mathcal{L}$. Moreover, if σ is minimal then $f(\sigma)$ is also minimal.*

Suppose σ is a minimal cycle of $\mathcal{V}$ that covers $\approx$. By Theorems 4.8 and 4.9, $F(\sigma)$ is a minimal cycle of $\mathcal{V}$. Since bijection F trivially is a congruence for $\approx$, it follows by Lemma 4.6 that $F(\sigma)$ covers $\approx$. Thus autocontramorphism F maps strategies to strategies.

Modulo the symmetries induced by rotation automorphism R and reflection autocontramorphism F, the 28 vacuum cleaning strategies found by zChaff reduce to the 5 strategies displayed in Figures 3, 4 and 5.

5 Snake Tilings

Now that we have a full understanding and classification of the vacuum cleaning strategies for a 3×3 grid, the natural question arises whether we can also solve this problem for arbitrary $m \times n$ grids, for $m, n \geq 1$. Model checking and SAT solving can only compute strategies in case m and n are small: Uppaal runs out of memory for a 5×5 grid, and the largest instance that we could solve using zChaff was a 7×7 grid.

We can at least prove the existence of vacuum cleaning strategies for arbitrary $m \times n$ grids using "tiles" with incoming and outgoing arrows. Figure 6 illustrates a tiling scheme that we may use to obtain a strategy for an arbitrary $m \times n$ grid, for m, n even and at least 4. The idea is that we can duplicate tile **B** $\frac{m-4}{2}$ times to obtain a bottom row of length m. The **B*****C** pattern can then be copied to the two rows above. Next the row of tile **D** can be copied $\frac{n-4}{2}$ times leading to a tiling of the $m \times n$ grid. Using similar tiling patterns one can prove the existence of vacuum cleaning strategies for arbitrary $m \times n$ grids.

The tilings of Figure 6 are closely related to the work of Kari [3] on infinite snake tiling problems, except that the trajectories ("snakes") of Kari may also turn left and do not have to return to their starting state. Similar tilings were also studied by Adleman et al. [1] in their work on self-assembly.

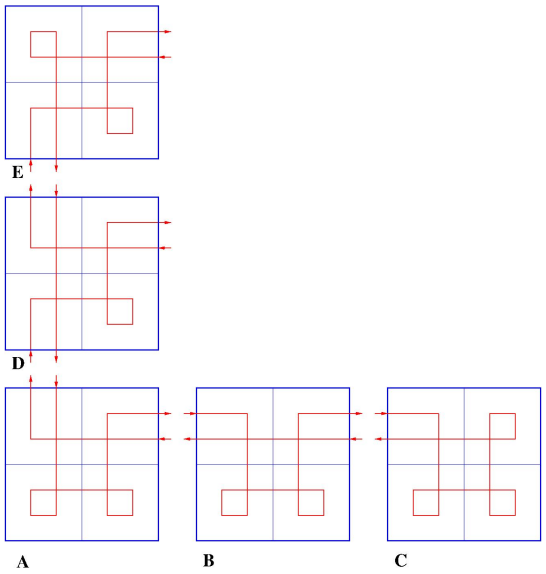


Figure 6: A tiling for $m \times n$ grids, with $m, n \geq 4$ even

6 Conclusion

The vacuum world example of Wooldridge [6] serves as a nice illustration of how the combined use of various tools and techniques from theoretical computer science may help to solve a problem.

We do not expect that this note will revolutionize the vacuum cleaning industry. After all, why would one impose the restriction that a vacuum cleaning robot may only turn right when electric motors just as easily run forward as backward? Why would one restrict to memoryless strategies when memory is so cheap and just adding a single bit to the domain of strategies makes it trivial to design schedules that visit each square infinitely often? Our strategies are also based on the unrealistic assumption that the floor is empty and without obstacles like tables and chairs that must be avoided.

The trajectories of Figures 3, 4 and 5 have some aesthetic quality and may serve as a basis for design of tilings, e.g. a long snake that bites itself in the tail.

The moral of our story is that authors of wonderful textbooks should be careful with the use of phrases like "similar rules can easily be generated". The risk is that colleagues will publish a note in the Bulletin of the EATCS pointing out that in fact the generation of such rules is impossible or at least tricky.

References

[1] Leonard M. Adleman, Jarkko Kari, Lila Kari, and Dustin Reishus. On the decidability of self-assembly of infinite ribbons. In *FOCS*, pages 530–537. IEEE Computer Society, 2002.

[2] Gerd Behrmann, Alexandre David, Kim G. Larsen, John Håkansson, Paul Pettersson, Wang Yi, and Martijn Hendriks. Uppaal 4.0. In *QEST 2006,* 11-14 September 2006, Riverside, CA, USA, pages 125–126. IEEE Computer Society, 2006.

[3] Jarkko Kari. Infinite snake tiling problems. In *DLT 2002, Kyoto, Japan, September 18-21, 2002, Revised Papers*, LNCS 2450, pages 67–77. Springer, 2002.

[4] Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik. Chaff: Engineering an efficient SAT solver. In DAC '01, pages 530–535, New York, NY, USA, 2001. ACM.

[5] Lawrence C. Paulson. Isabelle: The next 700 theorem provers. *CoRR*, cs.LO/9301106, 1993.

[6] Michael Wooldridge. *An Introduction to MultiAgent Systems, 2nd edition*. John Wiley & Sons Ltd, 2009.

The Distributed Computing Column

BY

Panagiota Fatourou

Department of Computer Science, University of Crete
P.O. Box 2208 GR-714 09 Heraklion, Crete, Greece
and
Institute of Computer Science (ICS)
Foundation for Research and Technology (FORTH)
N. Plastira 100. Vassilika Vouton
GR-700 13 Heraklion, Crete, Greece
faturu@csd.uoc.gr

BEATCS no 113

CONSISTENCY FOR TRANSACTIONAL MEMORY COMPUTING

Dmytro Dziuma
FORTH-ICS, Greece
dixond@acm.lviv.ua

Panagiota Fatourou*
FORTH-ICS & University of Crete, Greece
faturu@csd.uoc.gr

Eleni Kanellou
IRISA, Université de Rennes, France & FORTH-ICS, Greece
eleni.kanellou@irisa.fr

Abstract

This paper provides *formal definitions* for a comprehensive collection of consistency conditions for transactional memory (TM) computing. We express all conditions in a uniform way using a formal framework that we present.

For each of the conditions, we provide two versions: one that allows a transaction T to read the value of a data item written by another transaction T' that can be live and not yet commit-pending provided that T' will eventually commit, and a version which allows transactions to read values written only by transactions that have either commited before T starts or are commit-pending. Deriving the first versions was not an easy task but it has some benefits: (1) this version of each condition is weaker than the second one and so it results in a wider universe of algorithms which there is no reason to exclude from being considered correct, and (2) some definitions work, as is, for universal constructions contributing towards unifying the two models.

The formalism for the presented consistency conditions does not base on any unrealistic assumptions, such as that transactional operations are executed atomically or that write operations write distinct values for data items. Making such assumptions facilitates the task of formally expressing the consistency conditions significantly, but results in formal presentations of them that are unrealistic, i.e. that cannot be used to characterize the correctness of most of the executions produced by any reasonable TM algorithm.

*Currently with École Polytechnique Fédérale de Lausanne (EPFL), Switzerland, where she works as an EcoCloud visiting professor.

1 Introduction

Transactional memory (TM) [19, 27] is a promising programming paradigm that aims at simplifying parallel programming by using the notion of a transaction. A *transaction* is a piece of code containing accesses to pieces of data, known as *data items*, which are accessed simultaneously by several threads in a concurrent setting. A transaction may either *commit* and then its updates are effectuated or *abort* and then its updates are discarded. By using transactions, the naive programmer need only enhance its sequential code with invocations of special routines such as ReadDI and WriteDI (which we will call *transactional operations*) to indicate reads or writes for data items, respectively.

The TM algorithm provides a shared representation for each data item and implementations for ReadDI and WriteDI using the *base objects* supported by the system, so that all synchronization problems that may arise during the concurrent execution of transactional operations are addressed. When a transaction executes all its transactional operations it calls a routine called TryCommit in order to commit. TryCommit may return `TRUE` in which case the transaction commits or `FALSE` in which case the transaction aborts. We say that a transaction is *commit-pending* at some point in time if it has invoked TryCommit but it has not yet received a response. The implementation details of the TM algorithm are hidden from the naive programmer whose programming task is therefore highly simplified. TM has been given special attention in the last ten years with hundreds or even thousands of papers addressing different problems arising in TM computing (see e.g. [17, 16] for books addressing different aspects of TM computing).

One of the most fundamental problems of TM computing is *safety*. Most TM consistency conditions [3, 15, 16, 21, 12, 6, 7] originate from existing shared memory or database consistency models. However, in contrast to what happens in shared memory models where safety is defined in terms of read and write *operations* in memory, safety in TM computing is defined in terms of *transactions*, each of which may contain more than one read or write operations on data items. Comparing now to database transactions, the main difficulty when defining safety in TM computing is that transactional operations are executed by invoking ReadDI or WriteDI and therefore the execution of a transactional operation has duration and is usually overlapping with the execution of other transactional operations, whereas in database transactions read and write operations are considered to be atomic. For these reasons, existing safety definitions for these two settings (shared memory and database concurrent transactions) cannot be applied verbatim to TM algorithms. Formalizing safety definitions for TM computing requires more effort.

This article presents a comprehensive collection of consistency conditions for TM computing. All conditions are expressed in a uniform way using a formal framework that we present in Section 2. This article can therefore serve as a

survey of *consistency conditions* for TM computing. However, it aspires to be much more than this.

For all known TM consistency conditions we provide a new version, called *live*, in which a transaction T is allowed to read the value of a data item written by another transaction T' that can be live and not yet commit-pending provided that T' will eventually commit (or that T' will commit if T commits). Most TM consistency conditions [3, 6, 7, 15, 16, 21] presented thus far did not allow a transaction to read values that have been written by transactions that are neither committed nor commit-pending; we call this version of a consistency condition *commit-oriented*. The live version of a definition is weaker than its commit-oriented version, thus resulting in a wider universe of algorithms which should not be excluded from being considered correct. For instance, consider an algorithm which produces executions in which a transaction T is allowed to read a value for a data item x writtten by some transaction T' which has neither committed nor is commit-pending when T starts its execution. However, suppose that the algorithm has been designed in such a way that when this occurs, the algorithm ensures that T' will commit. Then, there is no reason for executions of the algorithm in which this behaviour is met not to be considered correct, i.e. such executions are correct. However, current consistency conditions, as they are formally expressed, exclude such executions from the set of executions they allow. The live version of a consistency condition we present here solves this problem.

A *universal construction* [18] is a mechanism for automatically executing pieces of sequential code in a concurrent environment. A universal construction supports a single operation PERFORM which takes as a parameter a pointer to a routine containing the piece of sequential code to execute concurrently and returns `TRUE` if this is done successfully. Similarly to TM, the sequential code must be enhanced so that accesses to data items are identified by calling routines READDI and WRITEDI. Aparently, universal constructions and TM algorithms are closely related since they both aim at simplifying parallel programming. There are however two basic differences between these two paradigms: (1) the application code must be programmed differently; specifically, in a universal construction, the piece of (the enhanced) sequential code must be included in a routine and a pointer to this routine must then be passed as a parameter to PERFORM, whereas in a TM setting, the code may contain direct invocations of READDI and WRITEDI, and (2) a TM algorithm allows the external environment to choose the action to be performed when a transaction aborts, whereas a call to PERFORM returns only when the simulated code has been successfully applied to the simulated state, i.e. after commit[1].

[1] We remark that the common behaviour for the external environment in a TM setting is to restart an aborted transaction until it eventually commits, so the difference is not essential.

A second benefit of the live versions of the consistency conditions presented here is that some of them work, as are, for universal constructions, by having a call to PERFORM to play the role of a transaction[2]. This contributes towards unifying the two models. It is remarkable that deriving the live version of consistency conditions was not an easy task so we consider their presentation as a significant contribution of this report.

For the derivation of the presented consistency conditions, we do not make any restrictive assumptions, such as that transactional operations are executed atomically or that write operations write distinct values for data items. Making such an assumption is unrealistically restrictive since all TM algorithms produce executions that do not satisfy these assumptions. Thus, a consistency condition that has been expressed making such an assumption cannot be used to characterize such executions, and thus fail to also characterize whether the TM algorithm itself satisfies the condition. We remark that making such assumptions significantly facilitates the task of formally expressing a consistency condition but the formal presentation that results is extremely restricitve since it cannot be used to characterize the correctness of most of the executions produced by any reasonable TM algorithm.

Related Work. Among the consistency conditions met in TM computing papers are the following: strict serializability [23], serializability [23], opacity [15, 16], virtual world consistency [21], TMS1 [12] (and TMS2 [12]), and snapshot isolation [2, 10, 25, 6, 7]. Weaker consistency conditions like processor consistency [7], causal serializability [6, 7] and weak consistency [7] have also been considered in the TM context when proving impossibility results.

Strict serializability, as well as serializability, are usually presented in an informal way in TM papers which cite the original paper [23] where these conditions have first appeared in the context of database research. Thus, the differences that exist between database and TM transactions have been neglected in TM research. We present formal definitions of these consistency conditions here. Additional consistency conditions originating from the database research are presented in [3]. To present their formalism, the authors of [3] make the restrictive assumption that transactional operations are atomic. The presentation of most of the other consistency conditions (e.g. opacity [15, 16], virtual world consistency [21], snapshot isolation [2, 10, 25, 6, 7] and weaker variants of them [6, 7]) is based on the assumption that a read for a data item by a transaction T can read a value written by either transaction that has committed or is commit-pending when T starts its execution. Finally, virtual world consistency [21] has been presented in a rather informal way and its definition is based on the assumption that each instance of

[2] This is not achieved by employing the commit-oriented version of the definitions since the notion of pieces of code that are "commit-pending" is not defined for universal constructions.

WRITEDI writes a distinct value for the data item it accesses (or that the transactional operations are executed atomically).

2 TM Model

In this section, we describe a model for transactional memory (TM) computing.

2.1 Transactions and histories

Transactional memory (TM) is a parallel programing paradigm which employs transactions to synchronize the execution of threads. A *transaction* is a piece of code which accesses pieces of data, called *data items*. A data item may be accessed by several threads simultaneously in a concurrent environment. A TM algorithm uses base objects to store the state of each data item and ensures synchronization between threads accessing the same data items. A *base object* has a state and supports a set of operations, called *primitives*, to access or update its state. Base objects are usually simple objects that are provided by the hardware.

In order to *read* or *write* a data item, the transaction's code must call specific routines, called READDI and WRITEDI, respectively. The TM algorithm provides implementations for these routines from the base objects. A transaction may commit or abort. If it *commits*, all its updates to data items are realized, whereas if it *aborts*, all its updates are discarded. The TM algorithm provides implementations for two routines, called ABORT and TRYCOMMIT, which are called to try to commit or to abort a transaction, respectively. We refer to all these routines as *transactional operations*. Whenever it is clear from the context, we use the term operation to refer to a transactional operation.

A transactional operation starts its execution when the thread executing it issues an *invocation* for it; the operation completes its execution when the thread executing it returns a *response*. A valid response for an instance of TRYCOMMIT executed by some transaction T can be either C_T which identifies that T has committed, or A_T which identifies that T has aborted. A valid response for an instance of ABORT executed by T is always A_T. A valid response for READDI can be either a value or A_T; finally, a valid response for WRITEDI can be either an acknowledgment or A_T. An *event* is either an invocation or a response of a transactional operation. A *history* is a finite sequence of events. We say that a response res *matches* an invocation inv in some history H, if they are both by the same thread p, res is a valid response for inv, res follows inv in H, and there is no other response by p between inv and res in H. A transactional operation is *complete*, if there is a response for it; otherwise, the operation is *pending*.

Thus, in a history H, there are two events for every completed operation op, an invocation $inv(op)$ and a matching response $res(op)$. H contains only the invocation of each pending operation in it. For each data item x, we denote by $H \mid x$ the subsequence of H containing the invocations and responses of all transactional operations that access x. For each thread p_i, we denote by $H \mid p_i$ the subsequence of H containing all invocations and responses of transactional operations executed by p_i. For each event e in H, we denote by $H \uparrow e$ the longest prefix of H that does not include e.

Consider any history H. We say that a transaction T (executed by a thread p_i) *is in* H or H *contains* T, if there are events in H issued by p_i when executing T. The *transaction subhistory* of H for T, denoted by $H \mid T$, is the subsequence of all events in H issued by p_i when executing T. Each transaction T in H for which $H \mid T$ contains at least one invocation of WriteDI is called an *update* transaction. A transaction in H is called *read-only*, if it is not an update transaction.

A history H is said to be *well-formed* if, for each transaction T in H, $H \mid T$ is an alternating sequence of invocations and matching responses, starting with an invocation, such that:

- no events in $H \mid T$ follow C_T or A_T;
- if T' is any transaction in H executed by the same thread that executes T, either the last event of $H \mid T$ precedes in H the first event of $H \mid T'$ or the last event of $H \mid T'$ precedes in H the first event of $H \mid T$.

From now on we focus on well-formed histories. Let H be any such history. A transaction T is *committed* in H, if $H \mid T$ includes C_T; a transaction T is *aborted* in H, if $H \mid T$ includes A_T. A transaction is *completed* in H, if it is either committed or aborted, otherwise it is *live*.

A transaction is *commit-pending* in H if it is live in H and $H \mid T$ includes an invocation to TryCommit for T. We denote by $comm(H)$ the subsequence of all events in H issued and received by committed transactions. Two histories H and H' are said to be *equivalent* if each thread p executed the same transactions, in the same order, in H and H', and for every transaction T in H, $H \mid T = H' \mid T$, i.e. for each transaction the same transactional operations are invoked and each of these operations has the same response in both histories.

Consider any history H. We denote by $Complete(H)$ a set of histories that extend H. Specifically, a history H' is in $Complete(H)$ if and only if, all of the following hold:

1. H' is well-formed, H is a prefix of H', and H and H' contain the same set of transactions;

2. for every live transaction[3] T in H:

   (a) if $H \mid T$ ends with an invocation of TryCommit, H' contains either C_T or A_T;

   (b) if $H \mid T$ ends with an invocation other than TryCommit, H' contains A_T;

   (c) if $H \mid T$ ends with a response, H' contains Abort_T and A_T.

Roughly speaking, each history in $Complete(H)$ is an extension of H where some of the commit-pending transactions in H appear as committed and all other live transactions appear as aborted.

A *configuration* is a vector consisting of the state of each thread and the state of each base object. In an *initial configuration*, threads and base objects are in initial states. A *step* of a thread consists of applying a single primitive on some base object, the response to that primitive, and zero or more local operations that are performed after the access and which may cause the internal state of the thread to change. As a step, we also consider the invocation of a transactional operation or the response to such an invocation; notice that a step of this kind does not change the state of any base object. Each step is executed atomically. An *execution* α is a sequence of steps starting from an initial configuration. An execution is *legal* starting from a configuration C if the sequence of steps performed by each thread follows the algorithm for that thread (starting from its state in C) and, for each base object, the responses to the operations performed on the object are in accordance with its specification (and the state of the object at configuration C). Given an execution α, the history of α, denoted by H_α, is the subsequence of α consisting of just the invocations and the responses of transactional operations.

The *execution interval* of a completed transaction T in an execution α is the subsequence of consecutive steps of α starting with the first step executed by any of the operations invoked by T and ending with the last such step. The *execution interval* of a transaction T that does not complete in α is the suffix of α starting with the first step executed by any of the operations invoked by T.

2.2 Relations and Partial Orders

Consider a well-formed history H. We define a partial order, called *real time order* and denoted $<_H$, on the set of *transactions* in H as follows:

- for any two transactions T_1 and T_2 in H, if T_1 is completed in H and the last event of $H \mid T_1$ precedes the first event of $H \mid T_2$ in H, then $T_1 <_H T_2$.

[3] We remark that the order in which the live transactions of H are inspected to form H' is immaterial, i.e. all histories that result from any possible such order are added in $Complete(H)$.

Transactions T_1 and T_2 are *concurrent* in H, if neither $T_1 <_H T_2$ nor $T_2 <_H T_1$. Similarly, transactions T_1 and T_2 are *concurrent* in an execution α, if neither $T_1 <_{H_\alpha} T_2$ nor $T_2 <_{H_\alpha} T_1$. We say that a history H (or an execution α) is *sequential* if no two transactions in H (or in α, respectively) are concurrent.

We also define a partial order, called *operation real-time* order and denoted by $<^{op}_H$, on the set of *transactional operations* in H as follows:

- for any two transactional operations op_1 and op_2 in H, if H contains a response for op_1 which precedes the invocation of op_2, then $op_1 <^{op}_H op_2$.

Operations op_1 and op_2 are *concurrent* in H, if neither $op_1 <^{op}_H op_2$ nor $op_2 <^{op}_H op_1$. H is *operation-wise sequential* if no two operations in H are concurrent.

Let S_{op} be an operation-wise sequential history equivalent to H. We say that S_{op} *respects* some relation $<$ on the set of *transactions* in H if the following holds: for any two transactions T_1 and T_2 in S, if $T_1 < T_2$, then $T_1 <_S T_2$. We say that S_{op} respects some relation $<^{op}$ on the set of *transactional operations* in H if the following holds: for any two operations op_1 and op_2 in S_{op}, if $op_1 <^{op} op_2$, then $op_1 <^{op}_{S_{op}} op_2$. Notice that a partial order is a relation, so these definitions hold for partial orders as well.

Consider any operation-wise sequential history S_{op} that is equivalent to H and respects $<_H$. We define a binary relation (with respect to S_{op}), called *reads-from* and denoted by $<^r_H$, between *transactions* in H such that, for any two transactions T_1, T_2 in H, $T_1 <^r_H T_2$ only if:

- T_2 executes a ReadDI operation op that reads some data item x and returns a value v for it,
- T_1 is the transaction in S_{op} which executes the last WriteDI operation that writes v for x and precedes op.

Notice that each operation-wise sequential history S_{op} that is equivalent to H, induces a *reads-from* relation. We denote by $\mathcal{R}_H$ the set of all reads-from relations that can be induced for H.

For each $<^r_H$ in $\mathcal{R}_H$, we define the *causal* relation for $<^r_H$ on transactions in H to be the transitive closure of $\bigcup_i \left(<_{H|p_i}\right) \cup <^r_H$. We define $\mathcal{C}_H$ to be the set of all causal relations in H.

2.3 Legality

A set $\mathcal{S}$ of sequences is prefix-closed if, whenever H is in $\mathcal{S}$, every prefix of H is also in $\mathcal{S}$. A history H is a *single data-item* history for some data item x, if

$H \mid x = H$. Consider a sequential history S and a transaction T in S. We say that T is *legal* in S, if for every invocation *inv* of ReadDI on each data item x that T performs whose response is not A_T the following hold:

1. if there is an invocation of WriteDI for x by T that precedes *inv* in S then v is the argument of the last such invocation,
2. otherwise, if there are no committed transactions preceding T in S which invoke WriteDI for x, then v is the initial value for x,
3. otherwise, v is the argument of the last invocation of WriteDI of any committed transaction that precedes T in S.

A complete sequential history S is legal if every transaction in S is legal.

3 TM Consistency

3.1 Strict Serializability

Strict serializability was first introduced in [23] as a (strong) consistency condition for executions of concurrent transactions in database systems. In TM computing, it can be expressed in several different flavors, two of which are discussed below. We start with *live strict serializability* (or *ℓ-strict serializability* for short).

Definition 1 (Live Strict Serializability or L-Strict Serializability)**.** *We say that an execution α is* ℓ-strictly serializable *if it is possible to do all of the following:*

- *If A is the set of all complete transactions in α that are not aborted, for each transaction $T \in A$, to insert a serialization point $*_T$ somewhere between T's first invocation of a transactional operation and T's last response for a transactional operation in α.*
- *To choose a subset B of the live transactions in α and, for each transaction $T \in B$, insert a serialization point $*_T$ somewhere after T's first invocation of a transactional operation in α.*

These serialization points should be inserted, so that, in the sequential execution σ that we get by serially executing each transaction $T \in A \cup B$ at the point that its serialization point has been inserted, the following hold:

- *for each transaction $T \in A$, the same transactional operations, as in α, are invoked by T in σ and the response of each such operation in σ is the same as that in α, and*

- *for each transaction $T \in B$, a prefix of the operations*[4] *invoked by T in σ is the same as the sequence of operations invoked by T in α and the response of each such operation in σ is the same as that in α.*

If an execution α is ℓ-strictly serializable, there exists a sequential execution σ that satisfies the properties of Definition 1; we say that σ *justifies* that α is ℓ-strictly serializable.

We continue to provide a stronger version of ℓ-strict serializability in Definition 2 called *commit-oriented strict serializability* (or *c-strict serializability* for short) which is based on the definition of $Complete$.

Definition 2 (C-Strict Serializability)**.** *A history H is* c-strictly serializable*, if there exist a history $H' \in Complete(H)$ and a history S equivalent to $comm(H')$ such that:*

- *S is a legal sequential history, and*
- *S respects $<_{comm(H')}$.*

We remark that Definition 1 provides a weaker version of strict serializability than Definition 2, since it allows a transaction to read a value for a data item written by another transaction that is not committed or commit-pending in H. This is allowed only if eventually, all complete transactions that are not aborted, and some of those that are still live can be "serialized" within their execution intervals. For instance, let's consider the history H and its prefix H_1 both shown in Figure 1. H is both ℓ-strictly serializable and c-strictly serializable, whereas H_1 is just ℓ-strictly serializable. Notice that since ℓ-strict serializability is weaker than c-strict serializability, the universe of algorithms that are ℓ-strictly serializable is larger than that of the algorithms that are c-strictly serializable.

We remark that c-strict serializability is not a prefix-closed property. On the contrary, ℓ-strict serializability is a prefix-closed property. We remark that prefix-closure can be imposed to c-strict serializability in an explicit way, i.e. by directly stating in Definition 2 that each prefix H_p of H must also satisfy the conditions imposed by the definition (as it is done in Definition 5 in Section 3.3). However, this would make Definition 2 even stronger, and therefore the resulted universe of c-strictly serializable TM algorithms even smaller.

3.2 Serializability

As with strict serializability, serializability was first introduced in [23] as a consistency condition for executions of concurrent transactions in database systems.

[4]Notice that since σ is a sequential execution, each transaction $T \in B$ commits in σ.

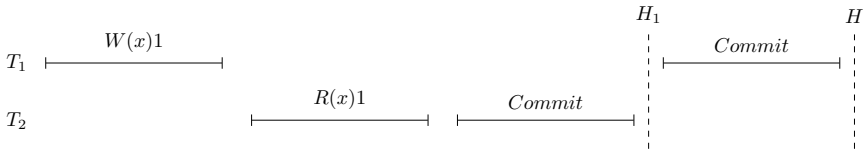


Figure 1: Example showing that strict serializability is not a prefix-closed property.

Below we discuss two different flavors of serializability in a way similar to that for strict serializability.

Definition 3 (L-Serializability)**.** *We say that an execution α is ℓ-*serializable *if it is possible to do all of the following:*

- *If A is the set of all complete transactions in α that are not aborted, for each transaction $T \in A$, to insert a serialization point $*_T$ in α.*
- *To choose a subset B of the live transactions in α and, for each transaction $T \in B$, insert a serialization point $*_T$ in α.*

These serialization points should be inserted, so that, in the sequential execution σ that we get by serially executing each transaction $T \in A \cup B$ at the point that its serialization point has been inserted, the following hold:

- *for each transaction $T \in A$, the same transactional operations, as in α, are invoked by T in σ and the response of each such operation in σ is the same as that in α, and*
- *for each transaction $T \in B$, a prefix of the operations invoked by T in σ is the same as the sequence of operations invoked by T in α and the response of each such operation in σ is the same as that in α.*

We continue to provide a stronger version of serializability in Definition 4, called *commit-oriented serializability* (or *c-serializability* for short), which is based on the definition of $Complete$.

Definition 4 (C-Serializability)**.** *A history H is* c-serializable*, if there exist a history $H' \in Complete(H)$ and a history S equivalent to $comm(H')$ such that:*

- *S is a legal sequential history.*

Notice that S in Definition 4 respects the program order of transactional operations executed by the same thread in H. This is implied by the definition of equivalent histories.

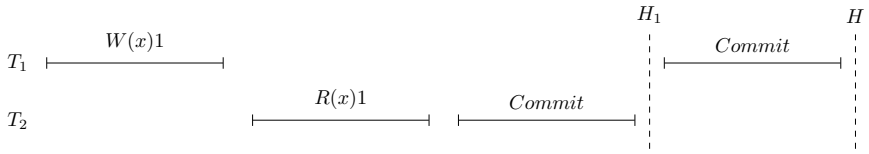


Figure 2: Example showing that serializability is not a prefix-closed property.

We remark that, similarly to the corresponding definitions of strict serializability, Definition 3 provides a weaker version of serializability than Definition 4.

The difference between serializability and strict serializability is that strict serializability additionally ensures that the real-time order of transactions is respected by the sequential history defined by the serialization points. Thus, every history/execution that is (c-) ℓ-strictly serializable is also (c-) ℓ-serializable but not vice versa.

It is worth-pointing out that ℓ-serializability and c-serializability are not prefix-closed properties. This is so, since it is easy to design a history H which is ℓ-serializable (as well as c-serializable) in which a committed transaction T (executed by some thread p) reads for some data item x a value v written by some other committed (or commit-pending) transaction T' such that T' is executed by some thread $p' \neq p$ in H and T''s execution has started after T has been completed. Aparently, the prefix of H up until C_T is neither ℓ-serializable, nor c-serializable.

We remark that prefix-closure can be imposed to ℓ-serializability (as well as to c-serializability) in an explicit way, as discussed for c-strict serializability above. It is not clear if the versions that would then result will be weaker than the corresponding versions of strict serializability. Imposing prefix closure to the consistency conditions presented in Sections 3.4.1-3.5 may be too restrictive as well. Thus, we present the non-prefix-closed versions of them given that it is straightforward to derive their prefix-closed versions, in an explicit way.

Several impossibility results [4, 8, 13] and lower bounds [4] in TM computing have been proved for strict serializability or serializability. Most TM algorithms in the literature (see e.g. [9, 28, 11, 26] for some examples) satisfy some form of serializability.

3.3 Opacity

Opacity was first introduced in [15]. In [16], a prefix-closed version of it was formally stated. Here, we will present the later version which we will call c-opacity (to be coherent with definitions in previous sections).

Definition 5 (C-Opacity [16])**.** *A history H is c-opaque if, for each prefix H_p of H, there exists a sequential history S_p equivalent to some history $H'_p \in Complete(H_p)$ such that:*

- *S_p is legal, and*
- *S_p respects $<_{H'_p}$.*

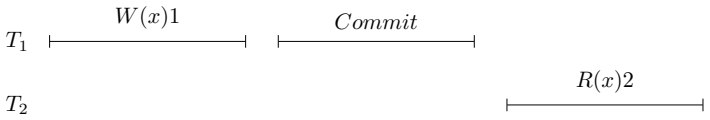


Figure 3: A strictly serializable history which is not opaque.

C-opacity is stronger than c-strict serializability. Figure 3 shows an example of a history that is not c-opaque but is c-strictly serializable. This history is not c-opaque because it violates the last condition of Definition 5; specifically, transaction T_2 cannot be legal.

Strict serializability (independently of the variant we consider) doesn't impose any restrictions on non-committed (or not commit-pending) transactions, whereas c-opacity requires that all reads of each transaction T (independently of whether the transaction is committed, aborted or live in the considered history) read values written by previously committed transactions (or by T itself). This additional property is required in order to avoid undesired situations where a transaction may cause an exception or enter into an infinite loop after reading a value for a data item written by a live transaction that may eventually abort.

It is remarkable that the first of these undesired situations (i.e. the production of an exception or an error code) can be avoided even by TM system that ensure only strict serializability if we make the following simple assumptions in our model. An exception (or an error code) that has been resulted by the execution of a transactional operation op is considered as a response for op. A transaction that has experienced an exception or has received an error code as a response, to one of its operations, is considered to be completed (but not aborted). Then, a (ℓ- or c-) strictly serializable TM implementation will never produce such exceptions (or error codes). Notice that the second undesirable situation, namely having some transaction enter an infinite loop, will not appear in TM systems that ensure standard progress properties, like lock-freedom, starvation-freedom, etc.

We continue to present live opacity (ℓ-opacity). Consider any history H and a transaction T in H. An instance op of ReadDI for some data item x executed by T is *global* if T has not invoked WriteDI on x in H before invoking op. Let

$T|read$ be the longest subsequence of $H|T$ consisting only of those invocations of ReadDI for which there is a response and this response is not A_T, followed by $\text{TryCommit}_T, C_T$. Let $T|read_g$ be the subsequence of $T|read$ consisting only of the invocations of the *global* instances of ReadDI and their responses, followed by $\text{TryCommit}_T, C_T$. We denote by T_r a transaction that invokes the same transactional operations (and in the same order) as those invoked in $T|read$. Similarly, denote by T_{gr} a transaction that invokes the same transactional operations (and in the same order) as those invoked $T|read_g$. For each ReadDI operation op on any data item x that is in T_r but not in T_{gr}, we say that the response for op (if it exists) is *legal*, if it is the value written by the last WriteDI for x performed by T before the invocation of op.

Definition 6 (L-Opacity)**.** *We say that an execution α is ℓ-*opaque *if there exists some sequential execution σ which justifies that α is ℓ-strictly serializable, and all of the following hold:*

1. *We can extend the history H_σ of σ to get a* sequential *history H'_σ such that:*
   - *for each transaction T in α that is not in σ, H'_σ contains $T|read_{gr}$*
   - *if $<$ is the partial order which is induced by the real time order $<_{H_\alpha}$ in such a way that for each transaction T in α that is not in σ, we replace each instance of T in the set of pairs of $<_{H_\alpha}$ with transaction T_{gr}, then H'_σ respects $<$*
   - *H'_σ is legal*
2. *for each transaction T in α that is not in σ, and for each transactional operation op in $T|read$ that is not in $T|read_{gr}$, the response for op is legal.*

We remark that most TM algorithms presented in the literature are opaque.

3.4 Causality-Related Consistency Conditions

3.4.1 Causal Consistency

Causal consistency was informally introduced as a shared memory consistency condition in [20], and it was formally defined in [1]. As in the previous sections, we provide two formal definitions of causal consistency for TM computing using the framework of Section 2.

Definition 7 (L-Causal Consistency)**.** *Consider an execution α and let A be the set of all complete transactions in α that are not aborted. We say that α is ℓ-*causally-consistent *if there exists a subset B of live transactions in α and a causal relation $<^c$ in $C_{H'_\alpha}$ where H'_α is the subsequence of H_α containing just the events*

of transactions in $A \cup B$, such that, for each thread p_i, *it is possible to do the following:*
For each transaction $T \in A \cup B$, to insert a serialization point $*_T$ in α so that, if σ_i is the sequential execution that we get by serially executing each transaction $T \in A \cup B$ at the point that its serialization point has been inserted, then the following hold:

- *H_{σ_i} respects $<^c$,*
- *for each transaction $T \in A$, the same transactional operations, as in α, are invoked by T in σ_i and the response of each such operation in σ_i is the same as that in α, and*
- *for each transaction $T \in B$, a prefix of the operations invoked by T in σ_i are the same as the sequence of operations invoked by T in α, the response of each such operation in σ_i is the same as that in α.*

Definition 8 (C-Causal Consistency)**.** *A history H is* c-causally consistent *if there exists a history $H' \in Complete(H)$ and a causal relation $<^c$ in $C_{comm(H')}$ such that, for each thread p_i, there exist a sequential history S_i such that:*

- *S_i is equivalent to comm(H'),*
- *S_i respects the causality order $<^c$, and*
- *every transaction executed by p_i in S_i is legal.*

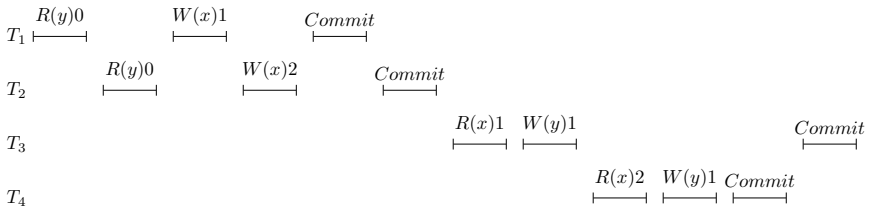


Figure 4: A causally consistent history which is not serializable.

L-causal consistency and c-causal consistency are weaker properties than ℓ-serializability and c-serializability, respectively. For instance, Figure 4 shows an example of a history which is (ℓ- and c-) causally consistent but not (ℓ- or c-) serializable. In this history both transactions T_1 and T_2 should be serialized before transactions T_3 and T_4, because both T_1 and T_2 read 0 from data item y

which is written by T_3 and T_4. Regardless of how the serialization points for T_1 and T_2 are ordered, both T_3 and T_4 should read the same value for data item x. Thus, this history is not serializable. However, it is causally consistent because threads running T_3 and T_4 may see writes executed by threads running T_1 and T_2 in a different order.

3.4.2 Causal Serializability

Causal serializability was introduced in [24] as a consistency condition which is stronger than causal consistency but weaker than serializability. Informally, in addition to the constraints imposed by causal consistency, the following constraint must also be satisfied: all transactions that update the same data item must be perceived in the same order by all threads.

Definition 9 (L-Causal Serializability)**.** *Consider an execution α and let A be the set of all complete transactions in α that are not aborted. We say that α is* ℓ-causally serializable *if there exists a subset B of live transactions in α and a causal relation $<^c$ in $C_{H'_\alpha}$ where H'_α is the subsequence of H_α containing just the events of transactions in $A \cup B$, such that,* for each thread *p_i, it is possible to do the following:*
For each transaction $T \in A \cup B$, to insert a serialization point $*_T$ in α so that, if σ_i is the sequential execution that we get by serially executing each transaction $T \in A \cup B$ at the point that its serialization point has been inserted, then the following hold:

- *H_{σ_i} respects $<^c$,*
- *for each transaction $T \in A$, the same transactional operations, as in α, are invoked by T in σ_i and the response of each such operation in σ_i is the same as that in α,*
- *for each transaction $T \in B$, a prefix of the operations invoked by T in σ_i are the same as the sequence of operations invoked by T in α, the response of each such operation in σ_i is the same as that in α.*
- *for each pair of transactions $T_1, T_2 \in A \cup B$ that write to the same data item, if $T_1 <_{H_{\sigma_i}} T_2$, then for each $j \in \{1, \ldots, n\}$, it holds that $T_1 <_{H_{\sigma_j}} T_2$.*

Definition 10 (C-Causal Serializability)**.** *A history H is* c-causally serializable *if there exists a history $H' \in Complete(H)$ and a causal relation $<^c$ in $C_{comm(H')}$ such that, for each thread p_i, there exist a sequential history S_i for which the following hold:*

- *S_i is equivalent to $comm(H')$,*

- *S_i respects the causality order $<^c$,*
- *every transaction executed by p_i in S_i is legal, and*
- *for each pair of transactions T_1 and T_2 in comm(H') that write to the same data item, if $T_1 <_{S_i} T_2$, then for each $j \in \{1, \ldots, n\}$, it holds that $T_1 <_{S_j} T_2$.*

Obviously, every (ℓ- or c-) causally serializable history satisfies the properties of (ℓ- or c-, respectively) causal consistency, but the oposite is not true. For instance, the history shown in Figure 4 is not causally serializable, since threads executing transactions T_3 and T_4 do not see writes from T_1 and T_2 to data item x in the same order.

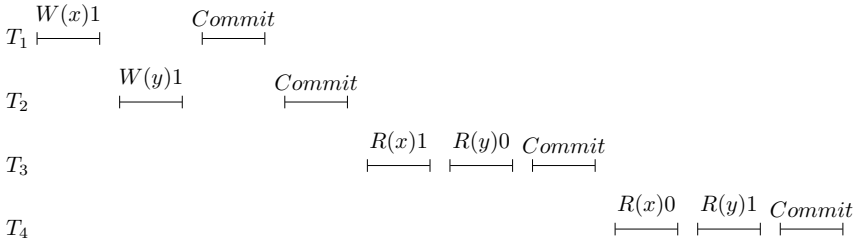


Figure 5: A causally serializable history which is not serializable.

Figure 5 shows an example of a history which is causally serializable but not serializable. Here, if transaction T_1 is serialized before T_2 (the opposite case is symmetrical), then it is not possible to serialize transaction T_4. However, by definition of causal serializability, sequential histories constructed for threads p_3 and p_4 may include transactions T_1 and T_2 in different orders.

In the context of TM research, causal consistency, as well as causal serializability, are interesting in the context of proving impossibility results [6, 7] and lower bounds. We remark that when proving such results, considering a weak consistency condition makes the result stronger. It is therefore an interesting open problem to see whether some of the TM impossibility results (e.g. [4, 8, 13]) that have been proved assuming some strong consistency condition, like opacity, strict serializability or serializability, can be extended to hold for weaker consistency conditions like those formulated in this or later sections. For instance in this avenue, the impossibility result proved in [14] assuming serializability is extended in [6, 7] to hold for a much weaker consistency condition.

3.4.3 Virtual World Consistency

Virtual World Consistency (VWC) was defined in [21] as a family of consistency conditions. Informally, VWC ensures serializability or strict serializability for the committed (and some of the commit-pending) transactions but a weaker condition than that imposed by opacity for the rest of the transactions.

For each transaction T in history H and each causal relation $<^c_H$ in C_H, we define the *causal past* of T denoted by $past_T(H, <^c_H)$ as the subsequence of all events produced either by transaction T in H itself or by any transaction T_i in H such that $T_i <^c_H T$.

Definition 11 (C-Virtual World Consistency)**.** *A history H is* c-virtual world consistent *if there exists a history $H' \in Complete(H)$ and a causal relation $<^c$ in $C_{H'}$ such that:*

- *there exists a legal sequential history S which is equivalent to $comm(H')$, and*
- *for each transaction T_i in H' that is not in S, there exists a legal sequential history S_i which is equivalent to $past_{T_i}(H', <^c)$ and respects the restriction of $<^c$ to those pairs whose components are transactions in $past_{T_i}(H', <^c)$.*

Definition 12 (C-Strong Virtual World Consistency)**.** *A history H is* c-strong virtual world consistent *if there exists a history $H' \in Complete(H)$ and a causal relation $<^c$ in $C_{H'}$ such that:*

- *there exists a legal sequential history S which is equivalent to $comm(H')$ and respects the real-time order of H', and*
- *for each non-committed transaction T_i in H', there exists a legal sequential history S_i which is equivalent to $past_{T_i}(H', <^c_{H'})$ and respects the restriction of $<^c$ to those pairs whose components are transactions in $past_{T_i}(H', <^c)$.*

Clearly, virtual world consistency is a stronger consistency condition than serializability. Similarly, strong virtual world consistency is stronger than strict serializability. Still, strong virtual world consistency (and therefore also virtual world consistency) is weaker than opacity. The history shown in Figure 6 is strong virtual world consistent but not opaque: regardless of the order of the serialization points of transactions T_1 and T_2, it is not possible to derive a sequential history where both transaction T_3 and T_4 are legal.

The history shown in Figure 7 is a slightly modified version of the history shown in Figure 6. This history is virtual world consistent but not strong virtual world consistent. In this history, transactions T_1 and T_2 are not concurrent, and

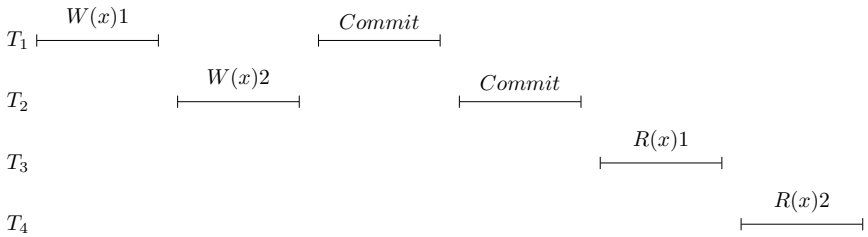


Figure 6: A virtual world consistent history which is not opaque.

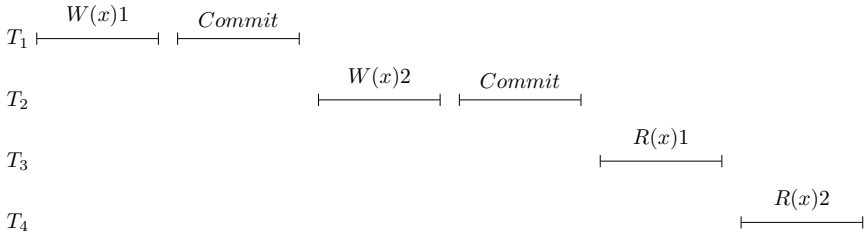


Figure 7: A virtual world consistent history which is not strong virtual world consistent.

since strong virtual world consistency respects the real-time order of transactions, there is only one way that the serialization points of these two transactions can be ordered in any equivalent sequential history.

We continue to present the live versions of virtual world consistency and strong virtual world consistency.

Definition 13 (L-Virtual World Consistency and L-Strong Virtual World Consistency)**.** *We say that an execution α is* ℓ-virtual world consistent *(ℓ-strong virtual world consistent) if there exists some sequential execution σ which justifies that α is ℓ-serializable (ℓ-strictly serializable, respectively), and the following holds:*

1. *for each transaction T_i in α that is not in σ there exists a legal sequential history S_i which is equivalent to $past_{T_i}(H', <^c)$ and respects the restriction of $<^c$ to those pairs whose components are transactions in $past_{T_i}(H', <^c)$.*

Strong consistency conditions such as opacity ensure the safe execution of non-committed transactions by imposing on them the same safety demands as those that committed transactions are required to obey. This has been criticized in [21] to result in TM algorithms that produce histories in which live transactions

are forced to abort in order to preserve the safety of other transactions that are deemed to also abort. Virtual world consistency relaxes the correctness criteria used for non-committed transactions in order to avoid such scenarios when possible, and by consequence, allow for more live transactions to commit, than a TM algorithm that implements a stronger criterion would.

3.5 Snapshot Isolation

Snapshot isolation was originally introduced as a safety property in the database world [5, 22]. Snapshot isolation is an appealing property for TM computing [2, 10, 25] since it provides the potential to increase throughput for workloads with long transactions [25]. The first formal definitions for TM snapshot isolation was given in [6, 7].

Consider a history H and let T be a transaction that either commits or is commit-pending in H. Recall that we have already defined the sequences $T|read$, $T|read_g$, as well as transactions T_r and T_{gr} in Section 3.3. Let $T|other$ be the subsequence of $H|T$ that consists of all invocations performed by T (and their matching responses) other than those comprising $T|read_g$, followed by $\text{TryCommit}_T, C_T$. Let T_o be a transaction that invokes the same transactional operations (and in the same order) as those invoked in $T|other$.

Definition 14 (C-Snapshot isolation [7])**.** *An execution α satisfies* c-snapshot isolation*, if there exists a set D consisting of all committed and some of the commit-pending transactions in α for which the following holds: for each transaction $T \in D$, it is possible to insert (in α) a global read point $*_{T,gr}$ and a write point $*_{T,w}$, so that if δ_α is the sequence defined by these serialization points, the following hold:*

1. *$*_{T,gr}$ precedes $*_{T,w}$ in δ_α,*
2. *both $*_{T,gr}$ and $*_{T,w}$ are inserted within the execution interval of T,*
3. *if H_{δ_α} is the history we get by replacing each $*_{T,gr}$ with T_{gr} and each $*_{T,w}$ with T_o in δ_α, then H_{δ_α} is legal.*

We finally present live snapshot isolation. Consider a legal execution α and let $C(\alpha)$ be the set of all legal executions such that each execution $\alpha' \in C(\alpha)$ is an extension of α such that the same transactions are executed in α and α' and no transaction is live in α'.

Definition 15 (L-Snapshot Isolation)**.** *Consider an execution α. We say that α satisfies* ℓ-snapshot isolation*, if there exists an execution $\alpha' \in C(\alpha)$ for which the following holds: if A is the set of transactions that commit in α' then for each*

transaction $T \in A$, it is possible to insert a global read point $*_{T,gr}$ and a write point $*_{T,w}$, so that:

1. *both $*_{T,gr}$ and $*_{T,w}$ are inserted within the execution interval of T in α*
2. *$*_{T,gr}$ precedes $*_{T,w}$, and*
3. *if σ is the sequential execution that we get when for each transaction $T \in A$, we serially execute transactions T_{gr} and T_o at the points that $*_{T,gr}$ and $*_{T,w}$ have been inserted, respectively, then for each transaction $T \in A$, the response of each transactional operation invoked by T_{gr} and T_o in σ is the same as that of the corresponding transactional operation in $T|read_g$ and $T|other$ (as defined based on α'), respectively.*

4 Acknowledgements

This work has been supported by the European Commission under the 7th Framework Program through the TransForm (FP7-MC-ITN-238639) project and by the ARISTEIA Action of the Operational Programme Education and Lifelong Learning which is co-funded by the European Social Fund (ESF) and National Resources through the GreenVM project.

We would like to thank Victor Bushkov for his valuable comments in a preliminary version of this paper and Eleftherios Kosmas for several useful discussions that motivated this work. Many thanks also to Hagit Attiya and Sandeep Hans for their comments on a previous version of this article.

References

[1] M. Ahamad, G. Neiger, J. E. Burns, P. Kohli, and P. W. Hutto. Causal memory: definitions, implementation, and programming. *Distributed Computing*, 9(1):37–49, 1995.

[2] M. S. Ardekani, P. Sutra, and M. Shapiro. The impossibility of ensuring snapshot isolation in genuine replicated stms. In *The 3rd edition of the Workshop on the Theory of Transactional Memory*, WTTM2011, 2011.

[3] H. Attiya and S. Hans. Transactions are Back-but How Different They Are? In *TRANSACT*, feb 2012.

[4] H. Attiya, E. Hillel, and A. Milani. Inherent limitations on disjoint-access parallel implementations of transactional memory. In *Proceedings of the twenty-first annual symposium on Parallelism in algorithms and architectures*, SPAA '09, pages 69–78, New York, NY, USA, 2009. ACM.

[5] H. Berenson, P. Bernstein, J. Gray, J. Melton, E. O'Neil, and P. O'Neil. A critique of ansi sql isolation levels. *SIGMOD Rec.*, 24(2):1–10, may 1995.

[6] V. Bushkov, D. Dziuma, P. Fatourou, and R. Guerraoui. Snapshot isolation does not scale either. Technical Report TR-437, Foundation of Research and Technology – Hellas (FORTH), 2013.

[7] V. Bushkov, D. Dziuma, P. Fatourou, and R. Guerraoui. The pcl theorem - transactions cannot be parallel, consistent and live. In *Proceedings of the 4th Annual ACM symposium on Parallelism in Algorithms and Architectures (SPAA 14)*. ACM Press, jul 2014.

[8] V. Bushkov, R. Guerraoui, and M. Kapałka. On the liveness of transactional memory. In *Proceedings of the 2012 ACM symposium on Principles of distributed computing*, PODC '12, pages 9–18, New York, NY, USA, 2012. ACM.

[9] L. Dalessandro, M. F. Spear, and M. L. Scott. Norec: streamlining stm by abolishing ownership records. In *Proceedings of the 15th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, PPoPP '10, pages 67–78, New York, NY, USA, 2010. ACM.

[10] R. J. Dias, J. Seco, and J. M. Lourenço. Snapshot isolation anomalies detection in software transactional memory. In *Proceedings of InForum 2010*, 2010.

[11] D. Dice and N. Shavit. What really makes transactions faster? In *Proc. of the 1st TRANSACT 2006 workshop*, 2006. Electronic, no. page numbers.

[12] S. Doherty, L. Groves, V. Luchangco, and M. Moir. Towards formally specifying and verifying transactional memory. *Formal Aspects of Computing*, pages 1–31, mar 2012.

[13] F. Ellen, P. Fatourou, E. Kosmas, A. Milani, and C. Travers. Universal constructions that ensure disjoint-access parallelism and wait-freedom. In *Proceedings of the 2012 ACM symposium on Principles of distributed computing*, PODC '12, pages 115–124, New York, NY, USA, 2012. ACM.

[14] R. Guerraoui and M. Kapalka. On obstruction-free transactions. In *Proceedings of the Twentieth Annual Symposium on Parallelism in Algorithms and Architectures*, SPAA '08, pages 304–313, New York, NY, USA, 2008. ACM.

[15] R. Guerraoui and M. Kapalka. On the correctness of transactional memory. In *Proceedings of the 13th ACM SIGPLAN Symposium on Principles and practice of parallel programming*, PPoPP '08, pages 175–184, New York, NY, USA, 2008. ACM.

[16] R. Guerraoui and M. Kapalka. *Principles of Transactional Memory (Synthesis Lectures on Distributed Computing Theory)*. Morgan and Claypool Publishers, 2010.

[17] T. Harris, J. Larus, and R. Rajwar. *Transactional Memory, 2Nd Edition*. Morgan and Claypool Publishers, 2nd edition, 2010.

[18] M. Herlihy. Wait-free synchronization. *ACM Trans. Program. Lang. Syst.*, 13(1):124–149, Jan. 1991.

[19] M. Herlihy and J. E. B. Moss. Transactional memory: architectural support for lock-free data structures. *SIGARCH Comput. Archit. News*, 21(2):289–300, May 1993.

[20] P. Hutto and M. Ahamad. Slow memory: weakening consistency to enhance concurrency in distributed shared memories. In *Distributed Computing Systems, 1990. Proceedings., 10th International Conference on*, pages 302–309, 1990.

[21] D. Imbs and M. Raynal. Virtual world consistency: A condition for STM systems (with a versatile protocol with invisible read operations). *Theoretical Computer Science*, 444(0):113 – 127, 2012. Structural Information and Communication Complexity ï£¡ SIROCCO 2009.

[22] R. Normann and L. T. Østby. A theoretical study of 'snapshot isolation'. In *Proceedings of the 13th International Conference on Database Theory*, ICDT '10, pages 44–49, New York, NY, USA, 2010. ACM.

[23] C. H. Papadimitriou. The serializability of concurrent database updates. *Journal of the ACM*, 26(4):631–653, oct 1979.

[24] M. Raynal, G. Thia-Kime, and M. Ahamad. From serializable to causal transactions for collaborative applications. In *EUROMICRO 97. New Frontiers of Information Technology., Proceedings of the 23rd EUROMICRO Conference*, pages 314–321, 1997.

[25] T. Riegel, C. Fetzer, and P. Felber. Snapshot isolation for software transactional memory. In *In Proceedings of the First ACM SIGPLAN Workshop on Languages, Compilers, and Hardware Support for Transactional Computing*, TRANSACT'06, 2006.

[26] T. Riegel, C. Fetzer, and P. Felber. Time-based transactional memory with scalable time bases. In *Proceedings of the Nineteenth Annual ACM Symposium on Parallel Algorithms and Architectures*, SPAA '07, pages 221–228, New York, NY, USA, 2007. ACM.

[27] N. Shavit and D. Touitou. Software transactional memory. In *Proceedings of the Fourteenth Annual ACM Symposium on Principles of Distributed Computing*, PODC '95, pages 204–213, New York, NY, USA, 1995. ACM.

[28] M. F. Spear, M. M. Michael, and C. von Praun. Ringstm: scalable transactions with a single atomic instruction. In *Proceedings of the twentieth annual symposium on Parallelism in algorithms and architectures*, SPAA '08, pages 275–284, New York, NY, USA, 2008. ACM.

The Logic in Computer Science Column

by

Yuri Gurevich

Microsoft Research

One Microsoft Way, Redmond WA 98052, USA

gurevich@microsoft.com

BEATCS no 113

CONTEXTUAL SEMANTICS: FROM QUANTUM MECHANICS TO LOGIC, DATABASES, CONSTRAINTS, AND COMPLEXITY

Samson Abramsky

Department of Computer Science

The University of Oxford

samson.abramsky@cs.ox.ac.uk

Abstract

We discuss quantum non-locality and contextuality, emphasising logical and structural aspects. We also show how the same mathematical structures arise in various areas of classical computation.

1 Introduction

In this paper we shall discuss some fundamental concepts in quantum mechanics: **non-locality**, **contextuality** and **entanglement**. These concepts play a central rôle in the rapidly developing field of quantum information, in delineating how quantum resources can transcend the bounds of classical information processing. They also have profound consequences for our understanding of the very nature of physical reality.

Our aim is to present these ideas in a manner which should be accessible to any computer scientist, and which emphasises the logical and structural aspects. We shall also show how the same mathematical structures which arise in our analysis of these ideas appear in a range of contexts in classical computation.

2 Alice and Bob look at bits

We consider the following scenario, depicted in Figure 1. Alice and Bob are agents positioned at nodes of a network. Alice can access local bit registers a_1

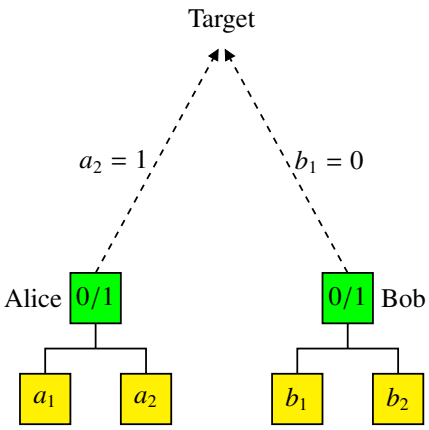


Figure 1: Alice and Bob look at bits

A	B	(0, 0)	(1, 0)	(0, 1)	(1, 1)
a_1	b_1	1/2	0	0	1/2
a_1	b_2	3/8	1/8	1/8	3/8
a_2	b_1	3/8	1/8	1/8	3/8
a_2	b_2	1/8	3/8	3/8	1/8

Figure 2: The Bell table

and a_2, while Bob can access local bit registers b_1, b_2. Alice can load one of her bit registers into a processing unit, and test whether it is 0 or 1. Bob can perform the same operations with respect to his bit registers. They send the outcomes of these operations to a common target, which keeps a record of the joint outcomes.

We now suppose that Alice and Bob perform repeated rounds of these operations. On different rounds, they may make different choices of which bit registers to access, and they may observe different outcomes for a given choice of register. The target can compile statistics for this series of data, and infer probability distributions on the outcomes. The probability table in Figure 2 records the outcome of such a process.

Consider for example the cell at row 2, column 3 of the table. This corresponds to the following event:

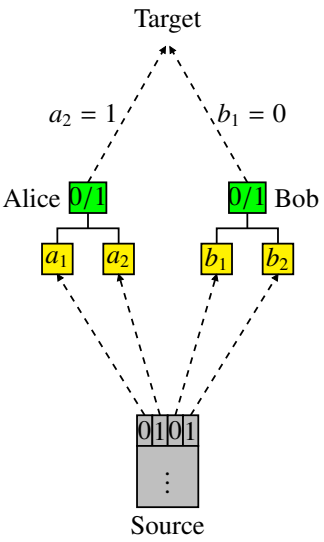


Figure 3: A Source

- Alice loads register a_1 and observes the value 0.
- Bob loads register b_2 and observes the value 1.

This event has the probability $1/8$, conditioned on Alice's choice of a_1 and Bob's choice of b_2.

Each row of the table specified a probability distribution on the possible joint outcomes, conditioned on the indicated choice of bit registers by Alice and Bob.

We can now ask:

How can such an observational scenario be realised?

The obvious classical mechanism we can propose to explain these observations is depicted in Figure 3.

We postulate a **source** which on each round chooses values for each of the registers a_1, a_2, b_1, b_2, and loads each register with the chosen value. Alice and Bob will then observe the values which have been loaded by the source. We can suppose that this source is itself randomised, and chooses the values for the registers according to some probability distribution P on the set of 2^4 possible assignments.

We can now ask the question: is there any distribution P which would give rise to the table specified in Figure 2?

Important Note A key observation is that, in order for this question to be non-trivial, we must assume that the choices of bit registers made by Alice and Bob are **independent** of the source.[1] If the source could determine which registers are to be loaded on each round, as well as their values, then it becomes a trivial matter to achieve **any** given probability distribution on the joint outcomes.

Under this assumption of independence, it becomes natural to think of this scenario as a kind of **correlation game**. The aim of the source is to achieve as high a degree of correlation between the outcomes of Alice and Bob as possible, whatever the choices made by Alice and Bob on each round.

3 Logic rings a Bell

We shall now make a very elementary and apparently innocuous deduction in elementary logic and probability theory, which could easily be carried out by students in the first few weeks of a Probability 101 course.

Suppose we have propositional formulas $\varphi_1, \ldots, \varphi_N$. We suppose further that we can assign a probability p_i to each φ_i.

In particular, we have in the mind the situation where the boolean variables appearing in φ_i correspond to empirically testable quantities, such as the values of bit registers in our scenario; φ_i then expresses a condition on the outcomes of an experiment involving these quantities. The probabilities p_i are obtained from the statistics of these experiments.

Now suppose that these formulas are **not simultaneously satisfiable**. Then (e.g.)

$$\bigwedge_{i=1}^{N-1} \phi_i \;\rightarrow\; \neg\phi_N, \quad \text{or equivalently} \quad \phi_N \;\rightarrow\; \bigvee_{i=1}^{N-1} \neg\phi_i.$$

Using elementary probability theory, we can calculate:

$$p_N \;\leq\; \mathsf{Prob}(\bigvee_{i=1}^{N-1} \neg\phi_i) \;\leq\; \sum_{i=1}^{N-1} \mathsf{Prob}(\neg\phi_i) \;=\; \sum_{i=1}^{N-1}(1-p_i) \;=\; (N-1) - \sum_{i=1}^{N-1} p_i.$$

The first inequality is the monotonicity of probability, and the second is sub-additivity.

Hence we obtain the inequality

$$\sum_{i=1}^{N} p_i \;\leq\; N-1.$$

[1] This translates formally into a conditional independence assumption, which we shall not spell out here; see e.g. [14, 17].

We shall refer to this as a **logical Bell inequality**, for reasons to be discussed later. Note that it hinges on a purely logical consistency condition.

3.1 Logical analysis of the Bell table

We return to the probability table from Figure 2.

	$(0,0)$	$(1,0)$	$(0,1)$	$(1,1)$
(a_1, b_1)	1/2	0	0	1/2
(a_1, b_2)	3/8	1/8	1/8	3/8
(a_2, b_1)	3/8	1/8	1/8	3/8
(a_2, b_2)	1/8	3/8	3/8	1/8

If we read 0 as true and 1 as false, the highlighted entries in each row of the table are represented by the following propositions:

$$\begin{aligned}
\varphi_1 &= (a_1 \wedge b_1) \vee (\neg a_1 \wedge \neg b_1) = a_1 \leftrightarrow b_1 \\
\varphi_2 &= (a_1 \wedge b_2) \vee (\neg a_1 \wedge \neg b_2) = a_1 \leftrightarrow b_2 \\
\varphi_3 &= (a_2 \wedge b_1) \vee (\neg a_2 \wedge \neg b_1) = a_2 \leftrightarrow b_1 \\
\varphi_4 &= (\neg a_2 \wedge b_2) \vee (a_2 \wedge \neg b_2) = a_2 \oplus b_2.
\end{aligned}$$

The events on first three rows are the correlated outcomes; the fourth is anticorrelated. These propositions are easily seen to be jointly unsatisfiable. Indeed, starting with φ_4, we can replace a_2 with b_1 using φ_3, b_1 with a_1 using φ_1, and a_1 with b_2 using φ_2, to obtain $b_2 \oplus b_2$, which is obviously unsatisfiable.

It follows that our logical Bell inequality should apply, yielding the inequality

$$\sum_{i=1}^{4} p_i \leq 3.$$

However, we see from the table that $p_1 = 1$, $p_i = 6/8$ for $i = 2, 3, 4$. Hence the table yields a violation of the Bell inequality by $1/4$.

This rules out the possibility of giving an explanation for the observational behaviour described by the table in terms of a classical source. We might then conclude that such behaviour simply cannot be realised. However, as we shall now see, **in the presence of quantum resources, this is no longer the case**.

3.2 A crash course in qubits

We shall now very briefly give enough information about some of the primitives of quantum computing to show how these can be used to realise behaviour such as that in the Bell table in Figure 2. There are a number of excellent introductions to quantum computing aimed at or accessible to computer scientists (see e.g. [31, 36, 32]), and we refer the reader seeking more detailed information to these.

A classical bit register of the kind we began our discussion with can hold the values 0 or 1; we can say that it has two possible states. The operations we can perform on such a register, or an array of such registers, include:

- Reading the value currently held in the register without changing the state of the register.

- Using the values currently held in one or more such registers to compute a new value according to any boolean function.

In quantum computing, we introduce a new object, the qubit, with very different properties. The key features of the qubit are best explained using the beautiful geometric representation in terms of the "Bloch sphere" (the unit 2-sphere), as illustrated in Figure 4.

Note the following key features:

- States of the qubit[2] are represented as points on the surface of the sphere. In Figure 4, a state $|\psi\rangle$ is depicted. Note that there are a continuum of possible states.

- Each pair $(\mathsf{Up}, \mathsf{Down})$ of antipodal points on the sphere define a possible measurement that we can perform on the qubit. Each such measurement has two possible outcomes, corresponding to Up and Down in the given direction. We can think of this physically e.g. as measuring Spin Up or Spin Down in a given direction in space.

- When we subject a qubit to a measurement $(\mathsf{Up}, \mathsf{Down})$, the state of the qubit determines a probability distribution on the two possible outcomes. For a geometric view on this see Figure 5. The probabilities are determined by the **angles** between the qubit state $|\psi\rangle$ and the points $(|\mathsf{Up}\rangle, |\mathsf{Down}\rangle)$ which specify the measurement. In algebraic terms, $|\psi\rangle$, $|\mathsf{Up}\rangle$ and $|\mathsf{Down}\rangle$ are unit vectors in the complex vector space $\mathbb{C}^2$, and the probability of observing Up

[2] More precisely, the pure states; mixed states are represented as points in the interior of the sphere.

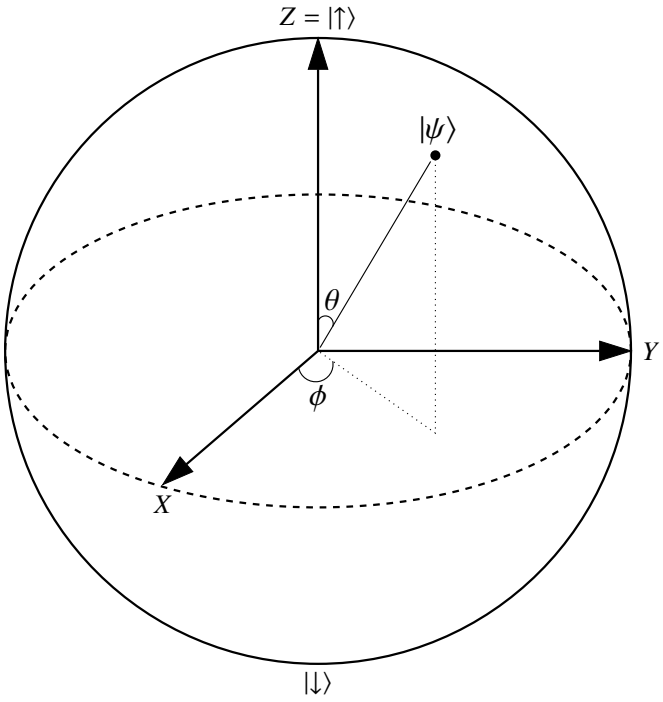


Figure 4: The Bloch sphere representation of qubits

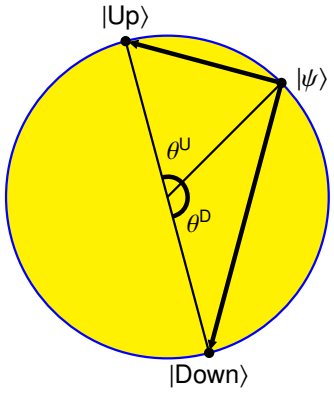


Figure 5: Truth makes an angle with reality

when in state $|\psi\rangle$ is given by the square modulus[3] of the inner product:

$$|\langle\psi|\mathsf{Up}\rangle|^2.$$

This is known as the **Born rule**. It gives the basic predictive content of quantum mechanics.

- Note in addition that a measurement has an **effect** on the state, which will no longer be the original state $|\psi\rangle$, but rather one of the states Up or Down, in accordance with the measured value.

The sense in which the qubit generalises the classical bit is that, for each question we can ask — *i.e.* for each measurement — there are just two possible answers. We can view the states of the qubit as superpositions of the classical states 0 and 1, so that we have a probability of getting each of the answers for any given state.

But in addition, we have the important feature that there are a continuum of possible questions we can ask. However, note that on each run of the system, we can only ask **one** of these questions. We cannot simultaneously observe Up or Down in two different directions. Note that this corresponds to the feature of the scenario we discussed in Section 1, that Alice and Bob could only look at one their local registers on each round.

[3]Recall that the square modulus of a complex number $z = a + ib$ is given by $|z|^2 = zz^* = (a + ib)(a - ib)$.

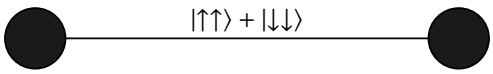


Figure 6: The Bell state

3.3 Compound systems and entanglement

The deeper features of quantum behaviour are revealed when we look at **compound systems** of multiple qubits.[4] It is here that we find the phenomena of quantum entanglement and non-locality.

Consider for example the 2-qubit system shown in Figure 6. We can think of Alice holding one qubit, and Bob the other. The combined state of the system is described by the vector $|\uparrow\uparrow\rangle + |\downarrow\downarrow\rangle$.[5] According to the standard postulates of quantum mechanics, when Alice measures her qubit, she may, with equal probability, get either answer (Spin Up or Down). If she gets the answer Spin Up, then the state of the entangled qubit becomes $|\uparrow\uparrow\rangle$, so that if Bob now measures his qubit, he can **only** get the answer Spin Up; while if she gets the answer Spin Down, the state becomes $|\downarrow\downarrow\rangle$, and Bob can only get the answer Spin Down. This is regardless of the fact that Bob may be far away from Alice (spacelike separated). This is the phenomenon that Einstein famously referred to as "spooky action at a distance", and which Schrödinger named **entanglement**.

How can the world be this way? This remains a challenge to our understanding of the nature of physical reality. Meanwhile, though, the field of quantum information seeks to understand how entanglement can be **used** as a new kind of resource, opening up new possibilities which transcend those of the classical models of information and computation.

3.4 From the Bell state to the Bell table

We refer again to the table in Figure 2. This table can be physically realised, using the Bell state $(|\uparrow\uparrow\rangle + |\downarrow\downarrow\rangle)/\sqrt{2}$, with Alice and Bob performing 1-qubit local measurements corresponding to directions in the XY-plane of the Bloch sphere, at relative angle $\pi/3$. Thus this behaviour is **physically realisable** using quantum entanglement, although, as we have seen, it has no realisation by means of a classical source.

[4] More generally, we can consider d-dimensional quantum systems for any positive integer d. A system of n qubits has dimension 2^n. Contextuality emerges already at dimension $d = 3$.

[5] We are ignoring normalisation constants.

Computing the Bell table

Some readers may find it helpful to see in detail how a table such as that in Figure 2 is computed. We shall now explain this. Nothing following this subsection will depend on this material, so it can safely be skipped.

We shall consider spin measurements lying in the equatorial plane of the Bloch sphere, *i.e.* the XY-plane as shown in Figure 4. For such a measurement at an angle ϕ to the X-axis, the Spin Up outcome is specified by the vector $(|\uparrow\rangle + e^{i\phi}|\downarrow\rangle)/\sqrt{2}$, while the Spin Down outcome is specified by $(|\uparrow\rangle + e^{i(\phi+\pi)}|\downarrow\rangle)/\sqrt{2}$. For the X direction itself, we have $\phi = 0$, and these are the vectors $(|\uparrow\rangle + |\downarrow\rangle)/\sqrt{2}$ and $(|\uparrow\rangle - |\downarrow\rangle)/\sqrt{2}$ respectively.

We shall use the measurement in the X direction for Alice's measurement a_1 and Bob's measurement b_1; while a_2 and b_2 will be interpreted by the measurements at angle $\phi = \pi/3$ to the X axis. Note that Alice's measurements are applied to the first qubit of the Bell state, while Bob's measurements are applied to the second qubit.

For example, consider the following situation: Alice performs the measurement a_1 on the first qubit and observes the outcome 0 (Spin Up), while Bob performs the measurement b_2 on the second qubit and observes outcome 1 (Spin Down). This corresponds to the cell in row 2, column 3 of the table in Figure 2. This event is represented by taking the tensor product of the vectors representing the outcomes for the local measurements by Alice and Bob on their qubits:

$$\frac{|\uparrow\rangle + |\downarrow\rangle}{\sqrt{2}} \otimes \frac{|\uparrow\rangle + e^{i4\pi/3}|\downarrow\rangle}{\sqrt{2}} \;=\; \frac{|\uparrow\uparrow\rangle + e^{i4\pi/3}|\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle + e^{i4\pi/3}|\downarrow\downarrow\rangle}{2}.$$

Call this vector M. The probability of observing this event when performing the joint measurement (a_1, b_2) on the Bell state $B = (|\uparrow\uparrow\rangle + |\downarrow\downarrow\rangle)/\sqrt{2}$ is given, using the Born rule, by $|\langle B|M\rangle|^2$. Since the vectors $|\uparrow\uparrow\rangle$, $|\uparrow\downarrow\rangle$, $|\downarrow\uparrow\rangle$, $|\downarrow\downarrow\rangle$ are pairwise orthogonal, this simplifies to

$$\left|\frac{1 + e^{i4\pi/3}}{2\sqrt{2}}\right|^2 \;=\; \frac{|1 + e^{i4\pi/3}|^2}{8}.$$

Using the Euler identity $e^{i\theta} = \cos\theta + i\sin\theta$, we have

$$|1 + e^{i\theta}|^2 \;=\; 2 + 2\cos\theta.$$

Hence

$$\frac{|1 + e^{i4\pi/3}|^2}{8} \;=\; \frac{2 + 2\cos(4\pi/3)}{8} \;=\; \frac{1}{8},$$

the value given in the table in Figure 2. The other entries can be computed similarly.

	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$
(a_1, b_1)	1			
(a_1, b_2)	0			
(a_2, b_1)	0			
(a_2, b_2)				0

Figure 7: The Hardy Paradox

Summary

More broadly, we can say that this shows that quantum mechanics predicts correlations which exceed those which can be achieved by any classical mechanism. This is the content of **Bell's theorem** [15], a famous result in the foundations of quantum mechanics, and in many ways the starting point for the whole field of quantum information. Moreover, these predictions have been confirmed by many experiments which have been performed [11, 10].

4 The "Hardy Paradox"

We shall now see how the same phenomena manifest themselves in a stronger form, which highlights a direct connection with logic. Consider the table in Figure 7.

This table depicts the same kind of scenario we considered previously. However, the entries are now either 0 or 1. The idea is that a 1 entry represents a positive probability. Thus we are distinguishing only between **possible** (positive probability) and **impossible** (zero probability). In other words, the rows correspond to the **supports** of some (otherwise unspecified) probability distributions. Moreover, only four entries of the table are filled in. Our claim is that just from these four entries, referring only to the supports, we can deduce that there is no classical explanation for the behaviour recorded in the table. Moreover, this behaviour can again be realised in quantum mechanics, yielding a stronger form of Bell's theorem, due to Lucien Hardy [20].[6]

[6]For a detailed discussion of realisations of the Bell and Hardy models in quantum mechanics, see Section 7 of [2]. Further details on the Hardy construction can be found in [20, 30].

4.1 What Do "Observables" Observe?

Classically, we would take the view that physical observables directly reflect properties of the physical system we are observing. These are objective properties of the system, which are independent of our choice of which measurements to perform — of our **measurement context**. More precisely, this would say that for each possible state of the system, there is a function λ which for each measurement m specifies an outcome $\lambda(m)$, **independently of which other measurements may be performed**. This point of view is called **non-contextuality**, and may seem self-evident. However, this view is **impossible to sustain** in the light of our **actual observations of (micro)-physical reality**.

Consider once again the Hardy table depicted in Figure 7. Suppose there is a function λ which accounts for the possibility of Alice observing value 0 for a_1 and Bob observing 0 for b_1, as asserted by the entry in the top left position in the table. Then this function λ must satisfy

$$\lambda : a_1 \mapsto 0, \quad b_1 \mapsto 0.$$

Now consider the value of λ at b_2. If $\lambda(b_2) = 0$, then this would imply that the event that a_1 has value 0 and b_2 has value 0 is possible. However, **this is precluded** by the 0 entry in the table for this event. The only other possibility is that $\lambda(b_2) = 1$. Reasoning similarly with respect to the joint values of a_2 and b_2, we conclude, using the bottom right entry in the table, that we must have $\lambda(a_2) = 0$. Thus the only possibility for λ consistent with these entries is

$$\lambda : a_1 \mapsto 0, \quad a_2 \mapsto 0, \quad b_1 \mapsto 0, \quad b_2 \mapsto 1.$$

However, this would require the outcome $(0, 0)$ for measurements (a_2, b_1) to be possible, and this is **precluded** by the table.

We are thus forced to conclude that the Hardy models are contextual. Moreover, we can say that they are contextual in a logical sense, stronger than the probabilistic form we saw with the Bell tables, since we only needed information about possibilities to infer the contextuality of this behaviour.

5 Mathematical Structure of Possibility Tables

Consider again a table such as

	$(0,0)$	$(1,0)$	$(0,1)$	$(1,1)$
(a_1, b_1)	1	1	1	1
(a_2, b_1)	0	1	1	1
(a_1, b_2)	0	1	1	1
(a_2, b_2)	1	1	1	0

Let us anatomise the structure of this table. There are **measurement contexts**

$$\{a_1, b_1\}, \quad \{a_2, b_1\}, \quad \{a_1, b_2\}, \quad \{a_2, b_2\}.$$

These are the possible combinations of measurements which can be made together, yielding the directly accessible empirical observations.[7] Each measurement has possible outcomes 0 or 1. More generally, we write O for the set of possible outcomes. Thus for example the matrix entry at row (a_2, b_1) and column $(0, 1)$ indicates the **event**

$$\{a_2 \mapsto 0,\ b_1 \mapsto 1\}.$$

The set of events relative to a context C is the set of functions O^C. Each row of the table specifies a **Boolean distribution** on events O^C for a given choice of measurement context C. Such a Boolean distribution is just a non-empty set of events.

Mathematically, this defines a **presheaf**. We have:

- A set of measurements X (the "space"). In our example, $X = \{a_1, a_2, b_1, b_2\}$.
- A family of subsets of X, the **measurement contexts** (a "cover"). In our example, these are

  $$\{a_1, b_1\}, \quad \{a_2, b_1\}, \quad \{a_1, b_2\}, \quad \{a_2, b_2\}$$

  as already discussed.
- To each such set C a boolean distribution (finite non-empy subset) on **local sections** $s : C \to O$, where O is the set of **outcomes**. Each row of the above table specifies such a distribution. Note that this notion of distribution generalises naturally to distributions valued in a **commutative semiring**. We assume that the distributions have finite support, and are normalised (have total weight 1). In our case, we are using the idempotent semiring of the booleans. We use the notation $\mathcal{D}_{\mathsf{B}}(X)$ for the set of boolean distributions on a set X.

[7] In quantum mechanics, these correspond to compatible families of observables.

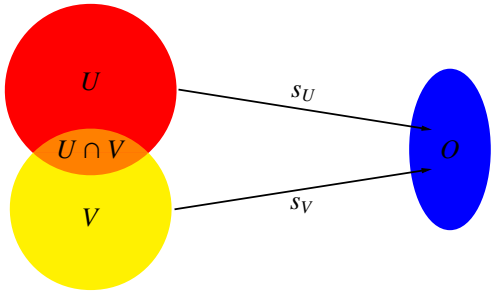


Figure 8: Gluing functions

Note that, if we use the semiring of non-negative reals instead, we obtain probability distributions with finite support.

- A distribution on C restricts to $C' \subseteq C$ by pointwise restriction of the local sections. More precisely, given such a distribution d on O^C, we restrict it to C' by defining, for $s \in O^{C'}$:

$$d|_{C'}(s) \;=\; \sum_{s' \in O^C, s'|_{C'} = s} d(s').$$

This definition makes sense for any semiring. In the boolean case, where addition is disjunction, it can be expressed equivalently as **projection**, where we think of the distribution as a finite set:

$$d|_{C'} \;=\; \{s|_{C'} \;:\; s \in d\}.$$

In the probability case, it gives the usual notion of **marginalisation**.

These local sections correspond to the directly observable **joint outcomes** of **compatible measurements**, which can actually be performed jointly on the system. The different sets of compatible measurements correspond to the different contexts of measurement and observation of the physical system. The fact that the behaviour of these observable outcomes cannot be accounted for by some context-independent global description of reality corresponds to the geometric fact that these local sections cannot be glued together into a **global section**.

For a picture of the familiar and simple situation of gluing functions together, consider the diagram in Figure 8. If $s_U|_{U\cap V} = s_V|_{U\cap V}$, they can be glued to form

$$s : U \cup V \longrightarrow O$$

branch-name	account-no	customer-name	balance
Cambridge	10991-06284	Newton	£2,567.53
Hanover	10992-35671	Leibniz	€11,245.75
...	...	...	...

Figure 9: A relation table

such that $s|_U = s_U$ and $s|_V = s_V$.

In geometric language, the Hardy paradox corresponds to the fact that there is a **local section** which cannot be extended to a **global section** which is compatible with the family of boolean distributions. In other words, the space of **local possibilities** is sufficiently logically 'twisted' to **obstruct** such an extension.

The quantum phenomena of **non-locality** and **contextuality** correspond exactly to the existence of obstructions to global sections in this sense. This geometric language is substantiated by the results in [7], which show that **sheaf cohomology** can be used to characterise these obstructions, and to witness contextuality in a wide range of cases.

This geometric picture and the associated methods can be applied to a wide range of situations in classical computer science, which do not seem to have anything in common with the quantum realm. In particular, as we shall now see, there is an isomorphism between the formal description we have given for the quantum notions of non-locality and contextuality, and basic definitions and concepts in relational database theory.

6 Relational Databases and Bell's Theorem

Consider an example of a table in a relational database, as in Figure 9.

Let us anatomise such tables:

- The columns are determined by a set A of **attributes**. Assume $A \subset \mathcal{A}$ for some global set $\mathcal{A}$ specified by the database schema.
- For each attribute a, there is a possible set of **data values** D_a. For simplicity, we collect these into a global set $D = \bigsqcup_{a \in \mathcal{A}} D_a$.
- An **A-tuple** is specified by a function $t : A \to D$.
- A **relation instance** or **table** of schema A is a set of A-tuples.

- A **database schema** is given by a family $\Sigma = \{A_1, \ldots, A_k\}$ of finite subsets of $\mathcal{A}$.
- A database **instance** of schema Σ is given by a family of relation instances $\{R_i\}$ where R_i is of schema A_i.

Does this look familiar? In fact, it is straightforward to express this structure in the language of presheaves:

- An A-tuple t is just a local section over A: $t \in D^A$.
- A relation table R of schema A is a boolean distribution on A-tuples:

$$R \in \mathcal{D}_{\mathsf{B}}(D^A).$$

- Note that if $A \subseteq B$, then restriction is just **projection**. For $R \in \mathcal{D}_{\mathsf{B}}(D^B)$

$$R|_A := \{t|_A \ : \ t \in R\}.$$

- We can regard a schema Σ as a cover of $\mathcal{A}$.
- A database instance of schema Σ is a family of elements $\{R_A\}_{A \in \Sigma}$.
- The compatibility condition for an instance is **projection consistency**:

$$R_A|_{A \cap B} = R_B|_{A \cap B}$$

  means that the two relations have the same projections onto their common set of attributes.

6.1 Universal Relations

A **universal relation** for an instance $\{R_A \ : \ A \in \Sigma\}$ of a schema Σ is a relation $R \in \mathcal{D}_{\mathsf{B}}(D^{\mathcal{A}})$ such that, for all $A \in \Sigma$:

$$R|_A = R_A.$$

Thus it is a relation defined on the whole set of attributes $\mathcal{A}$ from which each of the relations in the instance can be recovered by projection.

This notion, and various related ideas, played an important rôle in early developments in relational database theory; see e.g. [27, 19, 24, 26, 34]. Note that a universal relation instance corresponds exactly to the notion of **global section** for the database instance viewed as a compatible family. (Compatibility is obviously a necessary condition for such an instance to exist).

It is also standard that a universal relation need not exist in general, and even if it exists, it need not be unique. There is a substantial literature devoted to the issue of finding conditions under which these properties do hold.

There is a simple connection between universal relations and lossless joins.

Proposition. Let $(R_1, \ldots, R_k)$ be an instance for the schema $\Sigma = \{A_1, \ldots, A_k\}$. Define $R := \bowtie_{i=1}^{k} R_i$. Then a universal relation for the instance exists if and only if $R|_{A_i} = R_i$, $i = 1, \ldots, k$, and in this case R is the largest relation in $\mathcal{R}(\bigcup_i A_i)$ satisfying the condition for a global section. □

We can summarise the striking correspondence we have found between the realms of quantum contextuality and database theory in the following dictionary:

Relational databases	measurement scenarios
attribute	measurement
set of attributes defining a relation table	compatible set of measurements
database schema	measurement cover
tuple	local section (joint outcome)
relation/set of tuples	boolean distribution on joint outcomes
universal relation instance	global section/hidden variable model
acyclicity	Vorob'ev condition [35]

This dictionary goes beyond what we have discussed so far. The last entry concerns Vorob'ev's Theorem [35], a remarkable result motivated by game theory which provides a necessary and sufficient combinatorial condition on a set cover or hypergraph (formulated equivalently in terms of abstract simplicial complexes) such that any compatible family of probability distributions over this cover can be glued together into a global section — a joint distribution on the whole set of vertices which marginalises to yield the given distribution over each simplex. This condition is equivalent to the well-studied notion of **acyclicity** of database schemes [13, 25].

It seems that there is considerable scope for taking these connections and common structures further. For example, we can consider probabilistic databases, and more generally distributions valued in semirings. See [1] for a more detailed discussion.

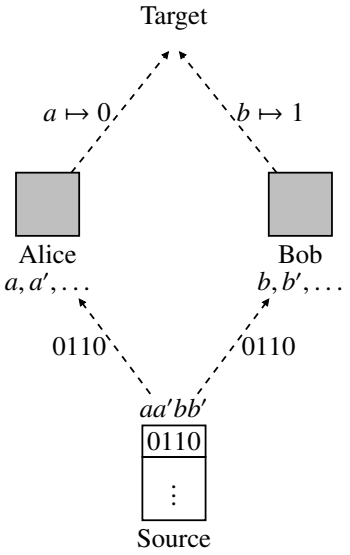


Figure 10: The Mermin instruction set picture

6.2 Hidden variables and all that

We mentioned hidden variable models in the above table, but have not otherwise done so in this article. Traditionally, such models have played a leading rôle in discussions of quantum non-locality and contextuality. Essentially, a local hidden-variable model is what we called a "classical source" in Section 3. Indeed, a standard way of picturing such a model, due to David Mermin [29], is shown in Figure 10. This is essentially the same picture as Figure 3. Mermin calls the hidden variables "instruction sets"; these correspond exactly to the global assignments we have been discussing, which can be considered as canonical forms of hidden variables. It is shown in [3, Theorem 8.1] that these are equivalent to the more general forms of hidden variable models which have been considered in the literature.

6.3 Contextual semantics

Why do such similar structures arise in such apparently different settings? The phenomenon of contextuality is pervasive. Once we start looking for it, we can find it everywhere! Examples already considered include: physics [3], computation [5], and natural language [8].

This leads to what we may call the **Contextual semantics hypothesis**: we can find common mathematical structure in all these diverse manifestations, and develop a widely applicable theory.

7 Kochen-Specker Models

We now return to quantum mechanics, and discuss another fundamental result, the Kochen-Specker theorem [23].[8] This result shows the contextuality of quantum mechanics in an even stronger form than Bell's theorem, in the sense that the argument is independent of any particular quantum state. Whereas our arguments for the Bell and Hardy theorems hinged on realising contextual behaviours using certain entangled quantum states, the Kochen-Specker argument rests on properties of certain families of measurements which hold for **any** quantum state.

There is, however, a trade-off. Whereas the conclusion of the Kochen-Specker theorem is stronger than that of Bell's theorem, its assumptions are also stronger, in that it assumes (for a contradiction) non-contextuality for measurements **in general**. By contrast, Bell's theorem applies to a particular class of measurement scenarios where Alice and Bob are spacelike separated; in these situations, the assumption of non-contextuality is supported by relativistic considerations, which imply that there can be no direct causal influence by the measurements on each other.

The stronger form of state-independent contextuality given by the Kochen-Specker theorem is nevertheless of great interest, and has been the subject of a number of recent experimental verifications [12, 22]. It is also a topic of current interest to develop methods for exploiting contextuality as a resource in quantum information, extending what has been done for non-locality. A feature of our sheaf-theoretic framework, as described in Section 5, is that it provides a unified setting for Bell's theorem, the Kochen-Specker theorem, and other results relating to non-locality and contextuality.

We recall the general setting discussed in Section 5. We have a set X of measurement labels, and a family $\mathcal{U}$ of subsets of X — a "measurement cover". The sets $C \in \mathcal{U}$ are the **measurement contexts**; those combinations of measurements which can be performed together. Formally speaking, $(X, \mathcal{U})$ is just a hypergraph.

For convenience we fix our set of outcomes as $O = \{0, 1\}$. Given $C \in \mathcal{U}$, we say that $s \in O^C$ satisfies the **KS property** if $s(x) = 1$ for exactly one $x \in C$. The **Kochen-Specker model** over $(X, \mathcal{U})$ is defined by setting d_C, for each $C \in \mathcal{U}$, to be the set of all $s \in O^C$ which satisfy the KS property. Note that the model is uniquely determined once we have given $(X, \mathcal{U})$.

[8] Since Bell independently proved a version of this result [16], it is often called the Bell-Kochen-Specker theorem.

Note that, if we regard the elements of X as propositional variables, we can think of $s \in O^C$ as a truth-value assignment.[9] Then the KS property for an assignment s is equivalent to s satisfying the following formula:

$$\mathsf{ONE}(C) \;:=\; \bigvee_{x \in C} (x \;\wedge \bigwedge_{x' \in C \setminus \{x\}} \neg x')$$

We say that the Kochen-Specker model over $(X, \mathcal{U})$ is contextual if there is no global assignment $s : X \to O$ on the whole set of variables X such that $s|_C \in d_C$ for all $C \in \mathcal{U}$. Equivalently, we can say that the model is contextual if the formula

$$\bigwedge_{C \in \mathcal{U}} \mathsf{ONE}(C)$$

is unsatisfiable.[10]

It is interesting to compare this with the property of the Hardy models discussed in Section 5. As we saw there, the contextuality property exhibited by these models was that there was a local section in the support at some $C \in \mathcal{U}$ which was not extendable to a global assignment on X which was compatible with the support. By contrast, the form of contextuality we are considering here is much stronger; that **there is no global assignment at all** which is consistent with the support. In fact, the Hardy models do not satisfy this stronger property.

The simplest example of a contextual Kochen-Specker model is the triangle, *i.e.* the cover

$$\{a, b\}, \{b, c\}, \{a, c\}$$

on $X = \{a, b, c\}$. For a more elaborate example, consider the set $X = \{m_1, \ldots, m_{18}\}$, and the measurement cover $\mathcal{M}$ whose elements are the columns of the following table:

m_1	m_1	m_8	m_8	m_2	m_9	m_{16}	m_{16}	m_{17}
m_2	m_5	m_9	m_{11}	m_5	m_{11}	m_{17}	m_{18}	m_{18}
m_3	m_6	m_3	m_7	m_{13}	m_{14}	m_4	m_6	m_{13}
m_4	m_7	m_{10}	m_{12}	m_{14}	m_{15}	m_{10}	m_{12}	m_{15}

How we do we show that a model such as this is contextual? We shall give a combinatorial criterion on $(X, \mathcal{U})$ which can be used for most of the examples which have appeared in the literature.

[9] Interpreting 1 as true and 0 as false.

[10] Note that in the general case where O is some finite set, this becomes a constraint satisfaction problem. Contextuality means that the problem has no solution.

For each $x \in X$, we define

$$\mathcal{U}(x) := \{C \in \mathcal{U} \ : \ x \in C\}.$$

Proposition [3, Proposition 7.1]. If the Kochen-Specker model on $(X, \mathcal{U})$ is non-contextual, then every common divisor of $\{|\mathcal{U}(x)| \ : \ x \in X\}$ must divide $|\mathcal{U}|$. □

Applying this to the above example, we note that the cover $\mathcal{M}$ has 9 elements, while each element of X appears in two members of $\mathcal{M}$. Thus the Kochen-Specker model on $(X, \mathcal{M})$ is contextual.

Quantum representations

What do these combinatorial questions have to do with quantum mechanics? A contextual Kochen-Specker model $(X, \mathcal{U})$ gives rise to a quantum mechanical witness of contextuality whenever we can label X with unit vectors in $\mathbb{R}^n$, for some fixed n, such that $\mathcal{U}$ consists exactly of those subsets C of X which form orthonormal bases of $\mathbb{R}^n$. The point of our example $\mathcal{M}$ above is that it is possible to label the 18 elements of X with vectors in $\mathbb{R}^4$ such that the four-element subsets in $\mathcal{M}$ are orthogonal [18]. This yields one of the most economical known quantum witnesses for contextuality.[11]

To connect this directly to quantum measurement, note that such a family of vectors can be used to define corresponding measurements, such that the measurements corresponding to orthogonal sets are compatible, and moreover for any quantum state $|\psi\rangle$, the support of the distribution on outcomes induced by performing this joint measurement on $|\psi\rangle$ will satisfy the KS property. Thus contextuality of the model yields a state-independent witness of quantum contextuality. For a detailed discussion of this point, see Section 9.2 of [3].

The smallest dimension for which contextuality witnesses appear in this form is $n = 3$. Currently, the smallest known Kochen-Specker model providing a contextuality witness in dimension 3 has 31 vectors [33]. Computational methods have established a lower bound of 18 [9].

8 Discussion and Further Reading

One aim of this paper has been to present some central concepts of quantum information and foundations in a form which will be accessible to computer scientists, in particular those with an interest in logical and structural methods. At the same time, we have also aimed to provide an introduction to recent research by

[11] By contrast, the triangle does **not** yield a quantum witness, since orthogonality is a pairwise notion; if all the pairs are orthogonal, the whole set must be also.

the author and a number of colleagues, which aims to use tools which have been developed within computer science logic and semantics to study these quantum notions. This "high-level" approach has led to a number of developments, both within quantum information, and in identifying the same formal structures in a number of classical computational situations; we have seen an example of this in the case of relational database theory.

We shall conclude by discussing some references where the interested reader can find further information, and see these ideas developed in greater depth.[12]

8.1 The sheaf-theoretic approach

As discussed briefly in Section 5, our analysis of non-locality and contextuality uses the mathematical framework of sheaves and presheaves. The issue of finding "local realistic" explanations of correlated behaviour is interpreted geometrically in terms of finding global sections in the sense of sheaf theory. These ideas, and many basic results, are developed in the paper [3] with Adam Brandenburger which laid the basis for this approach.

This leads to a number of developments in quantum information and foundations:

- The sheaf-theoretic language allows a unified treatment of non-locality and contextuality, in which results such as Bell's theorem [15] and the Kochen-Specker theorem [23] fit as instances of more general results concerning obstructions to global sections. In recent work [28], it has been shown how this framework can be used to **transform** contextuality scenarios into non-locality scenarios.

- A hierarchy of degrees of non-locality or contextuality is identified in [3]. This explains and generalises the notion of "inequality-free" or "probability-free" non-locality proofs, and makes a strong connection to logic, as developed in [2]. This hierarchy is lifted to a novel classification of multipartite entangled states, leading to some striking new results concerning multipartite entanglement, which is currently poorly understood. These results will appear in forthcoming joint publications with Carmen Constantin and Shenggang Ying.

- The obstructions to global sections witnessing contextuality are characterised in terms of sheaf cohomology in [7] with Shane Mansfield and Rui Barbosa, and a range of examples are treated in this fashion.

[12] The papers by the author which are referenced can be found at `arXiv.org`.

- A striking connection between no-signalling models and global sections with signed measures ("negative probabilities") is established in [3]. An operational interpretation of such negative probabilities, involving a signed version of the strong law of large numbers, is developed in [4].

8.2 Logical Bell inequalities

The discussion in Section 3 is based on [6]. Bell inequalities are a central technique in quantum information. In [6] with Lucien Hardy, a general notion of "logical Bell inequality", based on purely logical consistency conditions, is introduced, and it is shown that every Bell inequality (*i.e.* every inequality satisfied by the "local polytope") is equivalent to a logical Bell inequality. The notion is developed at the level of generality of [3], and hence applies to arbitrary contextuality scenarios, including multipartite Bell scenarios and Kochen-Specker configurations.

8.3 Contextual semantics in classical computation

We discussed the isomorphism between the basic concepts of quantum contextuality and those of relational database theory in Section 6. A number of other connections have been studied:

- In [2] connections between non-locality and logic are emphasised. A number of natural complexity and decidability questions are raised in relation to non-locality.

- Our discussion of the Hardy paradox in Section 5 showed that the key issue was that a local section (assignment of values) could not be extended to a global one consistently with some constraints (the "support table"). This directly motivated some joint work with Georg Gottlob and Phokion Kolaitis [5], in which we studied a refined version of **constraint satisfaction**, dubbed "robust constraint satisfaction", in which one asks if a partial assignment of a given length can always be extended to a solution. The tractability boundary for this problem is delineated in [5], and this is used to settle one of the complexity questions posed in [2].

- Application of the contextual semantics framework to natural language semantics is initiated in [8] with Mehrnoosh Sadrzadeh. In this paper, a basic part of the Discourse Representation Structure framework [21] is formulated as a presheaf, and the gluing of local sections into global ones is used to represent the resolution of anaphoric references.

Further connections and applications of contextual semantics are currently being studied, and it seems likely that more will be forthcoming.

References

[1] S. Abramsky. Relational databases and Bell's theorem. In V. Tannen, L. Wong, L. Libkin, W. Fan, W.C. Tan, and M. Fourman, editors, *In Search of Elegance in the Theory and Practice of Computation: Essays Dedicated to Peter Buneman*, pages 13–35. Springer, 2013.

[2] S. Abramsky. Relational hidden variables and non-locality. *Studia Logica*, 101(2):411–452, 2013.

[3] S. Abramsky and A. Brandenburger. The sheaf-theoretic structure of non-locality and contextuality. *New Journal of Physics*, 13(2011):113036, 2011.

[4] S. Abramsky and A. Brandenburger. An operational interpretation of negative probabilities and no-signalling models. In F. van Breugel, E. Kashefi, C. Palamidessi, and J. Rutten, editors, *Horizons of the Mind: A Tribute to Prakash Panagaden*, pages 59–75. Springer, 2014.

[5] S. Abramsky, G. Gottlob, and P.G. Kolaitis. Robust constraint satisfaction and local hidden variables in quantum mechanics. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, pages 440–446. AAAI Press, 2013.

[6] S. Abramsky and L. Hardy. Logical Bell inequalities. *Physical Review A*, 85(6):062114, 2012.

[7] S. Abramsky, S. Mansfield, and R.S. Barbosa. The cohomology of non-locality and contextuality. *Electronic Proceedings in Theoretical Computer Science*, 95:1–15, 2012.

[8] S. Abramsky and M. Sadrzadeh. Semantic Unification: A sheaf theoretic approach to natural language. In C. Casadio, B. Coecke, M. Moortgat, and P.J. Scott, editors, *Categories and Types in Logic, Language, and Physics, A Festshrift for Jim Lambek*, volume 8222 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2014.

[9] F. Arends, J. Ouaknine, and C.W. Wampler. On searching for small Kochen-Specker vector systems. In *Graph-Theoretic Concepts in Computer Science*, pages 23–34. Springer, 2011.

[10] A. Aspect. Bell's inequality test: more ideal than ever. *Nature*, 398(6724):189–190, 1999.

[11] A. Aspect, J. Dalibard, and G. Roger. Experimental test of Bell's inequalities using time-varying analyzers. *Physical review letters*, 49(25):1804, 1982.

[12] H. Bartosik, J. Klepp, C. Schmitzer, S. Sponar, A. Cabello, H. Rauch, and Y. Hasegawa. Experimental test of quantum contextuality in neutron interferometry. *Physical Review Letters*, 103(4):40403, 2009.

[13] C. Beeri, R. Fagin, D. Maier, and M. Yannakakis. On the desirability of acyclic database schemes. *Journal of the ACM (JACM)*, 30(3):479–513, 1983.

[14] J. Bell. An Exchange on Local Beables. *Dialectica*, 39:85–96, 1985.

[15] J.S. Bell. On the Einstein-Podolsky-Rosen paradox. *Physics*, 1(3):195–200, 1964.

[16] J.S. Bell. On the problem of hidden variables in quantum mechanics. *Reviews of Modern Physics*, 38(3):447–452, 1966.

[17] A. Brandenburger and N. Yanofsky. A classification of hidden-variable properties. *Journal of Physics A: Mathematical and Theoretical*, 41:425302, 2008.

[18] A. Cabello, J.M. Estebaranz, and G. García-Alcaine. Bell-Kochen-Specker theorem: A proof with 18 vectors. *Physics Letters A*, 212(4):183–187, 1996.

[19] R. Fagin, A.O. Mendelzon, and J.D. Ullman. A simplified universal relation assumption and its properties. *ACM Transactions on Database Systems (TODS)*, 7(3):343–360, 1982.

[20] L. Hardy. Nonlocality for two particles without inequalities for almost all entangled states. *Physical Review Letters*, 71(11):1665–1668, 1993.

[21] H. Kamp and U. Reyle. *From Discourse to Logic: Introduction to model-theoretic semantics of natural language, formal logic and discourse representation theory*. Springer, 1993.

[22] G. Kirchmair, F. Zähringer, R. Gerritsma, M. Kleinmann, O. Gühne, A. Cabello, R. Blatt, and CF Roos. State-independent experimental test of quantum contextuality. *Nature*, 460(7254):494–497, 2009.

[23] S. Kochen and E.P. Specker. The problem of hidden variables in quantum mechanics. *Journal of Mathematics and Mechanics*, 17(1):59–87, 1967.

[24] H.F. Korth, G.M. Kuper, J. Feigenbaum, A. Van Gelder, and J.D. Ullman. SYSTEM/U: a database system based on the universal relation assumption. *ACM Transactions on Database Systems (TODS)*, 9(3):331–347, 1984.

[25] D. Maier. *The Theory of Relational Databases*. Computer Science Press Rockville, 1983.

[26] D. Maier and J.D. Ullman. Maximal objects and the semantics of universal relation databases. *ACM Transactions on Database Systems (TODS)*, 8(1):1–14, 1983.

[27] D. Maier, J.D. Ullman, and M.Y. Vardi. On the foundations of the universal relation model. *ACM Transactions on Database Systems (TODS)*, 9(2):283–308, 1984.

[28] S. Mansfield and R.S. Barbosa. Extendability in the Sheaf-theoretic Approach: Construction of Bell Models from Kochen-Specker Models. *arXiv preprint arXiv:1402.4827*, 2014.

[29] N.D. Mermin. Quantum mysteries revisited. *Am. J. Phys*, 58(8):731–734, 1990.

[30] N.D. Mermin. Quantum mysteries refined. *American Journal of Physics*, 62:880, 1994.

[31] N.D. Mermin. *Quantum Computer Science*. Cambridge: Cambridge University Press, 2007.

[32] M.Q.C. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

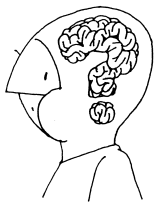
[33] A. Peres. *Quantum theory: concepts and methods*, volume 57. Springer, 1995.

[34] J.D. Ullman. *Principles of Database Systems*. Prentice Hall, 1983.

[35] N.N. Vorob'ev. Consistent families of measures and their extensions. *Theory of Probability & Its Applications*, 7(2):147–163, 1962.

[36] N.S. Yanofsky and M.A. Mannucci. *Quantum Computing for Computer Scientists*, volume 20. Cambridge University Press Cambridge, 2008.

News and Conference Reports



NEWS FROM NEW ZEALAND

BY

C. S. CALUDE

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

1 Scientific and Community News

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

441. C.S. Calude, R. Freivalds and F. Stephan. Deterministic Frequency Pushdown Automata. 09/2013

442. C.S. Calude. Quantum Randomness: From Practice to Theory and Back. 09/2013

443. A.A. Abbott, C.S. Calude and K. Svozil. Value Indefiniteness Is Almost Everywhere. 09/2013

444. C.S. Calude, L. Staiger and F. Stephan. Finite State Incompressible Infinite Sequences. 11/2013

445. A. Nies. Calculus of Cost Functions. 11/2013

446 S. Figueira and A. Nies. Feasible Analysis, Randomness, and Base Invariance. 11/2013

447. K. Wei and M.J. Dinneen, Comparing Two Local Searches in a (1+1) Restart Memetic Algorithm on the Clique Problem. 12/2013
448. C.S. Calude and L. Staiger. Liouville Numbers, Borel Normality and Algorithmic Randomness. 12/2013
449. F. Ferrarotti, S. Hartmann and S. Link. Reasoning about Functional and Full Hierarchical Dependencies Over Partial Relations. 12/2013
450. J. Kontinen, S. Link and J. Väänänen. Independence in Database Relations. 12/2013
451. V.B. Tran Le, S. Link and F. Ferrarotti. Effective Recognition and Visualization of Semantic Requirements by Perfect SQL Samples. 12/2013
452. H. Köhler, U. Leck and S. Link. Possible and Certain SQL Keys. 12/2013
453. S. Böttcher, S. Link and L. Zhang. LECQTER: Learning Conjunctive SQL Queries Through Exemplars. 02/2014
454. S. Hartmann and S. Link. Normal Forms and Normalization for Probabilistic Databases under Sharp Constraints. 02/2014
455. A. Gavruskin, S. Jain, B. Khoussainov and F. Stephan. Graphs Realised by R.E. Equivalence Relations. 01/2014
456. A. Gavruskin, B. Khoussainov and F. Stephan. Reducibilities Among Equivalence Relations Induced by Recursively Enumerable Structures. 01/2014
457. S. Jain, B. Khoussainov, F. Stephan, D. Teng and S. Zou. Semiautomatic Structures. 02/2014
458. A. A. Abbott, C. S. Calude and K. Svozil. On the Unpredictability of Individual Quantum Measurement Outcomes. 03/2014
459. L. Staiger. On the Hausdorff Measure of Regular ω-languages in Cantor Space. 04/2014
460. M. Hannula, J. Kontinen and S. Link. On Independence Atoms and Keys. 04/2014
461. C.S. Calude and N. Poznanović. Free Will and Randomness. 05/2014

2 A Dialogue with Mioara Mugur-Schachter on Information, Quantum mechanics and Probabilities

Professor Mioara Mugur-Schachter, `http://www.mugur-schachter.net` *is a physicist, mathematician and philosopher specialising in quantum mechanics, probability theory, information theory and epistemology. Her PhD Thesis (supervised by Nobel laureate Louis de Broglie) contains the first invalidation of von Neumann's famous proof stating the impossibility of hidden parameters compatible with the quantum mechanical formalism. This result was included in the volume "Etude du caractère complet de la mécanique quantique", (with a Preface by L. de Broglie) published in the collection "Les grands problèmes des sciences", Gauthiers Villars, Paris, 1964, two years before Bell's invalidation.*

Professor Mugur-Schachter has founded the Laboratoire de Mécanique Quantique & Structures de l'Information at the University of Reims, the Centre pour la Synthèse d'une Épistéémologie Formalisée and L'Association pour le Développement de la Méthode de Conceptualisation Relativisée.

CC: You have been born and educated in Romania. Tell us about your time at the University of Bucharest: subjects you studied, professors, general atmosphere.

MM-S: I began by studying mathematics and philosophy (especially logic and psychology). Then I chose to specialise in theoretical physics. For political reasons my studies suffered an interruption that seemed to be fated to be irreversible. But later the events evolved and I finally was allowed to resume my studies. So I graduated with a Master in theoretical physics. My Professors, as I remember them, were very remarkable indeed. Profoundly educated persons, and many among them endowed with genuine originality. The teaching was very thorough. For me however—from a subjective point of view—my student years have been a deeply troubled time about which I prefer not to focus my attention again. The general atmosphere after 1948, as I perceived it, was constantly growing more and more oppressive from a moral point of view.

CC: Your PhD Thesis was elaborated in Bucharest and sent to Louis de Broglie before you came to Paris. How did you choose your subject? Did you have any supervision in Bucharest for this work?

MM-S: During a recent public visit in a town from the South of France, a young man asked how Louis de Broglie had recruited me? I answered that in fact it was me who tried—very hard indeed—to recruit Louis de Broglie.

When I graduated, my former Professor of Atomic Physics, Horia Hulubei (who was a pupil of Jean Perrin, in Paris, and after the war was called back to

Romania to create an Institute of Atomic Physics) obtained for me a position in the team of theoretical physics of the new Institute.

The subject of research assigned to me was to calculate, using the method established by van Vleck, the interaction between three spins using the framework of quantum mechanics. While covering with matrix elements meter-long sheets of paper intended for architectural projects, I constantly suffered from a very disagreeable feeling of not 'understanding' at all what I was calculating in the prescribed way. This was a new feeling. The Newtonian mechanics seemed to me fully intelligible, and also thermodynamics, atomic physics, statistical physics, and even Maxwell's electromagnetism. But in the case of quantum mechanics I simply did not grasp *how the mathematical formalism manages to carry definite meanings*.

In that state of mind, reading a textbook of quantum mechanics translated from Russian I found the assertion that a certain von Neumann had proved a famous theorem stating that 'hidden parameters' that would 'complete' the quantum mechanical formalism, making it intelligible, are impossible. The proof was not given. Immediately I reacted with a mixture of satisfaction and astonishment. I felt happy to learn that other persons also perceived the unintelligibility and they were investigating it. But I was unable to imagine how it could be possible to prove a definitive impossibility. Inside what conceptual-formal environment could such a proof be achieved? Founded upon what assumptions? So I became very eager to examine the proof. I had a friend who worked at the library of the Academy and I convinced him to order an English translation of the German book by von Neumann where the proof was first presented. The book eventually arrived, but its access was restricted to the library basement. Using a trick, I found von Neumann's book and took it home.

During the next months I became an expert in von Neumann's book. Meanwhile the calculus of matrix elements suffered a nearly total stagnation. At the end of the year I was downgraded for not having finished my assignment. On the other hand, I had written in English the first draft of what I thought to be an invalidation of von Neumann's proof.

I then began asking teachers and colleagues to read my work. But it appeared that nobody around was interested in von Neumann's proof. At the same time everybody was *a priori* convinced that it was a 'definitive' result. This was my first collision with the social environment of scientific thought.

Meanwhile I kept improving the text. And when I finally thought it to be achieved I asked Professor Hulubei to do me the enormous favour to send the manuscript by diplomatic courier (correspondence with the West was restricted at that time in Romania) to Louis de Broglie as I learned indirectly that he believed the theorem to be false. Professor Hulubei accepted, though assuring me that I would never receive an answer.

During the period that followed my husband (who was a Professor of resistance of materials at the Polytechnic Institute of Bucharest) decided that both of us give up our professional positions in order to apply for a passport for leaving the country without creating a dangerous and useless small scandal. We knew quite well how illusory was such an action, but we felt that we just had to try. So we coldly put an end to our Romanian 'careers' and left Bucharest to start a long period of uncertainty (it lasted three full years) during which, quasi incognito, we wandered through the country with temporary jobs here and there. Which, unexpectedly, we enjoyed profoundly.

One morning, while we were living on a boat anchored on a void island in the delta of the Danube, where my husband was in charge of the construction of an irrigation system for a rice field, I rather miraculously got a telegram from my parents informing that Professor Hulubei wanted to see me as soon as possible. I left a small note on the boat, traversed swamps in a tractor, caught a train to Bucharest, and at the end of that very day I stood before Professor Hulubei. He said: "Do you know what? Louis de Broglie answered you! And he agrees that you have invalidated von Neumann's proof!". He handed me a very brief letter addressed to 'Mister Misare Mugur-Schächter (I abandoned that precious letter in Romania, like any other hand-written document). In essence, Louis de Broglie's letter said that it was curious to see that two minds so different as his and mine, reached the same conclusion about von Neumann's proof. But since I had taken a logical approach and had genuinely demonstrated the circularity character of the proof, he would be happy if my work could one day become a PhD under his supervision.

From that moment on I nourished only one dream: to manage to arrive in France. In 1962 this dream became true following an unrealistically adventurous detective path to obtain a passport. And in 1964 my PhD Thesis, titled *Étude du caractère complet de la mécanique quantique*[1] was defended at the University of Paris and published by Gauthier Villars in the collection "Les grands problèmes des sciences", in a volume prefaced by Louis de Broglie (`http://www.mugur-schachter.net`). The first part of the volume contains a French version of my initial invalidation (practically unchanged); the second part contains the proposal of an experiment derived from considerations on the quantum theory of measurement and from de Broglie's reinterpretation of quantum mechanics (the experiment has not been realised, but it might be some day).

CC: You arrived in Paris in 1962. Can you reminiscence about your first encounters with Louis de Broglie, the 7th duke de Broglie?

MM-S: As if it were yesterday. We were towards the end of April. I immediately announced my arrival and obtained a "rendez-vous". I was now waiting

[1] *Study of Completness of Quantum Mechanics.*

seated in the hall of the Academy of Science. An usher came and presented a silver tray asking me to put my visit card. I had no card, so I wrote my name on a piece of paper. And a little later Louis de Broglie himself arrived. He greeted me and invited me to follow him.

I shall never forget the instantaneous transition from the ocean of vague and moving inner images that had so long subsisted in my mind regarding the possible scene of my first meeting with Louis de Broglie, to that unique real scene, so radically definite in every detail, that was uncoiling with apodictic evidence. An upright, infinitely distinguished man, in a dark costume and a shirt with broken collar, was there, in front of me, confirming that he accepted me to become his "last student". He was Louis de Broglie, and I was in Paris, France, seated in an office from the Academy of Science.

During the two subsequent years we met practically every Wednesday to discuss a fragment of my work that I had left in his letterbox from Neuilly-sur-Seine, at least two days in advance. He never forgot and never postponed something that he had announced he would do. He never argued with an idea or a way of expressing something. He just stated his opinion. He also meticulously corrected my French. And very discreetly, he constantly helped me in essential ways to settle myself in France. His attitude influenced me profoundly.

CC: What was wrong with von Neumann's proof?

MM-S: It simply was circular. The hypotheses contained the conclusion. The conclusion of 'definitive' (absolute) impossibility of hidden parameters was in fact derived inside the mathematical formulation of quantum mechanics, namely using the particular way of representing probabilities that is specific to Hilbert spaces, not of micro-states. (If micro-states are represented by another mathematical syntax, different from that of Hilbert spaces—as it is indeed the case for the de Broglie-Bohm representation—then the proof ceases to hold.)

But this is not the unique insufficiency of von Neumann's argument. In my Thesis I have brought forth the unacceptable global structure of von Neumann's argument. The inadequacies of this argument overflow abundantly the strictly logical-formal aspects. They leak out into epistemology, method, and usual language. This 'proof' can be regarded as a striking illustration of the extreme difficulty to achieve a wholly and explicitly dominated mathematical representation of a domain of 'physical facts'. Such a representation involves quite essentially operations of various sorts, physical as well as abstract ones; it involves assumptions of various nature, in particular methodological choices and conventions; it involves *aims* of different natures, the aim to know in a precise way, of course, but also other aims that should be all composed under the constraint of a sort of global coherence. What thus comes out is a need of a sort of coherence that cannot be separated from a feeling of beauty, or on the contrary, of ugliness when certain

slopes of it are violated in some unspeakable way. I had tried as much as I was able to bring all these aspects together into one representation and to extract the essence of the whole. But I was very young, and this was my first research.

CC: You have also challenged Wigner's proof on the impossibility of a joint probability of position and momentum compatible with the formalism of quantum mechanics. Is the theorem false as well?

MM-S: I would not say that it is 'false'. I only showed that the asserted conclusion does not follow. I even identified a trivial counterexample and I showed how this counterexample is allowed to arise inside Wigner's construction. As in my experience with von Neumann's proof, as soon as I succeeded to achieve a sufficiently compact variant of this second critical work (which took more than two years and a long preliminary publication) I sent it to Wigner himself. Wigner invited me to visit him in his wooden house in Vermont, for a direct discussion. So I went there. He recommended the work for publication in the *Foundations of Physics*.

CC: What is the "opacity functional of a statistic" and how did you use it for a mathematical unification between the theory of probabilities and Shannon's theory of communication of information?

MM-S: This has been my first constructive work. It is the result of an attempt at explaining why Boltzmann's statistical distribution tied with the Carnot-Clausius definition of physical 'entropy', possesses a mathematical form that is identical with that of Shannon's purely mathematical concept of 'informational entropy'. My motivation came from the seemingly unconceivable fact that this formal identity between two concepts, that are so radically different in their semantic contents, is just a coincidence.

The central idea of the approach has been to construct—inside a Kolmogorov probability space—a pure mathematical definition of the probability of realisation of a given statistical distribution of the elementary events of the space.

Consider a Kolmogorov space that contains a universe of elementary events and a probability law on it. Consider a very long but finite random sequence of elementary events from this universe. The elementary events emerge inside this sequence in a certain order, and each elementary event possesses a certain relative frequency inside the sequence, which defines a certain 'statistical structure' of the sequence. It is obvious that: a) a given statistical structure can arise for various lengths of the sequence, b) for a fixed length, not any statistical structure is possible.

Two questions can be examined. The first one is: What is the expression of the probability for the realisation of a sequence with a given statistical structure, abstraction being made of the order and length? We have proved that the Kolmogorov expression of the limit of the ratio between the probability of the

sequence considered and the length of the sequence, is equal to the difference between two terms, the Shannon-entropy of the probability law from the considered Kolmogorov probability space, and 'the modulation of the probability law by the fixed statistical structure'. I called this difference the *opacity* of the (fixed) statistical structure of the sequence of elementary events with respect to the probability law of the Kolmogorov probability space.

The second question is: How does this probability evolve when the length of the considered sequence tends to inifity? The answer is: If the length of the sequence of elementary events tends to infinity, then the opacity functional satisfies the weak law of large numbers.

The opacity functional realises an abstract unification between the probabilistic and the informational approaches. This unification permits to construct deductively inside the theory of probability, the identity of form between, on the one hand, the concept of physical statistical entropy introduced by Carnot, Clausius and Boltzmann, and on the other hand, Shannon's concept of informational entropy of the probability law assigned to the signs from an alphabet of an information-source regarded as elementary events. The formal identity can now be clearly distinguished from the semantic specificities (physical, informational), while the relations between formalism and semantics are clearly defined in each case.

CC: Your work on "formalised epistemology" was characterised by Jean-Paul Baquiast, editor of "Automates intelligents", as *a revolution in the way of representing the processes by which we acquire knowledge....* Can you describe your method of "relativized conceptualisation"?

MM-S: The method of relativized conceptualisation (MRC) is similar to a grammar or 'a formal logic', that give syntactic rules for making use of a set of signs. But instead of dealing with this or that 'language' or symbolic way of constructing 'rational truths' (conclusions established deductively), MRC concerns the whole of human processes of conceptualisation: it is a general syntax for normalised creation of consensual knowledge. I say 'normalised' in the sense of 'methodologies': indeed, like any method, MRC is organically tied with aims, and MRC major aim is expressed in the following: The system of norms organised by MRC assures the realisation of 'safe scientific knowledge', that is, of communicable and consensual knowledge where any possibility of emergence of false problems or of paradoxes is excluded by construction.

MRC establishes a bridge from my initial investigations—exclusively critical and achieved with reference to norms that worked only implicitly and were devoid of generality—to a quite general and explicitly organised methodological framework.

Let me detail a little more. Any piece of knowledge that can be communi-

cated without resource restrictions (space or/and time) is a 'description' (pointing toward something restricts to co-presence on a same place at the same time, so it does not give a 'description'). What is not 'described' cannot be communicated in unrestricted ways, even if it is known by someone. So, MRC is a method of scientific and safe description.

MRC is constructed in a deductive way and uses the current natural logic. It involves 1 postulate, 3 principles, 1 convention, 22 main definitions and 6 proved "propositions". That is all.

A 'description' consists of some 'qualification'—in a certain generalised adjectival sense—of some 'entity-to-be-qualified'. According to MRC-norms, any description has to be realised within a previously defined 'epistemic referential (G, V) which consists of an explicitly defined operation of generation G of the object-entity oe_G to be 'qualified' ('described'), and a concept denoted V that is called a view which consists of a structure thatrealises precisely the desired sort of qualification.

The operation of generation G can consist of just selecting a pre-existing entity and assigning it the role of object oe_G for future qualifications; but G can also be a radically creative operation (as it happens indeed for a free micro-state to be studied according to quantum mechanics).

On the basis of very careful analyses, it appears that in order to avoid any arbitrary a priori restriction it is unavoidable to posit—even if a posteriori this posit is modified—that the object-entity oe_G stays in a one-to-one relation with its operation of generation G (this is expressed by the index G from the denotation oe_G). This a priori posit constitutes inside MRC an essential methodological decision.

The basic nature of V is analogous to that of a grammatical predicate. But its structure is far more complex, precise and general. A view V consists of a finite union $V = \bigcup_g V_g$ of aspect-views V_g. Each aspect-view V_g introduces a freely chosen 'semantic dimension g' (for instance the trivial one indicated by the word 'colour', but also any other more unusual or sophisticated one) endowed with a finite set of 'values' denoted g_k, where g is fixed and k varies in a finite set (for instance, for the semantic dimension of 'colour', one could place just green, red and yellow, or these and also other 15 colours, etc.). An aspect-view V_g is 'blind' with respect to the semantic dimensions different from its own, as well as with respect to any value g_k with which it has not been endowed by its definition: it is a filter. Moreover, each aspect-view V_g states explicitly (a) what conceptual-physical operations constitute an act of 'examination by V_g'; (b) what is the observable result of a given act of examination, and (c) how this result is translated into a value g_k of V_g (when oe_G is not directly perceivable, this requirement is highly non-trivial).

For the sake of effectiveness in the sense of computability, MRC operates with operationally specified entities using finite constructions.

The relativised genesis of any MRC description induces a definite global struc-

ture for the whole evolving volume. This structure possesses the character of a network of chains of increasing complexity, subject to explicit rules of mutual connection. Each relative description from this network reproduces the same basic epistemological structure $D = \{(g_k)\}$.

In the framework of MRC classical concepts and theories get a new form.

- The classical logic corresponds to a 'genetic relativized logic' that entails a calculus with relative descriptions.
- Classical probabilities correspond to relativised genetic probabilities.
- Genetic logic and genetic probabilities become essentially unified.
- Shannon's theory of communication of information, which by construction does not talk about the meaning of information, becomes relativised when it is embedded into the relativized theory of probabilities; some meaning can emerge.
- The MRC 'complexity' can be expressed by a set of relativised numerical 'measures' established by measurements.
- The concept of time acquires an explicit bi-dimensional representation.

New applications of MRC are developed. For example, a relativized concept of 'system' was constructed in H. Boulouet's Ph.D. Thesis *Relativized Systems Theory* to be submitted to the University of Valencienne (2014).

Furthermore, all classical disciplines are constructed and presented as if the descriptions 'mirror' things and facts that pre-exist quite independently of the model (even Wittgenstein's extraordinary analyses do not clearly challenge this conception). In contrast, MRC is explicitly founded on transferred descriptions. I dare assert that MRC is the first scientific general method of deliberate human conceptualisation.

CC: In which way did you recently collaborate with Giuseppe Longo, an expert in computability theory and discrete mathematics, areas seemingly far away from your main interests? Is this an indication that quantum physics might benefit from an interaction with these areas?

MM-S: I think so. For historical reasons, the beginnings of quantum mechanics have been marked by contributions expressed in terms of continuous mathematics; but also of contributions expressed in algebraic terms. I believe that in the future a discrete and finite, algebraic approach will predominate.

And I think the same is true for probabilities. (The opacity functional can be relativised and discretised.)

Anyhow, MRC is quite essentially finite, so discrete, by construction. With MRC I solved, I think, a major (though rarely discussed) difficulty of the classical probabilistic conceptualisation (see my paper "On the concept of probability", *Mathematical Structures in Computer Science*, special issue on "Randomness, Statistics and Probability", 2014 (in press)). Namely, the lack of a general procedure for constructing the numerical distribution of probability to be used in a factual situation that is generally considered to be probabilistic. I called this difficulty "Kolmogorov's aporia" because starting from 1983 Kolmogorov himself denounced this startling and scandalous situation. For example, in the paper "Combinatorial foundations of information theory and the calculus of probabilities", *Russia Mathematical Surveys*, 38 (1983) 29–40, Kolmogorov says:

> The applications of probability theory can be put on a uniform basis. It is always a matter of consequences of hypotheses about the impossibility of reducing in a way or another the complexity of the descriptions of the objects in question. Naturally this approach to the matter does not prevent the development of probability theory as a branch of mathematics being a special case of general measure theory.

The MRC solution to Kolmogorov's aporia consists of an explicit finite procedure for constructing, in a given factual probabilistic situation, the corresponding finite distribution of a numerically defined law of probability. Furthermore, an equation has been worked out, that expresses the formal consistency between the finite data that characterise the above-mentioned procedure and the mathematical theorem of large numbers.

Professor Longo was aware of this work and I think that he has understood its social difficulties. I must mention that the same special issue contains a very interesting discussion of probability from a historical perspective, C. Porter, "Kolmogorov on the role of randomness in probability theory", of which I was unaware while developing my work. In this way I learned that quite a number of mathematicians are well aware of what I have called Kolmogorov's aporia, but they called it long before "the applicability problem", clearly a better name.

Mathematicians seem to believe that the applicability problem can be solved by purely mathematical means, while I believe that this is fundamentally impossible. I believe that the semantic content cannot be reduced to pure syntax, nor entirely "mimed' by it (in the sense in which a mould can 'mime' a face).

The special issue referred above contains also a brief debate between several outstanding contributors about the ways of connecting factual data with mathematical syntax. This debate brings into evidence that the applicability problem—even though Kolmogorov himself considered it so essential—not only is surprisingly little known, but, even when it is raised in quite explicit and insistent terms, it captures very little attention.

I believe that this state of facts deserves closer examination. Human intuition is magic. Nevertheless, the introduction of explicit principles and rules for matching a given semantic content and an assigned syntactic expression, could be very fertile, acting like a vehicle for rapid and precise understanding and consensus. People have lived before Aristotle's syllogistic, but its creation has avoided heaps of sophisms in heaps of lost time and effort. MRC offers a framework for matching safely semantic contents and syntactic structures.

CC: Your last book *Principles of a 2nd Quantum Mechanics* (arXiv:1310.1728, in French) presents yet another quantum mechanical formalism. What is wrong with the "1st quantum mechanics"?

MM-S: It is simply devoid of a theory of measurement acceptable from a formal as well as from a conceptual point of view, with general factual validity.

The von Neumann-Hilbert theory of measurement is, both, fallacious and devoid of general validity. As long as one is confined inside the formalism itself it is very difficult to fully perceive this. (Personally, I am startled to discover what an incredibly long time I needed in order to acquire what I now believe to be a clear and coherent view on the global structure of the quantum mechanical formalism.)

The problem of 'interiority', i.e. of ways of transgressing the limitations inside which one is yourself imprisoned, is a very difficult problem indeed. If the imprisonment is absolute, this problem is radically devoid of solution. This may seem trivial, but many fine authors act as if they were unaware of it, in particular, all those who make assertions concerning the entire Universe. Wittgenstein stressed this epistemological fact in various contexts. He repeated that in order to be able to think of a 'whole' one has to be able to be inside as well as outside of that 'whole'. To which he added his well-known injunction: *Whereof one cannot speak, thereof one must be silent.*

Now, what happens when one wants to size up globally, as well as in its details, the structure of the quantum mechanical representation of micro-states? The imprisonment inside this representation, of course, is not absolute. One can place oneself outside it. But what is available outside, on which one can place the feet of one's mind? There is the classical physics and the whole classical thinking, with its "objects", its space-time and causal structures. But everybody says that quantum mechanics violates all this and nevertheless—marvellously—'is working'. An organised formalism (outside of the quantum mechanical one) permitting to perceive consensually expressible specificities, or necessities, or impossibilities, does not exist.

And this is quite understandable. Indeed, quantum mechanics is the very first physical theory that introduces—-implicitly—what I have called 'transferred descriptions' of the physical entities. And, as I have already stressed, the whole organised thinking that is exterior to quantum mechanics ignores the concept of

primordial transferred descriptions. So with respect to this concept there cannot exist an organised outside.

As long as these conditions persist nothing can be asserted on the formalism of quantum mechanics in terms endowed with a precise meaning and with a character of objectivity. This, as a fact, is manifest since tenths of years. What is cruelly lacking is an organised structure of reference, different from quantum mechanics itself, but constructed in a way that permits to be clearly related with quantum mechanics, that admits a controlled comparison with quantum mechanics, in the details as well as globally.

So I constructed such an organised structure of reference. I maintained invariant that what is represented inside quantum mechanics, namely states of micro-systems, 'micro-states', but I constructed another representation involving them. Quite independently of quantum mechanics, I brought into evidence just the necessary and sufficient conditions for constructing a communicable and consensual representation of micro-states, but nothing more.In this way an epistemological-operational-methodological representation of the geneses of human very first pieces of knowledge on micro-states is obtained. I called this infra-(quantum mechanics) to be understood as 'beneath the formalism of quantum-mechanics'.

By systematic reference to infra-(quantum mechanics), the formalism of quantum mechanics reveals unexpected deficiencies. Here are three of them:

- It does not distinguish clearly between the individual level of conceptualisation, and the statistical one. In fact it almost entirely occults the individual level.

- It does not represent at all, neither mathematically nor informally, the way in which a describable micro-state is generated. The process of generation of a physical and individual micro-state is confused with something radically different, namely the process of 'preparation for measurement of the mathematical state vector' that represents the statistics of results of measurement obtained with numerous replicas of the physical micro-state that is involved.

- Quantum mechanics lacks a generally valid theory of measurement.

I have sketched a 2nd quantum mechanics where the deficiencies enumerated above (and some others) have disappeared. This new representation—not a re-interpretation—introduces measurement operations based on the de Broglie-Bohm guidance relation, but assumed to be an observable process, not an only conceived process. And whether the process is indeed observable, or not,. . . can be observed.

CC: Are you preparing an English version?

MM-S: I have already notably improved the French version and I shall soon update it on arXiv of quantum physics. As for the English version, it will be available before the end of July, I hope. Meanwhile I shall try to publish somewhere an extended abstract in English.

CC: Do you believe in the possibility of a grand unification between quantum mechanics and relativity?

MM-S: One can postulate that *if* one could directly observe micro-systems via signals travelling with a universally invariant velocity, then we would construct descriptions of them that would obey Einstein's theories. There is a very strong tendency to extrapolate into absolute generality an approach that has produced remarkable successes in some given domain.

But—personally—I do not see any reason why that postulate should be particularly fertile. I do not believe that what is called a "grand unification" is the best choice of an aim of today's Physics. I believe that the unique sort of a genuinely fertile unification of scientific rationality—in its entirety—can only be of a purely methodological nature. The contents should be left free of a priori constraints. They should emerge explicitly from all the specific conditions that are brought into play, so marked by unlimited diversity.

CC: As a researcher you had good moments and bad moments . Can you recall one of them?

MM-S: By far the best moments that I have had as a researcher—-and not very seldom—have been those that have emerged unexpectedly, when without any expressible specific cause I have suddenly felt a sort of inner certitude to have finally "understood" something that before, and for a long time, had stubbornly resisted my understanding.

CC: Many thanks.

MM-S: The thanks, indeed, are from my part.

Report on BCTCS 2013

The 29th British Colloquium for Theoretical Computer Science

24–27 March 2013, University of Bath

Guy McCusker

The British Colloquium for Theoretical Computer Science (BCTCS) is an annual forum in which researchers in Theoretical Computer Science can meet, present research findings, and discuss developments in the field. It also provides an environment for PhD students to gain experience in presenting their work in a wider context, and to benefit from contact with established researchers.

BCTCS 2013 was hosted by the University of Manchester, and held from 24th – 27th March 2013. The event attracted over 49 participants, and featured an interesting and wide-ranging programme of four invited talks and 28 contributed talks, in large part from PhD students, covering the full gamut of topics in theoretical computer science; abstracts of the talks are provided below.

The conference began with an invited talk by Samson Abramsky, University of Oxford, entitled "From Quantum Mechanics to Logic, Databases, Constraints, and Complexity". Other invited talks were given by Angela Wallenburg, Altran UK ("Proof and test: will they blend?") and Assia Mahboubi, INRIA–École Polytechnique ("Computer-checked Mathematics"), As in previous years, the London Mathematical Society sponsored a keynote talk in Discrete Mathematics: Susanne Albers, Humboldt-Universität zu Berlin, gave an excellent lecture on "Energy Efficient Algorithms". The financial support of the London Mathematical Society (LMS) in support of this lecture is gratefully acknowledged. We also acknowledge the financial support of the Heilbronn Institute for Mathematical Research which made available 24 student bursaries to cover full costs of attendance.

Invited Talks at BCTCS 2013

Samson Abramsky, University of Oxford
From Quantum Mechanics to Logic, Databases, Constraints, and Complexity

Quantum Mechanics presents a disturbingly different picture of physical reality to the classical world-view. These non-classical features also offer new resources and possibilities for information processing. At the heart of quantum non-classicality are the phenomena of non-locality, contextuality and entanglement. We shall describe recent work in which tools from Computer Science are used to shed new light on these phenomena. This has led to a number of developments, including a novel approach to classifying multipartite entangled states, and a unifying principle for Bell inequalities based on logical consistency conditions. At the same time, there are also striking and unexpected connections with a number of topics in classical computer science, including relational databases, constraint satisfaction, and complexity theory. The lecture will present an introduction to contextual semantics, in a self-contained, tutorial fashion.

Angela Wallenburg, Altran UK
Proof and Test: Will They Blend?

Extensive and expensive testing is the primary method used to gain confidence in safety-critical software today. There are some notable exceptions where formal software verification has been successfully used and scaled to large industrial projects. SPARK is a programming language, a set of verification tools, and a design approach for such critical systems. A number of military and commercial high integrity projects, ranging from 10 000 to 5 million lines of code, have been developed in SPARK. Examples include Rolls Royce Trent (engine control), EuroFighter Typhoon (military aircraft), and NATS iFACTS (air traffic control). We have identified two reasons why formal program verification is still a hard sell: 1) the difficulty of reaching non-expert users, and 2) the lack of a convincing cost-benefit argument. In this talk I will describe our approach to solve those two problems in the design of the new SPARK 2014 language and its associated verifying compiler, developed jointly by Altran UK and AdaCore. I will give an overview of some lessons learned from the programming language and verification research community, from the development of industrial standards such as DO-178C, and from our experiences in the industrial use of SPARK. In particular I will describe our unique integration of testing and proving. We argue that subprogram level formal verification using SPARK 2014 can be cheaper than testing in DO-178C terms, and that our integrated approach allows a mix of test and proof so that the most cost-effective method can be used for each part of a program.

Susanne Albers, Humboldt-Universität zu Berlin, the LMS-sponsored keynote

speaker in Discrete Mathematics.
Energy-Efficient Algorithms

We study algorithmic techniques for energy savings in computer systems. We consider power-down mechanisms that transition an idle system into low power stand-by or sleep states. Moreover, we address dynamic speed scaling, a relatively recent approach to save energy in modern, variable-speed microprocessors. In the first part of the talk we survey important results in the area of energy-efficient algorithms. In the second part we investigate a setting where a variable-speed processor is equipped with an additional sleep state. This model integrates speed scaling and power-down mechanisms. We consider classical deadline-based scheduling and settle the complexity of the offline problem. As the main contribution we present an algorithmic framework that allows us to develop a number of significantly improved constant-factor approximation algorithms.

Assia Mahboubi, INRIA–École Polytechnique
Computer-checked Mathematics

For the last decades, computers have been playing an increasing role in the everyday activity of many researchers in mathematics: for typesetting articles, for testing conjectures, and sometimes even for validating parts of proofs by large computations. However most mathematicians are hardly familiar with "proof assistants", which are also pieces of software for "doing mathematics with a computer". These systems allow their users to trust with the highest degree of certainty the validity of the proofs they have carefully described to the machine. So far proof assistants have been successfully employed to verify the correctness of hardware and software components with respect to given specifications, scrutinizing proofs that are too long and pedestrian to be checked by hand. In September 2012, a proof of the Odd Order Theorem (Feit-Thompson, 1963), which is a milestone for the the classification of finite simple groups, was machine-checked by the Coq proof assistant. In this case, the computer has verified a proof which does not rely on heavy computations but on a sophisticated combination of mathematical theories resulting in one of the longest published proof of its time. In this talk we will give an overview of the panel of research areas and methodologies that should be combined in order to ensure the success of such a formalization. Black (or white) board will eventually never be surpassed to convey and give rise to the intuitions of the mind who discovers new mathematics, but having proofs checked by a machine rather than by a human reviewer may open some new perspectives we will discuss.

Contributed Talks at BCTCS 2013

Chris Bak, University of York
Rooted Graph Programs

We present an approach for programming with graph transformation rules in which graph programs can be as efficient as programs in imperative languages. The basic idea is to equip rules and host graphs with distinguished nodes, so-called roots. At the start of the search process for the match of a graph transformation rule, roots in rules are matched with roots in host graphs. This facilitates a local search of the host graph in the neighbourhood of its root nodes, enabling rules to be matched and applied in constant time, provided that host graphs have a bounded node degree (which in practice is often the case). Hence, for example, programs with a linear bound on the number of rule applications run in truly linear time. We demonstrate the feasibility of this approach with a case study in graph colouring using the graph programming language GP.

Mohamed Arikiez, University of Liverpool
Combinatorial Optimization Techniques in Domestic Renewable Power Management

Our work is in the emerging area of Computational Sustainability. We contend that the area has a great potential for fostering cutting-edge research in Computer Science and related disciplines. In particular, the main aim of the research presented here was to design an intelligent interactive control system that efficiently manages the household energy needs taking into account presence of renewable power (hybrid Solar/Wind) and the resident's preferences in order to reduce consumed power from the utility grid and increase the immediate renewable power (RP) utilization (the ratio of total consumed RP to total Generated RP) without decreasing the comfort level. Despite the fact that installing a domestic renewable power generation system can reduce power bills, the utilization of this power still needs improvement because sometimes the surplus of RP could hit 70% (depending on output power of generation system and consumed power in the building) but no intelligent mechanism exists to try and exploit such resource before it gets dumped to a storage system or the national grid. We describe a novel Knapsack formulation that can be used to solve the resulting allocation problem and analyse its performances both in a real-life and simulated environment. Our results suggest that the approach could allow the immediate use of as much as 90% of the generated power surplus.

Giles Reger, University of Manchester
A pattern-based technique for inferring first-order temporal specifications

Formal program specifications are useful for a number of different applications – the most obvious being formal program verification. But they can also aid

program understanding, test generation, bug location, software development and other new applications that are the subject of active research. However, formal specifications are difficult and costly to write, and as a consequence, precise specifications are often missing, incomplete or informal. This has led to a growing interest in the area of specification inference (also known as model inference, specification mining, automata learning). These techniques extract temporal properties or state-based invariants from code or, more often, dynamic program traces. These techniques are defined by the coverage they can achieve, the expressiveness of the specification language they target and their ability to scale with program size. In this talk I introduce a technique for inferring temporal specifications that deal with data. In order to handle data effectively I make use of a highly expressive specification language (Quantified Event Automata) developed within the context of runtime verification to infer specifications using a technique where specification patterns are mined from program traces and then combined together. By targeting an expressive specification language this technique is able to discover useful specifications whilst maintaining scalability by adopting algorithms for efficient runtime monitoring.

Andrew Lawrence, Swansea University
Program Extraction in Action: A Verified Clause Learning SAT Solver

Modern SAT solvers typically include optimizations such as clause learning which are rarely treated with formal methods in practice. In this talk we show how to obtain such an optimized SAT solver together with a formal correctness proof by the method of program extraction from proofs: we have formalized a constructive proof of completeness for a modified DPLL proof system combined with unit resolution and extract a conflict driven clause learning SAT algorithm. This algorithm is capable of learning information during the search for a proof as well as performing non-chronological backtracking. This is a new case study in the area of program extraction and opens up many possibilities for future work. It also demonstrates how efficiency considerations can be taken into account at the proof level. The formalization and extraction has been carried out in the interactive proof assistant Minlog.

Gregory Woods, Swansea University
A Case Study On Imperative Program Extraction

The process of program extraction has long been associated with functional programs with little research in the direction of imperative program extraction. While many useful tools exist to extract functional programs (Agda, Isabelle, Coq and NuPRL) the simple fact is that most programs that are written are more towards the imperative paradigm. In this talk we explore a case study which demonstrates that imperative program extraction is possible. The problem we choose to solve

using this method is the classic of sorting a list of numbers. Many algorithms exist to solve this problem and we will focus on one of the most famous, Quicksort. We present a successful attempt at extracting a program, that yields imperative behaviour, from a constructive proof. The software used for this is the interactive theorem prover Minlog.

Matthew Gwynne, Swansea University

Towards a theory of good SAT representations

We aim at providing a foundation of a theory of "good" SAT representations (CNF clause-sets) F of boolean functions f. The hierarchy UC_k of unit-refutation complete clause-sets of level k was introduced by the authors, based on notions of hardness and generalised unit-clause propagation (UCP). We argue UC_k provides the most basic target classes for representation. That is, for a good representation, F in UC_k is to be achieved for k as small as feasible.

The first level of the hierarchy, UC_1, is the same as the class UC of unit-refutation complete clause-sets, introduced in 1994. The aim of UC was to offer a class of clause-sets which was good for knowledge compilation and representation. More formally, UC is the class of clause-sets where unit-clause propagation (UCP), a simple linear-time inference algorithm, is sufficient decide questions of clausal entailment. In 1995 the class $SLUR$ (Single Lookahead Unit Resolution) was introduced as an umbrella class for efficient satisfiability (SAT) solving. The motivation was to offer an algorithm for efficiently deciding satisfiability for existing poly-time SAT classes, including renamable Horn, extended Horn, hidden extended Horn, simple extended Horn, and CC-balanced clause-sets. In previous work we generalise SLUR to a hierarchy $SLUR_k$, again using generalised UCP, and show that these two hierarchies are in fact equal ($SLUR_k = UC_k$). This brings together the two notions of representation and efficient SAT solving, and allows one to think of "finding a good representation" as a form of "SAT knowledge compilation". As a first application of this dual perspective, we show that, for (fixed) $k \geq 1$, deciding whether a clause-set is in UC_k is coNP-complete.

UC_k is directly related to the space complexity of tree resolution. However, in general, it is known that modern SAT solvers can (in some sense) simulate stronger proof systems such as full-resolution. Using the notion of resolution width, we introduce the hierarchy WC_k of clause-sets with width-hardness k; for all k the class UC_k is a subset of WC_k. We introduce lower bound methods for WC_k and use these to prove separation results between UC_{k+1} and UC_k, as well as between WC_{k+1}) and WC_k. More formally, we show that for every $k \geq 1$ there are sequences of boolean functions with polynomial size equivalent clause-set representations in UC_{k+1} which have no equivalent polynomial-size representations in WC_k. The boolean functions for these separations are "doped" minimally unsatisfiable clause-sets of deficiency 1; we generalise their construction and show a

correspondence to a strengthened notion of irredundant sub-clause-sets. Turning from lower bounds to upper bounds, we believe that many common CNF representations fit into the UC_k scheme, and we give some basic tools to construct representations in UC_1 with new variables, based on the Tseitin translation.

Augustine Kwanashie, University of Glasgow
The Hospitals/Residents Problem with Free Pairs

The classical Hospitals/Residents problem models the assignment of junior doctors to hospitals based on their preferences over one another. In an instance of this problem, a stable matching M is sought which ensures that no blocking pair can exist in which a resident r and hospital h can improve relative to M by becoming assigned to each other. Such a situation is undesirable as it could naturally lead to r and h forming a private arrangement outside of the matching. This however assumes that a blocking pair that exists in theory would invariably lead to a matching being undermined in practice. However such a situation need not arise if the lack of social ties between agents prevents an awareness of certain blocking pairs in practice. Relaxing the stability definition to take such a scenario into account can yield larger stable matchings.

In this talk, we define the Hospitals/Residents problem with Free pairs (HRF) in which a subset of acceptable resident-hospital pairs are defined as free. This means that they can belong to a matching M but they can never block M. Free pairs correspond to resident and hospitals that do not know one another. Relative to a relaxed stability definition for HRF, called local stability, we show that locally stable matchings can have different sizes and the problem of finding a maximum locally stable matching is NP-hard, though approximable within 3/2. Furthermore we give polynomial time algorithms for three special cases of the problem.

Alexander Baumgartner, RISC, Johannes Kepler University of Linz
A Variant of Higher-Order Anti-Unification

The anti-unification problem of two terms t_1 and t_2 is concerned with finding their generalization, a term t such that both t_1 and t_2 are instances of t under some substitutions. Interesting generalizations are the least general ones. The purpose of anti-unification algorithms is to compute such least general generalizations. For higher-order terms, in general, there is no unique least general higher- order generalization. Therefore, special classes have been considered for which the uniqueness is guaranteed. One of such classes is formed by higher-order patterns. These are lambda-terms where the arguments of free variables are distinct bound variables. A rule-based anti-unification algorithm in simply-typed lambda-calculus which computes a least general higher-order pattern generalization will be presented. The algorithm computes it in cubic time within linear space and it has been implemented.

Iain McBride, University of Glasgow

The Hospitals / Residents Problem with Couples

Large scale allocation processes can be modelled as matching problems involving sets of participants who may express preferences over members of other sets. Centralised matching schemes, which use algorithms to solve the underlying matching problems, are often employed in such allocation processes.

The National Resident Matching Program (NRMP) was established in 1952, in response to problems with the previous competitive system, to match graduating medical residents to hospitals in the US, matching 25,526 students in 2012. A similar process is used in Scotland to match medical graduates to Foundation Programme places via the Scottish Foundation Allocation Scheme (SFAS). These schemes may be modelled by a classical combinatorial problem, the Hospitals / Residents Problem (HR).

Centralised matching schemes such as these have had to evolve to accommodate couples who may wish to be allocated to (geographically) compatible hospitals. This extension, which can be modelled by the Hospitals / Residents Problem with Couples (HRC), has been in operation in the NRMP for a number of years and has also been applied more recently in the SFAS context.

The classical Gale-Shapley algorithm solves the Hospitals / Residents problem by finding a so called stable matching. We prove that, even under some very severe restrictions, the problem of deciding whether a stable matching exists, given an instance of HRC, is NP-complete. These complexity results drive the search for alternative methods of dealing with such problems.

We describe an Integer Programming model of the Hospitals / Residents Problem with Couples which produces exact, optimal solutions in larger instances where previously only heuristics, which are not guaranteed to terminate, have been applied. We prove the validity of the model and demonstrate the empirical performance of an implementation over a number of randomly generated datasets in addition to anonymised real data from the SFAS context.

Nosheen Gul, University of Leicester

A Process Calculus for Ubiquitous Computing

In the ubiquitous computing setting computing devices are distributed and could be mobile, and interactions among devices are concurrent and often depend on the location of the devices. Process calculi are formal models of concurrent systems and mobile agents. In particular, Calculus of Communicating Systems (CCS, for short) of Milner is a well suited formalism for agents executing concurrently, and Mobile Ambients (MA) by Cardelli and Gordon is a formalism for agents' mobility. We propose a process calculus for specifying mobility, communication, and concurrency in the ubiquitous computing setting. The calculus is inspired by CCS

and Mobile Ambients. We use the idea of ports as in CCS, that allow agents to communicate on, and ambient capabilities as in Mobile Ambients, allowing the agents to move around. We give an LTS-based operational semantics for our calculus, which is inspired by Merro and Hennessy operational semantics. Then we provide some examples to show the usefulness of our calculus.

Andrew Fish, University of Brighton
Ordered Gauss Paragraphs

The talk will discuss recent work on the EPSRC funded Automatic Diagram Generation project which aims to build a unified framework for the automatic generation of mixed-type diagrams arising as the integration of Euler diagrams, knot diagrams, and graphs. There has been limited prior consideration of mixed-type diagram generation, and the intent is bring theoretical benefits by developing methods which make use of any commonality of abstraction, together with practical oriented benefits in terms of providing the groundwork for generic tool support for such diagrams that may be used in areas such as diagrammatic logics, or ontology and network visualisations. The talk will focus on Euler diagrams, which are collections of closed curves used to visualise set systems, discussing a new encoding for Euler diagrams, using Ordered Gauss Paragraphs, making use of an existing code together with methods for solving the planarity problem for knots in order to solve the corresponding planarity problem for Euler diagrams. We indicate how the code encapsulates the topology of the diagram, demonstrate the generality of the approach, and provide a link between knots and Euler diagrams via a construction which yields a family of Brunnian links which project to Edwards' construction of Venn diagrams, observing that the code rewriting methods developed are more widely applicable

Kevin McDonald, University of Aberdeen
A Substructural Logic of Layered Graphs

Complex systems, be they natural or synthetic, are ubiquitous. In particular, complex networks of devices and services underpin most of society's operations. By their very nature, such systems are difficult to conceptualize and reason about effectively. The concept of layering is widespread in complex systems, but has not been considered conceptually. Noting that graphs are a key formalism in the description of complex systems, I will establish a notion of a layered graph and provide a logical characterization of this notion of layering using a non-associative, non-commutative substructural, separating logic.

Layering need not be defined in one direction only: it may be that two graphs are layered over each other. In modelling terms, this would mean that whilst it remains useful to separate the two layers, resources can flow both up and down. To this end, I establish a notion of 'bi-layering' that is consistent with the basic

notion of layering and also the intuitive notion of a stack.

I will define a class of algebraic models that includes layered graphs for which soundness and completeness results can be obtained. This gives a mathematically substantial semantics to this very weak logic.

The notion of layering that I develop has many natural applications in complex systems modelling. One particularly appealing area of application lies in security, such as instances of security circumvention or a flaw in the security policy of an organisation based on lax protocols. There are many others in a variety of network settings, the IP Stack, for example. I will present some simple examples before discussing how my work could be applied to more complex security issues such as investigating how I may begin to compose security models such as Bell-LaPadula and Biba in the layered environment.

Abiar S. Al-Homaimeedi, King's College London
Achieve pi-calculus Style Mobility in CSP

In process calculi, passing channel names is considered as transferring of communication capabilities from one process to another, usually called mobility. Introducing mobility into CSP as in the pi-calculus is not straightforward for the following reasons: (i) the parallel composition in CSP is parametrised with an interface set which governs the synchronisation between participants. Events in this set should be simultaneously per- formed by all participants whereas events outside this set (even if they are shared) are not. Although CSP parallel composition improves com- munication exibility, it lets processes alphabets play a significant role in the communication. Therefore, the silent growth of alphabets as in pi-calculus is not enough. Processes alphabets should be grown explicitly because of its relation with the parallel operator. (ii) restricting communication to names as the pi-calculus, is insuficient in the CSP. The restriction will compromise the CSP typed multi-way communications To overcome this problem several solutions have been proposed. However, each of these models have some drawbacks, therefore, in this talk, we propose a new mobility model to accommodate mobility into CSP. Our mobility model generalises the notations and relaxes the restrictions which are made by one of the previously proposed models. Additionally, we introduce a novel dynamic algorithm to update the synchronisation set of the generalised parallel operator.

Shang Chen, Loughborough University
Computability of Hybrid Systems

In this presentation, we will introduce several models of hybrid systems and discuss the computability of reachability and convergence properties of them. Hybrid systems are a model incorporating both discrete and continuous dynamics in the same formalism, which can be used to describe a large number of real-world

applications. They are often used in places where we have some form of discrete device acting in a continuous environment. Firstly, we will introduce several mathematical models studied in this area such as piecewise constant derivative systems (PCD), piecewise affine maps (PAM), timed automata (TA) and rectangular automata (RA). We will then explain the problems we are interested in for these models, which include reachability, control and stability problems. We will then survey some known results from the literature from the point of view of decidability. Finally we will discuss future research directions, some applications of hybrid systems and some new areas which seem worthy of study.

Casper Bach Poulsen, Swansea University
Partial Derivation in Modular Structural Operational Semantics

Abstract: The scientific study of programming languages requires a formal specification of their semantics. However, the incentives of applying formal specification frameworks during programming language design are often outweighed by more pragmatic concerns, such as developing and maintaining an executable interpreter for the language under design.

One way of bridging the gap between formal specification and pragmatic programming language design is by making formal specifications pragmatic for the language developer. Modular structural operational semantics (MSOS), a modular variant of structural operational semantics (SOS), is a formalism that supports incremental and scalable language design, e.g., by taking a component-based approach to semantic specification.

Interpreting the transitive closure of the transition function for a set of MSOS rules gives a prototype interpreter, where evaluation corresponds to proof derivation using the underlying MSOS rules. However, a naive implementation of such an interpreter has a worst-case interpretive overhead where each proof step requires a number of inferences that is linear in the depth of the input term. Furthermore, while small-step semantics have several declarative advantages, term reduction using small-step rules requires more inference steps than when using their big-step counterparts. For the programming language designer who is concerned with efficiency, the considerable interpretive overhead incurred by a naive interpretation may be unacceptable in practice.

Here, we explore how to reduce interpretive overhead of small-step MSOS rules through partial evaluation techniques which, in our modular structural proof system setting, we will call partial derivation. Combining ideas from partial evaluation in logic programming, bisimulation theory, and refocusing in reduction semantics we show how to derive rules whose proofs require fewer inferences (and hence, whose evaluation requires less computation). Applying partial derivation to a semantics is a fully mechanisable transformation that gives a provably semantically equivalent set of rules. Furthermore, the techniques are broadly applicable,

being constrained by only a very mild set of conditions for correctness. The transformations result in rules with a big-step flavour, hinting at the inter-derivability of small-step and big-step style semantics.

As a proof of concept, we have prototyped semantic rules in Prolog, where we can observe a significant reduction in the running time of interpreters based on partially derived semantics in comparison with their naive counterparts. We conclude that partial derivation is a viable technique for reducing interpretive overhead in modular structural proof systems and practical interpreters derived from these, and that partial derivation is a viable tool for prototypical and pragmatic language design.

Timothy Revell, University of Strathclyde
Relational Semantics of Type Systems

Category Theory, in particular cartesian closed categories, provide a powerful semantics for the simply typed lambda calculus (STLC). Logical relations are another model of the STLC using the category of relations. In this talk, we shall describe the relationship between these two models using a fibrational framework. We show how two important results, the Fundamental Theorem of Logical Relations (otherwise known as the Parametricity Theorem) and the Identity Extension Lemma have natural and simple formulations within this fibrational framework. We will conclude by discussing fibred category theory and how it can describe concisely the ideas of this talk. In particular, parametricity simply means that we shift from working in a categorical universe of categories, functors and natural transformations, to working in a fibrational universe of fibrations, fibred functors and fibred natural transformations.

David Wilson, University of Bath
Advances in Cylindrical Algebraic Decomposition

Cylindrical Algebraic Decomposition (CAD) was initially introduced to tackle the classic problem of quantifier elimination over real closed algebraic fields, however it has since seen many applications in its own right. Given a set of polynomials, multiple algorithms exist to produce a CAD such that over each cell the polynomials have constant sign. Inherently doubly exponential in the number of variables present, much work has been done to make CAD a practical tool through preconditioning, more efficient construction and truncated algorithms.

I will give a brief history of CAD before covering work conducted by the University of Bath real geometry research group. Recently, we have shifted emphasis to try and produce a CAD for a given *problem* rather than the set of polynomials involved. A major step forward is research on Truth Table Invariant CADs (TTICADs) for which a set of given clauses have invariant truth value over each cell. This research has also led to further investigation of how problems are formulated

for input into various related algorithms.

Alongside new research, key applications will be discussed. In particular, recent work on the use of CAD to verify identities involving multi-valued functions over the complex numbers will be described. This work will be included in the forthcoming release of the computer algebra system MAPLE 17.

This work was conducted with James Davenport, Russell Bradford and Matthew England at the University of Bath. The work on TTICADs was also conducted jointly with Scott McCallum of Macquarie University.

Joseph Davidson, Heriot-Watt University
Elegance requires eloquence

oChaitin's exploration of his notion of program elegance using the Lisp language does not explicitly take into account the balance between a notation's expressive power and the richness of its semantics. To investigate further this link, we have developed a flavour of the Random Access Stored Program (RASP) machine to compare with the traditional Turing machine model.

By implementing interpreters and compilers from RASP to TM and vice versa, in both RASP and TM, we believe that we can gain a more precise view into the expressive power of these languages. Bootstrapping the compilers on one another will allow examination of the models from a common representation. We can also investigate the full abstract chain of the model, from the most abstract - the operational semantics - to the most concrete - the implementation of programs which actually run on realisations of these models.

This talk presents where we have come from, where we want to end up and what we hope to find along the way.

Paolo Torrini, Swansea University
Parametric polymorphism, value restriction and resource logic

Hindley-Milner polymorphism is a form of parametric polymorphism that is widely used in functional languages, for efficiency reasons. It is also known as *let* polymorphism, as it allows for generalisation of type variables that do not occur free in the environment of *let* expressions. The soundness of this form of generalisation relies on the logic of propositional quantification as enshrined in system F, although it can be syntactically defined on top of a distinction between types and type schemes, making it possible to dispense with explicit use of quantifiers.

In languages with references, there are well-known problems that arise when the term fed to the *let* expression is not a value. If the evaluation of this term requires allocation of new references, and its type depends on type variables that occur in the types of such references, one may end up with typeable expressions that still lead to runtime errors. This problem is usually dealt with by means of some form of the so-called *value restriction*.

In the classic approach, which dates back to the early '90 and is essentially due to Mads Tofte, value restriction is handled by distinguishing variables that may occur in the type of references (imperative), from those that cannot (applicative). The analysis in Tofte's paper shows that the justification of value restriction boils down, again, to the fact that variables can be generalised only when they do not occur free in the environment — though this time in an extended sense, that should take the store into account.

Tofte's analysis may then suggest, that by relying on a more expressive logic, allowing for premises to represent resources needed for evaluation, a more declarative formulation of the restriction could be given, simply based on the free variable criterion. In fact, given a typeable program, when the types of the references that need to be allocated for its evaluation are included in the premises of its typing judgement, the restriction on generalisation turns out be logically enforced without further ado.

In this talk, we first present the standard approach to value restriction in terms of imperative and applicative variables. We then outline an alternative approach based on intuitionistic linear logic, allowing for more expressive typing judgements which include store types. At a basic level, the new typing may be intuitively understood as obtained by reversing the operational semantic evaluation big step $\rho \vdash \langle t, \sigma\rangle \longrightarrow \langle v, \sigma'\rangle$ where the value v has type τ, and the new store σ' is obtained by extending σ with the newly allocated references $a_1 \mapsto v_1, \ldots, a_k \mapsto v_k$ of types $\tau_1, ..., \tau_k$, into a judgement of form $\Gamma; \Delta \vdash t \Rightarrow \tau$ where Γ types the environment ρ, and crucially, $\Delta = \{l_1 : \tau_1, \ldots, l_k : \tau_k\}$ types the difference between the old store and the new one, in terms of the locations $l_1, \ldots, l_k$ needed to allocate the new references.

Thomas Gorry, University of Liverpool
Faster Communication-less Agent Location Discovery on the Ring

This talk will be about our on going study of a randomised distributed communication-less coordination mechanism for n uniform anonymous agents located on a circle with unit circumference. We assume the agents are located at arbitrary but distinct positions, unknown to other agents. The agents perform actions in synchronised rounds. At the start of each round an agent chooses the direction of its movement (clockwise or anticlockwise), and moves at unit speed during this round. Agents are not allowed to overpass, i.e., when an agent collides with another it instantly starts moving with the same speed in the opposite direction. Agents cannot leave marks on the ring, have zero vision and cannot exchange messages. However, on the conclusion of each round each agent has access to (some, not necessarily all) information regarding its trajectory during this round. This information can be processed and stored by the agent for further analysis. The location discovery task to be performed by each agent is to determine the initial position of every other

agent and eventually to stop at its initial position, or proceed to another task, in a fully synchronised manner. Our primary motivation is to study distributed systems where agents collect the minimum amount of information that is necessary to accomplish this location discovery task. Previously we have shown that by using a fully distributed randomised technique this location discovery problem can be solved in $O(n \log^2 n)$ rounds. However, we can now show improvements to this with an algorithm that solves the location discovery problem w.h.p in $O(n+\log^2 n)$ rounds.

Diana Cionca, University of Surrey
Path Dependency Analysis in Complex Systems

Nowadays, business environments are changing very fast from centralised and closed to distributed and open. Typically, they involve a large number of entities or agents who interact in a dynamic, uncertain and unpredictable fashion. There is growing interest in the development of analytical tools for understanding the behaviour of such complex systems both from an individual's point of view and from the global interaction perspective. Agent-based scenario analysis has been proposed for the analysis of complex systems. The agent's behaviour is considered the key factor that influences the overall system's evolution. An agent can reason to achieve certain goals, can act autonomously, has a knowledge-base about its environment and can interact with other agents. The objective here is to predict and model the evolution of a complex system through a set of rules which describe the behaviour and interactions of participating agents. We look into web services applications for open and distributed systems like the Web, and find that similar issues arise, especially with regard to orchestration (individual viewpoint) and choreography (global viewpoint) of participating services. We argue that the way these interactions are modelled, in particular with respect to handling concurrency, is important when it comes to specification and verification (conformance and realisability) of a choreography specification. In addition, we are keen to investigate the use of business rules in arriving at a choreography specification in a declarative fashion. We take a case study from the ERIE project on global food supply chains as a complex system and build a model which can be used to reason about the system's behaviour, in terms of inter-dependencies and different possible outcomes.

Ben Horsfall, University of Sussex
Using a separation logic for verification of reflective programs

Reflective programming allows one to construct programs that can manipulate or examine their behaviour or structure at runtime. One of the benefits is the ability to create more generic code that is able to adapt to being incorporated in different larger programs without modification to suit each concrete situation. Due to

the runtime nature of reflection, static verification is limited and has largely been ignored. This talk gives an overview of research into a method for specification and verification of a reflective library by utilising a separation logic that has been extended with support for stored procedures. The approach stores the metadata on the heap such that a reflective library can be implemented and verified in terms of primitive commands, rather than developing new proof rules for the reflective operations. The specified library may then be used to verify programs that use reflection. The support for stored procedures in the logic is important for the chosen technique, where the metadata representation of method and constructor objects are realised as stored procedures. The supported reflective operations characterise a subset of Java's reflection library, and the approach is supported by a tool providing semi-automated verification.

Ferdinand Vesely, Swansea University
Compiler back-end for a component-based semantic specification framework

Traditional approaches to formal programming language specification are generally criticised for being difficult to use. This difficulty impedes their wider adoption. The main points of criticism are usually the notation, which requires too much effort to penetrate, or lacking tool support. Action semantics is one example of a framework that was designed to address the issue of comprehensibility in particular. It provides a closed collection of semantic entities. Concrete programming language constructs are defined by translations into action notation. The notation itself has some shortcomings, such as a somewhat unusual syntax using action combinators. A new framework is currently being developed that will provide an open ended collection of named fundamental constructs, or funcons. Each funcon has formally defined dynamic and static semantics and is stored in a repository. Real programming language constructs are defined in terms of funcons and thus programs can be translated into funcon terms. Case studies on a subset of OCaml and Caml Light have already been carried out and there is tool support for translating program terms into funcon terms as well as an interpreter for these translations. Modular SOS is used do give definitions of individual funcons. This variant of SOS was designed to address modularity issues of standard SOS and allows independent definition of language constructs. This is made possible by using transition labels for auxiliary entities and automatically propagating all unmentioned entities between the premises and conclusion of a rule.

As good tool support for prototyping of the language being designed is deemed critical for the success of a specification framework, multiple tools have been developed for action notation. Iversen developed a compiler for action notation which translates actions into Standard ML code. This code can then be compiled by an optimizing ML compiler. The compiler chain has been tested on Standard ML programs with satisfying results in execution speeds. The action compiler

itself did not perform any optimisations. We build on Iversen's work on the action compiler and aim to develop an optimising compiler back-end for translating funcons into executable code. In the first phase, translations of funcons into Caml Light programs will be designed and implemented. A specification for this language is already available and we can use the existing tool support as a front-end to translate Caml Light programs into funcon terms. Once we have a working prototype of the back-end, we should be able to do a round-trip by translating from a Caml Light program into a funcon term and then back into Caml Light in a similar manner to Iversen's action compiler. This will allow us to evaluate our approach by comparing performance of code generated through funcons to code generated directly by the Caml Light compiler. In this talk we will discuss preliminaries and observe the differences between action notation and funcons. We will give an overview of Iversen's action compiler and suggest an approach to compiling funcons into Caml Light.

Andrew Collins, University of Liverpool
Visualisation and Analysis of Graphs

In this talk I will discuss work completed by the authors in the area of graph visualiation and graph analysis. Specifically I will show our current work in force directed algorithms for graph layout and the methods that we use to identify a significant vertex within a graph. Further I will make a reference to the future directions that we hope to take the work we have completed. While I will be showing mostly applied concepts, nearly all aspects of the work are backed by deeply theoretical work. Throughout this talk we will look at the visualisation and analysis of the retirement of Pope Benedict XVI and (possibly) the election of his successor.

Patrick Totzke, University of Edinburgh
Checking Equivalences and Preorders of One-Counter Processes

I will outline recent results on the Verification of Pushdown Systems, specifically on checking Bisimulation, Simulation and Trace inclusion of various restrictions of One-Counter Processes.

Of particular interest is a model called One-Counter Nets, that can be seen both as restriction of PDA and Petri nets and inherits a structural monotonicity from the latter. I will provide some intuition on how to expoit this property to provide decision procedures.

If time permits I will discuss the interplay of monotonicity and infinitely branching.

Robert Powell, Durham University
Skew Bisubmodularity and Valued CSPs

An instance of the finite Valued Constraint Satisfaction Problem (VCSP) is given by a finite set of variables, a finite domain of values, and a set of finite valued functions, where each function depends on a subset of the variables. The goal is to find an assignment of values to the variables that minimises the total sum of the functions. We study (assuming that PTIME $\neq$ NP) how the complexity of this very general problem depends on the functions allowed in the instances. The case when the variables can take only two values was classified by Cohen et al., with submodular functions giving rise to the only tractable case. Any non-submodular function can be used to express, in a certain specific sense, the NP-hard Max Cut problem. We investigate the case when the variables can take three values. We identify a new infinite family of conditions, that includes bisubmodularity as a special case, which can collectively be called skew bisubmodularity. By a recent result of Thapper and Zivny, this condition implies that the corresponding VCSP can be solved by linear programming. We prove that submodularity with respect to a total order and skew bisubmodularity give rise to the only tractable cases, and, in all other cases, Max Cut can be expressed. We also show that our characterisation of tractable cases is tight, that is, none of the conditions can be omitted. Thus, our results provide a new dichotomy theorem in constraint satisfaction research, and lead to a whole series of intriguing open problems in submodularity research.

Jules Hedges, Queen Mary University of London
Selection functions and games

Selection functions are a family of higher-type functionals related to continuations, introduced by Martin Escardo and Paulo Oliva to extract computational content from proofs in classical analysis. An unexpected connection with game theory arose: many apparently unrelated proofs in constructive mathematics can be seen as computing subgame-perfect equilibria of a suitable kind of generalised sequential game. I show that a certain amount of classical game theory carries over to this more general setting: generalised sequential games can be turned into simultaneous games based on von Neumann's 'strategic-form' construction, and Nash's theorem for the existence of mixed-strategy equilibria of finite games still holds.

Report on BCTCS 2014

The 30th British Colloquium for Theoretical Computer Science

9–11 April 2014, Loughborough University

Paul Bell, Daniel Reidenbach

The British Colloquium for Theoretical Computer Science (BCTCS) is an annual forum in which researchers in Theoretical Computer Science can meet, present research findings, and discuss developments in the field. It also provides an environment for PhD students to gain experience in presenting their work in a wider context, and to benefit from contact with established researchers.

BCTCS 2014 was hosted by the Department of Computer Science at Loughborough University, and held from 9th – 11th April 2014. The event attracted over 40 participants from sixteen universities, and featured an engaging and wide-ranging programme of four invited talks and 25 contributed talks. These were in large part from PhD students and covered the full gamut of topics in Theoretical Computer Science. Abstracts of the talks are provided below.

The conference began with an invited talk by Leszek Gąsieniec, University of Liverpool, entitled "Distributed maintenance of mobile entities". Other invited talks were given by Timo Kötzing, Friedrich-Schiller-Universität Jena/Germany ("Recent advances in inductive inference") and Achim Jung, University of Birmingham ("A modal Belnap logic"). As in previous years, the London Mathematical Society (LMS) sponsored a keynote talk in Discrete Mathematics, which this year was given by Jeffrey Shallit, University of Waterloo/Canada, on "Open problems in automata theory". The financial support of this lecture by the LMS is gratefully acknowledged. We also acknowledge the generous financial support of the Heilbronn Institute for Mathematical Research, which provided 24 bursaries to cover the full costs of attendance for research students.

Invited Talks at BCTCS 2014

Leszek Gąsieniec, University of Liverpool
Distributed maintenance of mobile entities

With the recent advent of ad-hoc, not well-structured, large, and (very often) dynamic network environments there is a strong need for more robust, universal, and inexpensive distributed network protocols. The purpose of these protocols is to support basic network formation and integrity mechanisms as well as more dedicated tasks such as information dissemination, network search and exploration, network monitoring and others.

One of the novel and promising alternatives in supporting such network protocols are dedicated teams of mobile entities (MEs) that can work independently on top of basic network system routines. The MEs' ability to communicate and to move within the environment impels the design and implementation of efficient formation, communication and navigation mechanisms including motion control and coordination mechanisms that allow MEs to perform dedicated tasks collectively.

We will provide an introduction to the field and will discuss several extensively studied algorithmic problems as well as those just touched upon in the recent years. The talk will be concluded with open problems.

Achim Jung, University of Birmingham
A modal Belnap logic

Four valued logic was introduced by Nuel Belnap in the 70s. It is very easy to motivate and seems to be central to Computer Science; in fact, one of his papers on the subject was called "How a computer should think". Adding mathematical structure to his basic ideas turned out not to be so easy, however. Much work was done by Arieli and Avron in the 90s, and more recently, by Umberto Rivieccio, a collaborator on the work to be presented, which concerns a modal extension of Belnap's work. The topic is also related to my longstanding interest in using Stone Duality to link semantics and logic for computer science.

Timo Kötzing, Friedrich-Schiller-Universität Jena/Germany
Recent advances in inductive inference

3,5,7,11,13 – what's next? What general rule (apparently) produces this sequence? Maybe the sequence lists all the odd primes, but what if the next datum is 15? Maybe all odd numbers that are not squares? Since the 1960's there are formal models defining what it means to learn or predict such sequences; this area of research is called inductive inference. In this talk I will briefly review the main classical results and then focus on recent advances in inductive inference, espe-

cially concerning the development of general techniques. Applications of these techniques include, for example, questions regarding avoidance of seemingly inefficient learning behavior.

Jeffrey Shallit, University of Waterloo/Canada, the LMS-sponsored keynote speaker in Discrete Mathematics
Open problems in automata theory
In this talk I will survey some of my favorite open problems from automata theory, including the separating words problem, decidability problems related to number theory and the Endrullis-Hendriks problem on transducers.

Contributed Talks at BCTCS 2014

Eleni Akrida, University of Liverpool
Ephemeral networks with random availability of links: diameter and connectivity
In this work we consider temporal networks, the links of which are available only at random times (randomly available temporal networks). Our networks are ephemeral in the sense that their links appear sporadically, only at certain times, within a given maximum time (called lifetime of the network). More specifically, our temporal networks notion concerns networks, whose edges are assigned one or more random discrete-time labels drawn from a set of natural numbers. The labels of an edge indicate the discrete moments in time at which the edge is available. In such networks, information (e.g., messages) have to follow temporal paths, i.e., paths, the edges of which are assigned a strictly increasing sequence of labels. We first examine a very hostile network: a clique, each edge of which is known to be available only one random time in the time period $\{1, 2, ..., n\}$ (where n is the number of vertices). How fast can a vertex send a message to all other vertices in such a network? To answer this, we define the notion of the Temporal Diameter for the random temporal clique and prove that it is $\Theta(\log n)$ with high probability and in expectation. In fact, we show that information dissemination is very fast with high probability even in this hostile network with regard to availability. This result is similar to the results for the random phone-call model. Our model, though, is weaker. Our availability assumptions are different and randomness is provided only by the input. We show here that the temporal diameter of the clique is crucially affected by the clique's lifetime, a, e.g., when a is asymptotically larger than the number of vertices, n, then the temporal diameter must be $\Omega((a/n) * \log n)$. We, then, consider the least number, r, of random instances at which an edge is available, in order to guarantee at least a temporal path between any pair of vertices of the network (notice that the clique is the only network for

which just one instance of availability per edge, even non-random, suffices for this). We show that r is $\Omega(\log n)$ even for some networks of diameter 2. Finally, we compare this cost to an (optimal) deterministic allocation of labels of availability that guarantees a temporal path between any pair of vertices. For this reason, we introduce the notion of the Price of Randomness and we show an upper bound for general networks.

Theofanis Apostolopoulos, King's College London

Sparse signal recovery as a non-linear problem

My contributed talk will focus on a novel research field, called Compressed Sensing (CS) method, which has attracted considerable research with several new application areas, mainly signal and image compression. It was introduced recently for simultaneously sampling and compressing signals and enabling new reconstruction techniques for applications where the standard sampling process is not feasible or very expensive. In fact, CS adopts a new sampling scheme that does not follow the principle of conventional approach depicted by the sampling theorem of Nyquist-Shannon. The goal is to efficiently recover any type of signal, such as speech and image data, using what was previously considered as highly incomplete and inaccurate (under-sampled) measurements. This is an ill-posed inverse problem, which can be solved as an l0 norm based optimisation problem, with the aim to find the best fit which minimises the difference between the solution and the observations while satisfying all the given constraints. In this talk, I will also introduce a new swarm based heuristic for efficiently recovering signals, with high probability. It is an iterative process which finds an approximation of the l0-norm based problem viewed as a combinatorial optimization problem. In each iteration every agent calculates and carries a slightly different feasible solution based on the current best (optimal) solution, which is necessary so as to avoid being trapped to one of the numerous local minima. This method is very efficient and quick compared to other conventional methods, such as the classical log-barrier and Least squares methods, even under the presence of noise, based on experimental results. In particular, the heuristic is compared with other alternative sparse recover methods in terms of complexity, computational time, samples size, and recovery error. Possible improvement for enhancing the performance of the heuristic could be to re-weight the approximate l0 norm, by using coefficients at every iteration; an approach that has been applied successfully to similar l0, l1 and l2 norm based CS problems.

Christopher Bak, University of York

Towards an implementation of rooted graph programs

Rooted Graphs are used to improve the efficiency of graph matching when applying graph rewriting rules. The basic idea is to automatically match a node in the

rule to a node in the host graph, restricting the search space to a small area of the host graph. The increase in efficiency comes at the cost of flexibility: graph programs consisting of rooted rules are significantly more complicated than equivalent programs with standard rules. I present a Topological Sorting graph program which strikes a balance by using both rooted and non-rooted rules to solve the problem, and discuss some issues in the ongoing implementation of rooted graph programs in the graph programming language GP.

Jannis Bulian, University of Cambridge
Graph isomorphism parameterized by elimination distance to bounded degree

A commonly studied means of parameterizing graph problems is the deletion distance from triviality, which measures the number of vertices that need to be deleted from a graph to place it in some class for which efficient algorithms are known. In the context of graph isomorphism, we define triviality to mean a graph with maximum degree bounded by a constant, as such graph classes admit polynomial-time isomorphism tests. We generalise deletion distance to a measure we call elimination distance to triviality, based on elimination trees or tree-depth decompositions. We establish that graph isomorphism is FPT when parameterized by elimination distance to bounded degree, generalising results of Bouland et al. on isomorphism parameterized by tree-depth.

Leroy Chew, University of Leeds
The complexity of theorem proving in circumscription and minimal entailment

Circumscription is one of the main formalisms for non-monotonic reasoning. It uses reasoning with minimal models, the key idea being that minimal models have as few exceptions as possible. In this contribution we provide the first comprehensive proof-complexity analysis of different proof systems for propositional circumscription. In particular, we investigate two sequent-style calculi: MLK defined by Olivetti (J. Autom. Reasoning, 1992) and CIRC introduced by Bonatti and Olivetti (ACM ToCL, 2002), and the tableaux calculus NTAB suggested by Niemelä (TABLEAUX, 1996). In our analysis we obtain exponential lower bounds for the proof size in NTAB and CIRC and show a polynomial simulation of CIRC by MLK. This yields a chain NTAB < CIRC < MLK of proof systems for circumscription of strictly increasing strength with respect to lengths of proofs.

Michalis Christofi, King's College London
Worst-case behavior of distributed algorithms for the maximum concurrent flow problem

A Multicommodity Flow Problem is a problem of designing flows of commodities in a common network. The flows must be feasible, that is they cannot exceed the edge capacities, and they must satisfy the demand of each commodity. Multi-

commodity flow problems have a wide variety of important applications in areas such as VLSI circuit design, network design, production and distribution of goods, transportation systems and communication systems. We consider the multicommodity flow problem which is called the Maximum Concurrent Flow problem. The objective is to minimise the maximum edge congestion, where the congestion of an edge is defined as the ratio of the flow to the capacity. In this talk we discuss algorithms which solve this problem in the following distributed manner: one agent controls one commodity, and the agents communicate at the end of each computation round via a billboard. Algorithms of this type were proposed by Awerbuch, Khandekar and Rao [SODA 2007] and Awerbuch and Khandekar [PODC 2007], who showed that an approximate solution can be reached in the number of rounds which is linear in the maximum length L of a path followed by any flow. We show that this running-time bound is asymptotically tight by constructing a worst-case input network and analyzing the performance of the algorithms on this network. We also propose a heuristic improvement of these algorithms, analyze its performance on our worst-case input, and indicate why we should expect that it improves running times on general networks.

Alejandro Erickson, Durham University

Computer science takes back data centre networks from engineering

Companies like Google, Amazon, and Microsoft house massive warehouses full of interconnected computers which provide services to the whole world. The demand for such services is pushing the limits of traditional data centre designs, and this research area, long dominated by engineers, is becoming a hot topic in theoretical computer science. How can currently available equipment be interconnected in order to increase the size and performance of data centres while reducing the relative cost?

I give an overview of some recent "computer science-y" developments in the world of data centres, and I discuss a novel approach for converting an arbitrary graph into a dual-port server-centric data centre network.

This work is supported by the EPSRC grant "INPUT: Interconnection Networks, Practice Unites with Theory".

Carl Feghali, Durham University

On the complexity of partitioning graphs into disjoint cliques and a triangle-free subgraph

We investigate the computational complexity of deciding whether the vertices of a graph can be partitioned into a disjoint union of cliques and a triangle-free subgraph. This problem is known to be NP-complete on arbitrary graphs. Our hardness results are on planar graphs and perfect graphs. In contrast, we provide a finite list of forbidden induced subgraphs for cographs with such a partition, thus

yielding a linear time recognition algorithm.

(Joint work with Faisal N. Abu-khzam and Haiko Müller.)

Michael Gale, University of Cambridge
Solving an existential crisis in Haskell

Haskell's type system provides mechanisms for type refinement within the scope of certain value expressions if GADTs or type classes are used. The type system propagates sufficient information to ensure that nothing can go wrong even if types are erased from the run-time representation of a program. This is not the case when we are using existential types, where we deliberately hide concrete types from the type system. Nevertheless, we may desire to eliminate existential types in a different part of a program in order to restore the original types. For this purpose, we propose an extension to Haskell which allows programmers to restrict existential types within individual data constructors to finite, but open, domains of types. Each type in such a domain must be associated with a value tag that is then stored at run time to allow it to serve as witness in a case expression.

Thomas Gorry, University of Liverpool
The evacuation problem: group search on the line

This talk will consider the Group Search Problem, or Evacuation Problem, in which K mobile entities located on the line perform a search for a specific destination. The mobile entities are initially placed at the same point (origin) on the line and the target is located at some unknown distance (d) either to the left or to the right of the origin. All mobile entities must simultaneously occupy the destination, and the goal is to minimize the time necessary for this to happen. The problem where $K = 1$ is called the cow-path problem, and the complexity of this is know to be $9d$ in the worst case (when the cow moves at unit speed), it is also known that this is the case for $K \geq 1$ mobile entities travelling at unit speed. This talk presents a clear argument for this claim as well examining the case when $K = 2$ mobile entities with different speeds, showing a surprising result that the bound of $9d$ can still be achieved when 1 mobile entity has unit speed and the other moves with speed at least $1/3$.

Ivaylo Hristakiev, University of York
Analysing graph programs for confluence

The graph programming language GP, developed at York, is an experimental domain-specific language for high-level problem solving on graphs and graph-like structures. In general, graph programs are highly nondeterministic because of their rule-based nature. However, a special case on nondeterminism called confluence ensures the functional behaviour of the execution. Confluence detection is done through construction of critical pairs, which represent conflicts in minimal

context. This technique has been extended to several variations of graph transformation, but not to GP. In this talk, I will present what these extensions are together with their associated issues and also report on ongoing work on static confluence checking of GP programs.

Augustine Kwanashie, University of Glasgow
Profile-based optimal matchings in the student/project allocation problem

In the *Student/Project Allocation problem (SPA)* we seek to assign students to group or individual projects offered by lecturers. Students are required to provide a list of projects they find acceptable in order of preference. Each student can be assigned to at most one project and there are constraints on the maximum number of students that can be assigned to each project and lecturer. A matching in this context is a set of student-project pairs that satisfies these constraints.

We seek to find matchings that satisfy optimality criteria based on the *profile* of a matching. This is a vector whose ith component indicates the number of students obtaining their ith-choice project. Various profile-based optimality criteria have been studied. For example, one matching M_1 may be preferred to another matching M_2 if M_1 has more students with first-choice projects than M_2.

In this talk we present an efficient algorithm for finding optimal matchings to SPA problems based on various well known profile-based optimality criteria. We model SPA as a network flow problem and describe a modified augmenting path algorithm for finding a maximum flow which can then be transformed to an optimal SPA matching. This approach allows for additional constraints, such as project and lecturer lower quotas, to be handled flexibly without modifying the original algorithm.

Karoliina Lehtinen, University of Edinburgh
Syntactic and semantic complexity in modal μ

The modal μ calculus is a temporal logic evaluated on labelled transition systems. It combines next-state modalities with greatest and least fixpoint operators, resulting in a logic capable of expressing both finite and infinite behaviour such as reachability, safety, eventual safety and much more. In particular, it subsumes many temporal logics such as LTL and CTL. Despite its high expressiveness, the core algorithmic problems around modal μ remain decidable: the model-checking problem is widely conjectured to be in P and satisfiability is EXPTIME-complete. This makes modal μ a widely studied formalism for program verification.

Modal μ's expressiveness is based on a simple but productive syntax: by increasing the number of alternations between greatest and least fixpoint operators, modal μ can express properties of increasing complexity. However, formulas of large alternation depth can also express much simpler properties and currently we lack the tools to differentiate between inherent and accidental complexity. Given

that the current best model checking algorithms are exponential in a function of the alternation depth of a formula, deciding whether a formula can be expressed with fewer alternations remains one of the main open problems surrounding modal μ.

This talk presents work on identifying and simplifying non-strict formulas, that is to say formulas that are equivalent to a formula with fewer alternations. The strictness of a formula implies the satisfiability of a set of derived formulas describing systems that witness the necessity of each alternation. If these witnesses do not exist for some formula, there are syntactic transformations which yield a formula of lower alternation depth.

Hsiang-Hsuan Liu, University of Liverpool
Scheduling for electricity cost in smart grid

We study an online scheduling problem arising in demand response management in smart grid. Consumers send in power requests with a flexible set of timeslots during which their requests can be served. For example, a consumer may request the dishwasher to operate for one hour during the periods 8am to 11am or 2pm to 4pm. The grid controller, upon receiving power requests, schedules each request within the specified duration. The electricity cost is measured by a convex function of the load in each timeslot. The objective of the problem is to schedule all requests with the minimum total electricity cost. As a first attempt, we consider a special case in which the power requirement and the duration a request needs service are both unit-size. For this problem, we present a polynomial time online algorithm that gives an optimal solution and show that the time complexity can be further improved if the given set of timeslots is a contiguous interval.

Iain McBride, University of Glasgow
Modelling practical placement of trainee teachers to schools

Several countries successfully use centralized matching schemes to assign students to study places or recent graduates to their first positions in a labour market. In this work we describe a model motivated by specific features of the Slovak and Czech education systems where each recently graduated trainee teacher specializes in a small number of subjects, each school has an overall capacity and further each school has partial capacities with respect to each of the available subjects. We show that the problem is unlikely to be efficiently solvable even under severe restrictions on the total number of subjects available, the partial capacities of schools for the available subjects and the number of acceptable schools each trainee teacher may list. Since these results suggest an efficient method of producing optimal solutions is unlikely, we present an integer programming model for finding a maximum cardinality matching in an instance of the teachers assignment problem and we present the results of the application of this IP model to real data

from the allocation process for allocating trainee teachers in Slovakia.

This is joint work with Tamás Fleiner (Budapest University of Technology and Economics), Katarína Cechlárová (P.J. Šafárik University, Košice, Slovakia), David Manlove (University of Glasgow) and Eva Potpinková (P.J. Šafárik University, Košice, Slovakia).

Markus Pfeiffer, University of St Andrews
The rational hierarchy of semigroups

The word problem for semigroups is known to be undecidable in general. On the other hand, deciding the word problem of the natural numbers or the integers is simple. My research focuses on finding classes of semigroups with word problem decidable by different types of automata. In this talk I will introduce what I call the rational hierarchy of semigroups, semigroups that have word problem decidable by asynchronous, two-tape, finite state automata, and conjunctions, Boolean combinations of such automata.

Jean Jose Razafindrakoto, Swansea University
Provably total search problems in fragments of bounded arithmetic below polynomial-time

In bounded arithmetic, a host of theories have been developed and which correspond to complexity classes within the polynomial hierarchy and below polynomial-time (see Cook and Nguyen's monograph "Logical Foundations of Proof Complexity, Cambridge University Press, New York, NY, USA, first ed., 2010", for an overview). Recent research tries to characterize the provably total NP search problems in such theories, where a total NP search problem is provably total in a theory $\mathcal{T}$ if it can be formalized in the language of $\mathcal{T}$ and $\mathcal{T}$ can prove that for each instance, there exists a solution to the search problem.

Given a class S of provably total NP search problems for some theory, the general aim of our research project is to identify some specific provably total NP search problem class (usually defined via some specific combinatorial principle) which is complete within S under AC^0-many-one reduction; completeness should be proven using AC^0-reasoning only. For the theory related to polynomial-time, we identify the search problem class Inflationary Iteration (*IITER*) which serves our above described aim. A function F (defined on finite strings) is inflationary if X is a subset of $F(X)$ (under the natural identification of strings with finite sets). An IITER principle is defined as a special case of the iteration principle, in which the iterated function has to be AC^0-computable and inflationary.

Cook and Nguyen have a generic way of defining a bounded arithmetic theory VC for complexity classes C below polynomial-time. For such a theory VC, we define a search problem class $KPT[C]$ which serves our above described aim. These problems are based on a version of Herbrand's theorem, proven by Kra-

jíček, Pudlak and Takeuti in "Bounded arithmetic and the polynomial hierarchy, Ann. Pure Appl. Logic, 52(1-2):143-153, 1991, International Symposium on Mathematical Logic and its Applications (Nagoya, 1988)".

This is joint work with Arnold Beckmann.

Paolo Serafino, Teesside University
Heterogeneous facility location without money

Mechanism Design is a novel research field mainly concerned with optimization problems that have to operate under the assumption that their input is distributed across *selfish* agents. In this setting, *mechanisms* (i.e., typically *allocation algorithms*) have to elicit their input from the agents and have to ensure (usually via suitable payment functions) that agents report *truthfully* the part of input they possess. The challenge faced in this setting is that agents are not reliable, in the sense that they can misreport their private information. Alas, it is often the case that monetary transfers between the mechanism and the agents cannot be performed. Motivated by this kind of considerations, Procaccia and Tennenholz (*Approximate Mechanism Design Without Money*, EC09) proposed the research agenda of approximate mechanism design without money, which aims at leveraging approximation, instead of payments, as a means to enforce truthfulness. In this line of work, the simple yet general and elegant problem of facility location has attracted much interest. The model which is typically considered therein features single-parameter agents (i.e., agents whose type is a single number encoding their position on a real line). In the wake of this line of research, we formulate and initiate the study of *heterogeneous facility location without money*, a problem akin to the traditional facility location problem but featuring multi-parameter agents. More specifically, we study truthful mechanisms without money for the problem in which *heterogeneous* facilities (facilities serving different purposes) have to be located and agents are only interested in some of them. We study the approximation ratio that can be achieved by truthful mechanisms in this setting, deriving some approximation bounds which make a surprising parallel with our knowledge of truthfulness for the classical single-dimensional facility location problem.

Yiannis Siantos, King's College London
Inferring network properties and embedded structure using random walks

We study the use of random walks to estimate global properties of graphs, for example the number of edges, vertices, triangles, and generally, the number of small fixed subgraphs. We consider two methods for this based on first returns of random walks: the cycle formula of regenerative processes and random walks with weights based on the property under consideration. In addition we use these methods to infer the embedded structure of graphs, such as whether a pre-defined subset of vertices is better connected internally than the rest of the graph. We dis-

cuss theoretical foundations for both methods and present experimental results on the rate of convergence of all the estimates. Both theory and experiments highlight the importance of high-weight vertices for the efficiency of either method.

Rob van Stee, University of Leicester
An optimal online bin packing algorithm

In the online bin packing problem, items of size at most 1 arrive one by one and need to be packed into bins of size 1 without knowledge of future items. We measure the performance of algorithms for this problem by comparing the number of bins used to the optimal number of bins. The competitive ratio of an algorithm is the highest possible ratio between these numbers (i.e., for all possible inputs).

We present an online bin packing algorithm with absolute competitive ratio $5/3$, which is best possible. The previous best known algorithms for this problem were Best Fit and First Fit, which were only recently shown to be exactly 1.7-competitive.

Anthony Stewart, Durham University
Parallel knock-out schemes for special graph classes

We consider parallel knock-out schemes for graphs. These schemes proceed in rounds. In the first round each vertex in the graph selects exactly one of its neighbours, and then all the selected vertices are eliminated simultaneously. In subsequent rounds this procedure is repeated in the subgraph induced by those vertices not yet eliminated. The scheme continues until there are no vertices left, or until an isolated vertex is obtained (since an isolated vertex will never be eliminated). A graph is reducible if there exists a parallel knock-out scheme that eliminates every vertex in the graph (for instance a graph that has a Hamilton cycle is reducible within one round). The Parallel Knock-Out problem is that of deciding whether a graph is reducible. This problem is known to be NP-complete. We discuss known results for this problem together with a number of new results for special graph classes (such as split graphs).

Nihan Tokac, Durham University
Fixed parameter tractability of hybridization number and rooted subtree prune and regraft distance

The decision problems computing hybridization number and rooted subtree prune and regraft distance are important to understand and model reticulation events in evolutionary biology. In this paper, we show that computing hybridization number is fixed parameter tractable when the parameter is the minimum level of network on T and T'. As well as, computing rooted subtree prune and regraft distance between two rooted binary phylogenetic trees on the same label set is fixed parameter tractable when the parameter is the minimum rSPR-level of network on T

and T'.

Chalita Toopsuwan, King's College London
Maximal anti-exponent of gapped palindromes

A palindrome is a string that reads the same backward and forward. We consider gapped palindromes which are words of the form $uv\tilde{u}$ for some words u, v with $|v| \geq 2$ where $\tilde{u}$ denotes the reversal of u. Mimicking the standard notion of string exponent, we define the antiexponent of a gapped palindrome $uv\tilde{u}$ as the quotient of $|uv\tilde{u}|$ over $|uv|$. We apply techniques based on the use of a suffix automaton and on the reversed Lempel-Ziv factorisation to an input string y containing no ordinary palindrome, and design an algorithm to compute the maximal anti-exponent of gapped palindromes of the string. Our algorithm runs in linear-time on a fixed-size alphabet in contrast to a naive cubic time solution.

William Whistler, Durham University
The counting complexity of planar graph homomorphism problems

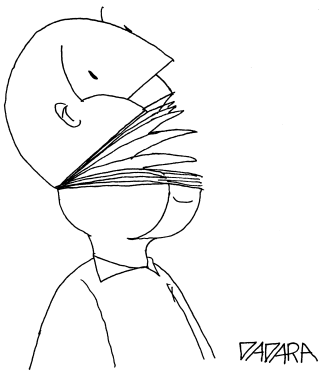
In this talk I present my current progress on classifying the counting complexity of graph homomorphism problems with inputs restricted to planar graphs.

Michele Zito, University of Liverpool
Relaxation and rounding for appliance allocation in the smart grid

We introduce a scheduling algorithm for a set of air-conditioners deployed in a building whose electricity comes from the grid as well as from renewable sources. The main objective of this study is to introduce a heuristic algorithm that is able to reduce electricity bills, keeping the temperature within comfort levels in the building, and maximizing the utilization of domestic renewable power. The algorithm uses relaxation in order to convert a Mixed Integer Linear Program into an LP problem and then a rounding mechanism to increase the utilization of domestic renewable power by scheduling the load on times where there is enough renewable power even this period is peak hours in terms of cost.

(This is joint work with M. Arikiez, A. Fernandez Anta, F. Grasso, and D. Kowalski.)

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

BEATCS no 113

BOOLEAN FUNCTION COMPLEXITY

ADVANCES AND FRONTIERS

Stasys Jukna*

> Go to the roots of calculations! Group the operations. Classify them according to their complexities rather than their appearances! This, I believe, is the mission of future mathematicians.
>
> – Evariste Galois

What it is all about?

My book [5] is all about proving lower bounds.

Roughly speaking, research in Computational Complexity has two tightly interconnected strands. One of these strands—*structural complexity*—deals with high-level complexity questions: is space a more powerful resource than time? Does randomness enhance the power of efficient computation? Is it easier to verify a proof than to construct one? So far we do not know the answers to any of these questions; thus most results in structural complexity are *conditional* results that rely on various unproven assumptions.

My book [5] is about the life on the second strand—*circuit complexity*. Inhabitants of this strand deal with establishing *unconditional* lower bounds on the computational complexity of specific problems, like multiplication of matrices or detecting large cliques in graphs. This is essentially a low-level study of computation; it typically centers around particular models of computation such as decision trees, branching programs, boolean formulas, various classes of boolean circuits, communication protocols, proof systems and the like.

Why yet another book?

More than twenty years have passed since the well-known books on circuit complexity by Savage (1976), Nigmatullin (1983), Wegener (1987) and Dunne (1988)

*Goethe University Frankfurt, Germany, and Institute of Mathematics and Informatics, Vilnius University, Lithuania. jukna@thi.informatik.uni-frankfurt.de. Research supported by the DFG grant SCHN 503/6-1.

as well as a famous survey paper of Boppana and Sipser (1990) were written. Albeit in the meanwhile some excellent books in computational complexity appeared—including those by Savage (1998), Goldreich (2008) and Arora and Barak (2009)—these were mainly about the life on the first strand—structural complexity. So, it was the time to collect the new developments in circuit complexity during these two decades.

Almost everything is complex

It is known for now more than 70 years that most boolean functions require circuits of exponential size. In particular, Shannon, Lupanov and his students even established the following tight asymptotic for the maximum $\{\wedge, \vee, \neg\}$-circuit complexity $\mathrm{C}(n) = \max_f \mathrm{C}(f)$ of a boolean function of n variables:

$$1 + \frac{\log_2 n}{n} - O\Big(\frac{1}{n}\Big) \le \mathrm{C}(n) \cdot \frac{n}{2^n} \le 1 + \frac{\log_2 n}{n} + \frac{\log_2 \log_2 n}{n} + O\Big(\frac{1}{n}\Big)$$

Using these estimates, one can, say, easily prove the "Circuit Hierarchy Theorem": for every $n \le t(n) \le 2^{n-2}/n$, there are boolean functions computable by circuits of size $4t$, but having no circuits of size t. In a similar vein is the result that, for every $k \ge 1$, there exist boolean functions f_n of DNF-size n^{2k+1} such that $\mathrm{C}(f_n) > n^k$.

Unfortunately, these (and many other) results only show a mere *existence* of hard boolean functions. An ultimate goal of circuit complexity, however, is to exhibit such hard functions, and to understand *why* they are hard. Say, why the threshold function (does a given graph have k edges) is "simple", whereas the clique function (does a given graph has a clique with k edges) is "hard". And here, as in many other fields of mathematics—where the question comes to *construct* particular objects—the situation is much worse: the strongest known lower bounds on the unrestricted $\{\wedge, \vee, \neg\}$-circuit complexity of explicit boolean functions remain of the form cn for some small constants c; the current record remains $c = 5$.

Strong (even exponential) lower bounds were only obtained for various restricted circuit models. Below I give a rough overview of the book's contents.

Forget what was done: Formulas

Formulas are $\{\wedge, \vee, \neg\}$-circuits whose underlying graphs are trees. That is, these are the circuits without any memory: if we want to use some already computed (by a sub-formula) function g in another place, we are forced to re-compute g again. Some results:

- Formulas can be balanced: if f can be computed by a formula of size $\mathrm{L}(f)$, then f can be computed by a formula of depth $\mathrm{D}(f) \le 1.73 \log_2 \mathrm{L}(f)$. For

circuits, we only know that if f can be computed by a circuit of size S, then f can be also computed by a circuit of depth $O(S/\log S)$.

- The maximum of $D(f)$ over all boolean functions f of n variables is asymptotically equal $n - \log_2 \log_2 n$.

- If a boolean function f can be computed by a depth-d $\{\wedge, \vee, \neg\}$-formula using *unbounded* fanin AND and OR gates and having S leaves, then $D(f) \leq d - 1 + \lceil \log_2 S \rceil$. Note that a trivial upper bound, obtained by simulating each gate by a binary tree, is only $D(f) = O(d \log S)$.

- The depth of a circuit (or formula) is equal to the communication complexity of the following "find a separating bit" gate: Alice gets a vector $a \in f^{-1}(1)$, Bob gets a vector $b \in f^{-1}(0)$, and their goal is to find a bit $i \in [n]$ such that $a_i \neq b_i$.

- Khrapchenko's lower bound: Form a bipartite graph G_f with parts $f^{-1}(1)$ and $f^{-1}(0)$ by drawing an edge (a, b) if and only if a and b differ in exactly one bit. Then $L(f)$ is at least the product of the average degrees of the left and right parts of the graph G_f. This gives the lower bound $L(\oplus_n) \geq n^2$ for the Parity function $\oplus_n(x) = x_1 \oplus x_2 \oplus \cdots \oplus x_n$.

- There were many attempts to extend Khrapchenko's measure to obtain larger lower bounds. His measure is sub-modular and convex. It turned out, however, that no sub-modular or convex complexity measures can break down this quadratic barrier.

- A weaker bound $L(\oplus_n) = \Omega(n^{3/2})$ was earlier proved by Subbotovskaya by inventing the method of *random restrictions*. Currently, this method is widely used, in particular, to prove lower bounds for constant-depth circuits and communication protocols.

- When properly applied, Subbotovskaya's approach yields up to $\Omega(n^{3-o(1)})$ lower bounds, and this is a current record for $\{\wedge, \vee, \neg\}$-formulas.

- Other lower-bounds arguments for formulas are known as well: the method of "universal" functions for formulas where all binary boolean functions are allowed as gates, the method based on graph entropy, and the relation of formula size with the affine dimension of graphs.

- Lower bounds for *monotone* formulas were obtained by using rank as well as communication complexity arguments. In particular, rank arguments give tight superpolynomial lower bounds $n^{\Theta(\log n)}$ for functions induced by Paley graphs, as well as tight lower bounds for monotone quadratic functions.

Communication complexity arguments yield lower bounds $n^{\Theta(\log n)}$ even for such "simple" boolean functions as the *s-t* connectivity function.

- Superpolynomial lower bounds $n^{\Theta(\log n)}$ were also obtained for so-called monotone *span programs*, a model which may be even exponentially more powerful than monotone formulas. A span program for a boolean function $f(x_1, \ldots, x_n)$ is a 0/1 matrix whose rows are labeled by literals (variables and their negations); one literal can label several rows. A program is monotone if there are no negated labels. When an input $a \in \{0,1\}^n$ arrives, all rows whose labels are inconsistent with a are removed, and input a is accepted if the remaining rows span the all-1 vector over $GF(2)$.

Forbid negations: Monotone circuits

These are circuits with fanin-2 AND and OR gates, but no NOT gates. Despite of its seeming "simplicity", this model resisted any attempts to prove larger than linear lower bounds.

- The situation changed in 1985-86 when Razborov came with his "method of approximations", and proved a super-polynomial lower bound for the clique function. After that some modifications and extensions of his method were suggested. Razborov approximated gates by monotone DNFs and used the Sunflower Lemma of Erdős and Rado to convert CNFs to DNFs.

- Later, Sipser's notion of "finite limits" and a monotone Switching Lemma have led to a symmetric version of Razborov's argument, where both DNFs and CNFs are used to approximate gates (a two-side approximation). This resulted into the following general lower bounds criterion: if a monotone boolean function has a monotone circuit of size t, then it is *t-approximable*.

  Being t-approximable mean that there exist integers $2 \le r, s \le n$, a monotone s-CNF $C(x)$, a monotone r-DNF $D(x)$, and a subset $I \subseteq [n]$ of size $|I| \le s-1$ such that $|C| \le t \cdot (r-1)^s$, $|D| \le t \cdot (r-1)^s$, and either $C \le f$ or $f \le D \vee \bigvee_{i \in I} x_i$ hold. Important here is that the s-CNF C has only $t \cdot (r-1)^s$ out of all possible $\binom{n}{s}$ clauses, and similarly for the r-DNF D.

- This criterion holds even when any monotone *real-valued* functions $g : \mathbb{R}^2 \to \mathbb{R}$ are allowed as gates, and enables one to obtain in a uniform way strong lower bounds for a full row of explicit boolean functions. Together with appropriate interpolation theorems, these bounds have also led to the first exponential lower bounds for the length of the cutting-plane proofs.

- Why should one care about monotone circuits? The point is that this model has a purely "practical" importance. Namely, lower bounds for such circuits imply the same lower bounds for (min, +)-circuits, and hence, for dynamic programming. In this respect, our knowledge about the power of monotone circuits remains unsatisfying. Say, we still cannot prove that the *s*-*t* connectivity or even the connectivity function require circuits of size $\Omega(n^3)$. Known dynamic programming algorithms give circuits of size $O(n^3)$ for both these functions.

- It is known that there are monotone boolean functions f (like the Perfect Matching function) that can be computed by non-monotone circuits of polynomial size, but any monotone circuit for them must have a super-polynomial number of gates. This rises a question about the role of NOT gates.

- A classical result of Markov implies that $M(n) = \lceil \log_2(n+1) \rceil$ NOT gates are enough to compute any boolean function of n variables.

- Fisher and other authors substantially improved this by showing that restricting the number of NOT gates to $M(n)$ can only increase the size of a circuit by only an additive factor of $O(n \log^2 n)$.

- It is also know that the Markov–Fisher bound can almost be reached: there are explicit monotone (multi-output) boolean functions f which have polynomial size circuits only if more than $M(n) - O(\log \log n)$ NOT gates are used.

Restrict the time: Bounded-depth circuits

Yet another possibility to "bind circuits hands" is to allow NOT gates as well as AND and OR gates of unbounded fanin, but to restrict the depth (parallel time) of the circuit. This model is known as AC^0-circuits ("alternating circuits of constant depth"), and is currently quite intensively investigated.

- AC^0 circuits were considered by many authors since early 80's. When dealing with them, two major techniques emerged: the depth-reduction method via appropriate versions of the Switching Lemma, as well as approximation by low-degree polynomials.

- The depth-reduction argument has led to a tight $2^{\Theta(n^{1/(d-1)})}$ lower bound on the size of depth-d circuits computing the Parity function. Moreover, this function cannot even be approximated by such circuits of polynomial size.

- The approximation by low-degree polynomials argument has led to an exponential lower bound $2^{\Omega(n^{1/2d})}$ for the Majority function, even if Parity functions are also allowed as gates. Similar lower bounds were also proved when instead of Mod-2 gates, arbitrary prime-modulo functions are allowed as gates.

- The case of arbitrary, including *composite* modulo gates remains open. The class of boolean functions computable by such circuits of polynomial size is usually denoted by ACC^0. Still, Williams (2011) has recently shown that $\mathrm{NEXP} \not\subseteq \mathrm{ACC}^0$. This wakes a hope that we will be able to expose a boolean function $f \notin \mathrm{ACC}^0$ lying in NP or even in P. Actually, the Majority function still remains as a possible candidate.

- Even AC^0-circuits of depth-3 are interesting: by the results of Valiant, any lower bound $2^{\phi(n)}$ with $\phi(n) \gg n/\log\log n$ would give an example of a boolean function which cannot be computed by a linear size (fanin-2) circuit of logarithmic depth; proving such a bound is now a more than 30 years old open problem, and no such bound is known even for $\{\oplus, 1\}$-circuits.

- Known lower bounds for depth-3 circuits are only of the form $2^{\Omega(\sqrt{n})}$, and can be obtained using so-called "finite limits" and quite simple combinatorics. If we require that the circuit must have parity gates (instead of OR gates) at the bottom (next to the inputs) level, then arguments of graph complexity allow us to prove even truly exponential lower bounds $2^{\Omega(n)}$. Unfortunately, Valiant's construction does not carry over such circuits.

- Motivated by neural networks, people have also considered circuits with threshold gates. A boolean function $f(x_1, \ldots, x_n)$ is a threshold function if there exist real numbers $w_0, w_1, \ldots, w_n$ such that for every $x \in \{0,1\}^n$, $f(x) = 1$ if and only if $w_1x_1 + \cdots + w_nx_n \geq w_0$. For unbounded-depth circuits with threshold functions as gates, only linear lower bounds are known. Even depth-3 is here not well understood. Exponential lower bounds are only known for depth-2 circuits.

Restrict the time, but allow omnipotent power

In *general* circuits, *arbitrary* boolean functions are allowed as gates. The *size* of such a circuit is defined as the total number of wires (rather than gates). Of course, then every single-output boolean function f of n variables can be computed by a circuit of size n: just take one gate—the function f itself. The problem, however, becomes nontrivial if instead of one function, we want to *simultaneously* compute m boolean functions $f_1, \ldots, f_m$ on the same set of n variables $x_1, \ldots, x_n$, that is,

to compute an (n,m)-*operator* $f : \{0,1\}^n \to \{0,1\}^m$. Note that in this case the phenomenon which causes complexity of circuits is *information transfer* instead of *information processing* as in the case of circuits computing a single function.

- It is clear that every (n,m)-operator can be computed using nm wires, even in depth 1. However, already circuits of depth 2 constitute a rather non-trivial model: any operator with $\omega(n^2/\log\log n)$ depth-2 wire complexity also cannot be computed by linear-size, logarithmic-depth boolean circuits of fanin 2.
- The strongest known lower bounds for depth-2 are of the form $\Theta(n^{3/2})$, and were proved for natural operators like the product of two $0/1$ $\sqrt{n} \times \sqrt{n}$ matrices over $GF(2)$. These bounds were proved using particular entropy arguments.
- A lot of work was done when trying to prove strong lower bounds for general depth-2 circuits computing *linear* operators $f_A(x) = Ax$ over $GF(2)$. Lower bounds for such operators are usually derived using appropriate algebraic arguments (matrix rigidity) as well as graph-theoretic arguments (various superconcentration properties of graphs).
- The strongest known lower bound for linear operators f_A in depth 2 is about $n \cdot \phi(n)^2$ where $\phi(n) = (\ln n)/(\ln\ln n)$. The lower bound is proved using superconcentration properties. Unfortunately, it is known that such arguments cannot yield larger than $n \ln^2 n/\ln\ln n$ lower bounds. Interestingly, the *upper* bounds for these operators are proved in the class of linear circuits, i.e. depth-2 circuits with only Parity gates. In fact, the question on whether non-linear gates can help to compute linear operators over $GF(2)$ remains widely open.

Allow only to branch and join: Branching programs

Decision trees constitute one of the "simplest" models of computation, and a lot of interesting results were proved for it. Just to mention some of them:

- P = NP∩co-NP holds for decision tree *depth*; this is proved using elementary combinatorics.
- P ≠ NP ∩ co-NP holds for decision tree *size*; this is proved using spectral arguments.
- The depth of decision trees is related to sensitivity and block-sensitivity of the computed functions, as well as to the degree of their representation as polynomials.

- Non-trivial depth lower bounds are also known when arbitrary real threshold functions (not just $x_i \geq 1$) are used as decision predicates. In particular, the Inner Product function requires depth $n/2$ even in this generalized model.

The model of branching programs (BP) is a generalization of decision trees: the underlying graph may now be an arbitrary acyclic graph (not just a tree). The size here is the number of edges.

- For unrestricted BPs the progress was rather minor: the strongest lower bounds remain $\Omega(n^2/\log^2 n)$ for deterministic, and $\Omega(n^{3/2}/\log n)$ for nondeterministic BPs, both proved more than 40 years ago by Nechiporuk.
- For *symmetric* boolean functions, Nechiporuk's argument cannot yield any super-linear lower bounds. Such bounds were proved using more subtle arguments by many authors around 1990.
- One of the most surprising results for general BPs is the theorem of Barrington stating that branching programs of width-5 are not much weaker than formulas.
- Exponential lower bounds for BPs were proved only when either each variable can be re-tested constant times along each computation path, or when at most cn variables for a sufficiently small constant $c > 0$ are allowed to be tested more than once along each computation. The arguments here use a rather non-trivial combinatorics, probabilistic arguments as well as expander graphs.
- Still, the situation even with restricted BPs remains rather unsatisfying. In particular, we are still unable to prove any strong lower bounds for the following one of the simplest non-deterministic models of "almost read-once" BPs: these are nodeterministic BPs where every consistent paths must be read-once (no variable can be tested more than once). The problem here is that we have no restrictions on inconsistent paths (those containing contradictory tests $x_i = 0$ and $x_i = 1$ on some variable).

Allow only to chat: Communication complexity

Since communication complexity has a comprehensive treatment in an excellent book by Kushilevitz and Nisan of 1997, we have restricted ourselves to results essentially used later in our book, as well to some newer results. In particular, we describe the progress concerning the so-call "rank-conjecture", prove that P = NP ∩ co-NP holds for fixed-partition games, whereas P ≠ NP ∩ co-NP holds for best-partition games, present lower bounds on randomized protocols, and Forster's (2002) celebrated lower bound on the sign-rank of ±1 matrices.

Applications: Proof complexity

The last two chapters of the book are devoted to some applications of the previous results when lower-bounding the length of resolution and cutting-planes proofs. The point is that so-called regular resolution proofs are, in fact, read-once branching programs solving particular search problems (find an unsatisfied clause in the given CNF). On the other hand, the length of cutting-plane proofs can be lower-bounded using some communication complexity arguments or using the interpolation theorem together with lower bounds on the size of monotone circuits with real-valued gates.

What's new: Some features

- The book discusses some topics, like graph complexity or method of finite limits, that are not known well enough even for specialists in circuit complexity.
- Gives new proofs of classical results, like lower bounds for monotone circuits, monotone span programs and constant-depth circuits.
- Presents some topics never touched in existing complexity books, like graph complexity, span programs, bounds on the number of NOT gates, bounds on Chvátal rank, lower bounds for circuits with arbitrary boolean functions as gates, etc.
- Relates the circuit complexity with one of the "hottest" nowadays topics – the proof complexity.
- Contains more than 40 specific open problems, two of which were already re-solved after the book was published.
- The main feature, however, is the inclusion of many results of Russian mathematicians which remained unknown in the West. Just to give an example, the following result proved by Lupanov already in 1956 was later re-discovered by many authors (with much more involved proofs): every bipartite $n \times m$ graph can be decomposed into edge-disjoint bipartite cliques so that the sum of their nodes does not exceed $(1 + o(1))nm/\log_2 n$.

Epilogue

At the end of the book, I shortly sketch some stuff not discussed in the main text: pseudo-random generators, natural proofs, the fusion method for proving

lower bounds, and indirect (diagonalization) arguments. The Appendix contains all necessary mathematics.

Acknowledgement

I am thankful to Sasha Razborov for his comments on this summary.

References

[1] S. Arora and B. Barak (2009): *Computational Complexity: A Modern Approach*. Cambridge University Press. www.cs.princeton.edu/theory/complexity/

[2] R. B. Boppana and M. Sipser (1990): The complexity of finite functions, in: Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity (A), 757-804.

[3] P. E. Dunne (1988): *The Complexity of Boolean Networks*. Academic Press Professional, Inc., San Diego, CA. http://cgi.csc.liv.ac.uk/~ped/RESUME.html

[4] O. Goldreich (2008): *Computational Complexity: A Conceptual Perspective*. Cambridge University Press. www.wisdom.weizmann.ac.il/~oded/cc.html

[5] S. Jukna (2012): *Boolean Function Complexity: Advances and Frontiers*. Springer-Verlag. The home page of the book with a suplementary material: www.thi.cs.uni-frankfurt.de/~jukna/boolean/

[6] R. G. Nigmatullin (1983): *The Complexity of Boolean Functions*. Izd. Kazansk. Univ. (Kazan University Press, in Russian).

[7] J. E. Savage (1976): *The Complexity of Computing*. Wiley, New York.

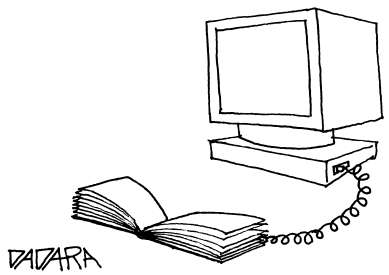
[8] J. E. Savage (1998): *Models of Computation: Exploring the Power of Computing*. Addison-Wesley. The book is freely available for download at:

http://www.cs.brown.edu/~jes/book/home.html

[9] I. Wegener (1987): *The Complexity of Boolean Functions*. Wiley-Teubner. The book is freely available for download at:

http://eccc.hpi-web.de/static/books/The_Complexity_of_Boolean_Functions/

Announcements



IFIP Summer School on Privacy and Identity Management for the Future Internet in the Age of Globalisation

Ninth International Summer School

organised jointly

by the IFIP Working Groups 9.2, 9.5, 9.6/11.7, 11.4, 11.6, Special Interest Group 9.2.2

IFIP Summer School on Privacy and Identity Management for the Future Internet in the Age of Globalisation

Patras University/Greece, 7-12 September 2014

In cooperation with the FP7 EU projects ABC4Trust, A4Cloud, FutureID, PRISMS, AU2EU

Introduction

Much research in privacy and identity in recent years has focused on the privacy issues associated with new technologies such as social media, cloud computing, big data, ubiquitous and ambient technologies. Due to the fact that many of these technologies operate on a global scale their use not only touches the countries where they originate (in many cases, the US), but individuals and groups around the globe.

We are especially inviting contributions from students who are at the stage of preparing either a master's or a PhD thesis. The school is interactive in character, and is composed of keynote lectures and workshops with master/PhD student presentations. The principle is to encourage young academic and industry entrants to the privacy and identity management world to share their own ideas, build up a collegial relationship with others, gain experience in making presentations, and potentially publish a paper through the resulting book proceedings. Students that actively participate, in particular those who present a paper, can receive a course certificate which awards 3 ECTS at the PhD level. The certificate can certify the topic of the contributed paper so as to demonstrate its relation (or non-relation) to the student's master's or PhD thesis.

The Summer School takes a holistic approach to society and technology and supports interdisciplinary exchange through keynote lectures, tutorials, workshops, and research paper presentations. In particular, participants' contributions that combine technical, legal, regulatory, socio-economic, social or societal, ethical, anthropological, philosophical, or psychological perspectives are welcome. The interdisciplinary character of the work is fundamental to the school. The research paper presentations and the workshops have a particular focus on involving students, and on encouraging the publication of high-quality, thorough research papers by students/young researchers. To this end, the school has a two-phase review process for submitted papers. In the first phase submitted papers (short versions) are reviewed and selected for presentation at the school. After the school, these papers can be revised (so that they can profit from their discussion at the school) and are then reviewed again for selection into the school's proceedings which will be published by Springer. Of course, submissions by senior researchers and European, national, or regional/community research projects are also very welcome.

IFIP Summer School on Privacy and Identity Management for the Future Internet in the Age of Globalisation

Contributions

The school seeks contributions in the form of research papers, tutorials, and workshop proposals from all disciplines (e.g., computer science, economics, ethics, law, psychology, sociology and other social sciences).

Topics of interest include, but are not limited to:

- data breaches and cybercrime,
- data retention and law enforcement,
- impact of legislative or regulatory initiatives on privacy,
- impact of technology on social exclusion/digital divide/social and cultural aspects,
- privacy and identity management (services, technologies, infrastructures, usability aspects, legal and socio-economic aspects),
- privacy by design and privacy by default,
- privacy-enhancing technologies (PETs),
- privacy issues and PETs relating to eIDs, social networks, crowdsourcing, big data analysis biometrics, and cloud computing, social computing,
- privacy standardisation,
- profiling and tracking technologies,
- semantic web security and privacy,
- social accountability and ethics,
- surveillance and privacy and identity management,
- surveillance and sensor networks,
- transparency-enhancing technologies (TETs),
- trust management and reputation systems.

Submissions

All submissions must be made in PDF format using the Easychair system.

Important dates and other information

Extended abstracts or short papers (> 2,000 words in Springer LNCS format, PDF)	25 May 2014
Notification of acceptance:	6 June 2014
Short paper (up to 8 pages) for pre-proceedings:	1 August 2014
Final paper:	28 November 2014
Notification of acceptance of the final paper:	30 January 2015
Summer School Website:	http://ifip2014.cti.gr/

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 Nerode Award (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."

(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
- Copenhagen, Denmark 2014
- 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

(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 (Wien), J. Hromkovic (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 G. Ausiello (Rome) and D. Sannella (Edinburgh).

ADDITIONAL EATCS INFORMATION

For further information please visit `http://www.eatcs.org`, or contact the President of EATCS:
Prof. Dr. Luca Aceto,
School of Computer Science
Reykjavik University
Menntavegur 1 IS-101 Reykjavik, Iceland
Email: `president@eatcs.org`

EATCS MEMBERSHIP

DUES

The dues are € 30 for a period of one year (two years for students). 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, a subscription form can be downloaded from `www.eatcs.org` to

be filled and sent together with the annual dues (or a multiple thereof, if membership for multiple years is required) to the **Treasurer** of EATCS:

Prof. Dr. Dirk Janssens,
University of Antwerp, Dept. of Math. and Computer Science
Middelheimlaan 1, B-2020 Antwerpen, Belgium
Email: `treasurer@eatcs.org`, *Tel: +32 3 2653904, Fax: +32 3 2653777*

The dues can be paid (in order of preference) by VISA or EUROCARD/MASTERCARD credit card, by cheques, or convertible currency cash. Transfers of larger amounts may be made via the following bank account. Please, add € 5 per transfer to cover bank charges, and send the necessary information (reason for the payment, name and address) to the treasurer.

Fortis Bank, Jules Moretuslei 229, B-2610 Wilrijk, Belgium
Account number: 220–0596350–30–01130
IBAN code: BE 15 2200 5963 5030, SWIFT code: GEBABE BB 18A