Report from Dagstuhl Seminar 15171

Theory and Practice of SAT Solving

Edited by

Armin Biere[1], Vijay Ganesh[2], Martin Grohe[3], Jakob Nordström[4], and Ryan Williams[5]

1 **Universität Linz, AT,** biere@jku.at
2 **University of Waterloo, CA,** vganesh@uwaterloo.ca
3 **RWTH Aachen, DE,** grohe@informatik.rwth-aachen.de
4 **KTH Royal Institute of Technology, SE,** jakobn@kth.se
5 **Stanford University, US,** rrwilliams@gmail.com

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 15171 "Theory and Practice of SAT Solving". The purpose of this seminar was to explore one of the most significant problems in all of computer science, namely that of computing whether formulas in propositional logic are satisfiable or not. This problem is believed to be intractable in general (by the theory of *NP*-completeness). However, the last two decades have seen dramatic developments in algorithmic techniques, and today so-called SAT solvers are routinely and successfully used to solve large-scale real-world instances in a wide range of application areas.

A surprising aspect of this development is that the best current SAT solvers are still to a large extent based on methods from the early 1960s, which can often handle formulas with millions of variables but may also get hopelessly stuck on formulas with just a few hundred variables. The fundamental question of when SAT solvers perform well or badly, and what underlying mathematical properties of the formulas influence SAT solver performance, remains very poorly understood. Another intriguing aspect is that much stronger mathematical methods of reasoning about propositional logic formulas are known today, in particular methods based on algebra and geometry, and these methods would seem to have great potential based on theoretical studies. However, attempts at harnessing the power of such methods have conspicuously failed to deliver any significant improvements in practical performance.

This seminar gathered leading researchers in applied and theoretical areas of SAT and computational complexity to stimulate an increased exchange of ideas between these two communities. We see great opportunities for fruitful interplay between theoretical and applied research in this area, and believe that this seminar showed beyond doubt that a more vigorous interaction between the two has potential for major long-term impact in computer science, as well for applications in industry.

Seminar April 19–24, 2015 – http://www.dagstuhl.de/15171

1998 ACM Subject Classification I.2.3 Deduction and Theorem Proving, F.2.2 Nonnumerical Algorithms and Problems: Complexity of proof procedures, F.4.1 Mathematical Logic, D.2.4 Software/Program Verification: Formal methods

Keywords and phrases SAT, Boolean SAT solvers, SAT solving, conflict-driven clause learning, Gröbner bases, pseudo-Boolean solvers, proof complexity, computational complexity, parameterized complexity

Digital Object Identifier 10.4230/DagRep.5.4.98

Edited in cooperation with Marc Vinyals (KTH Royal Institute of Technology)

1 Executive Summary

Armin Biere
Vijay Ganesh
Martin Grohe
Jakob Nordström
Ryan Williams

This seminar brought together researchers working in the areas of applied SAT solving on the one hand, and in proof complexity and neighbouring areas of computational complexity theory on the other, in order to communicate new ideas, techniques, and analysis from both the practical and theoretical sides.

The goals of this endeavour are to better understand why modern SAT solvers work so efficiently for many large-scale real-world instances, and in the longer term to discover new strategies for SAT solving that could go beyond the present "conflict-driven clause-learning" paradigm and deliver substantial further gains in practical performance.

Topics of the Workshop

This seminar explored one of the most significant problems in all of mathematics and computer science, namely that of proving logic formulas. This is a problem of immense importance both theoretically and practically. On the one hand, it is believed to be intractable in general, and deciding whether this is so is one of the famous million dollar Clay Millennium Problems (the P vs. NP problem). On the other hand, today so-called SAT solvers are routinely and successfully used to solve large-scale real-world instances in a wide range of application areas (such as hardware and software verification, electronic design automation, artificial intelligence research, cryptography, bioinformatics, operations research, and railway signalling systems, just to name a few examples).

During the last 15–20 years, there have been dramatic – and surprising – developments in SAT solving technology that have improved real-world performance by many orders of magnitude. But perhaps even more surprisingly, the best SAT solvers today are still based on relatively simple methods from the early 1960s, searching for proofs in the so-called resolution proof system. While such solvers can often handle formulas with millions of variables, there are also known tiny formulas with just a few hundred variables that cause even the very best solvers to stumble. The fundamental question of when SAT solvers perform well or badly, and what underlying properties of the formulas influence SAT solver performance, remains very poorly understood. Other practical SAT solving issues, such as how to optimize memory management and how to exploit parallelization on modern multicore architectures, are even less well studied and understood from a theoretical point of view.

Another intriguing fact is that although other mathematical methods of reasoning are known that are much stronger than resolution in theory, in particular methods based on algebra and geometry, attempts to harness the power of such methods have failed to deliver any significant improvements in practical performance – indeed, such solvers often struggle even to match the performance of resolution-based solvers. And while resolution is a fairly well-understood proof system, even very basic questions about these stronger algebraic and geometric methods remain wide open.

We believe that computational complexity can shed light on the power and limitations on current and possible future SAT solving techniques, and that problems encountered in SAT solving can spawn interesting new areas in theoretical research. We see great potential for interdisciplinary research at the border between theory and practice in this area, and believe that more vigorous interaction between practitioners and theoreticians could have major long-term impact in both academia and industry.

Goals of the Workshop

A strong case can be made for the importance of increased exchange between the two fields of SAT solving on the one hand and proof complexity (and more broadly computational complexity) on the other. While the two areas have enjoyed some exchanges, it seems fair to say that there has been relatively low level of interaction, given how many questions would seem to be of mutual interest. Below, we try to outline some such questions that served as motivation for organizing this seminar. We want to stress that this list is far from exhaustive, and in fact we believe one important outcome of the seminar was to stimulate the process of uncovering other questions of common interest.

What Makes Formulas Hard or Easy in Practice for Modern SAT Solvers?

The best SAT solvers known today are based on the DPLL procedure, augmented with optimizations such as conflict-driven clause learning (CDCL) and restart strategies. The propositional proof system underlying such algorithms, resolution, is arguably the most well-studied system in all of proof complexity.

Given the progress during the last decade on solving large-scale instances, it is natural to ask what lies behind the spectacular success of CDCL solvers at solving these instances. And given that there are still very small formulas that resist even the most powerful CDCL solvers, a complementary interesting question is if one can determine whether a particular formula is hard or tractable. Somewhat unexpectedly, very little turns out to be known about these questions.

In view of the fundamental nature of the SAT problem, and in view of the wide applicability of modern SAT solvers, this seems like a clear example of a question of great practical importance where the theoretical field of proof complexity could potentially provide useful insights. In particular, one can ask whether one could find theoretical complexity measures for formulas than would capture the practical hardness of these formulas in some nice and clean way. Besides greatly advancing our theoretical understanding, answering such a question could also have applied impact in the longer term by clarifying the limitations, and potential for further improvements, of modern SAT solvers.

Can Proof Complexity Shed Light on Crucial SAT Solving Issues?

Understanding the hardness of proving formulas in practice is not the only problem for which more applied researchers would welcome contributions from theoretical computer scientists. Examples of some other possible practical questions that would merit from a deeper theoretical understanding follow below.

- Firstly, we would like to study the question of memory management. One major concern for clause learning algorithms is to determine how many clauses to keep in memory. Also, once the algorithm runs out of the memory currently available, one needs to determine which clauses to throw away. These questions can have huge implications for performance, but are poorly understood.

- In addition to clause learning, the concept of restarts is known to have decisive impact on the performance on modern CDCL solvers. It would be nice to understand theoretically why this is so. The reason why clause learning increases efficiency greatly is clear – without it the solver will only generate so-called tree-like proofs, and tree-like resolution is known to be exponentially weaker than general resolution. However, there is still ample room for improvement of our understanding of the role of restarts and what are good restart strategies.
- Given that modern computers are multi-core architectures, a highly topical question is whether this (rather coarse-grained) parallelization can be used to speed up SAT solving. Our impression is that this is an area where much practical work is being carried out, but where comparatively little theoretical study has been done. Thus, the first step here would consist of understanding what are the right questions to ask and coming up with a good theoretical framework for investigating them.

  While there are some successful attempts in parallelizing SAT, obtained speed-ups are rather modest. This is a barrier for further adoption of SAT technology already today and will be become a more substantial problem as thousands of cores and cloud computing are becoming the dominant computing platforms. A theoretical understanding on how SAT can be parallelized will be essential to develop new parallelization strategies to adapt SAT to this new computing paradigm.

Can we build SAT Solvers based on Stronger Proof Systems than Resolution?

Although the performance of modern CDCL SAT solvers is impressive, it is nevertheless astonishing, not to say disappointing, that the state-of-the-art solvers are still based on simple resolution. Resolution lies very close to the bottom in the hierarchy of propositional proof systems, and there are many other proof systems based on different forms of mathematical reasoning that are known to be strictly stronger. Some of these appear to be natural candidates for serving as a basis for stronger SAT solvers than those using CDCL.

In particular, proof systems such as polynomial calculus (based on algebraic reasoning) and cutting planes (based on geometry) are known to be exponentially more powerful than resolution. While there has been some work on building SAT solvers on top of these proof systems, progress has been fairly limited. As part of the seminar, we invited experts on algebraic and geometric techniques to discuss what the barriers are that stops us from building stronger algebraic or geometric SAT solvers, and what is the potential for future improvements. An important part of this work would seem to be to gain a deeper theoretical understanding of the power and limitations of these proof methods. Here there are a number of fairly long-standing open theoretical questions. At the same time, only in the last couple of years proof complexity has made substantial progress, giving hope that the time is ripe for decisive break-throughs in these areas.

Organization of the Workshop

The scientific program of the seminar consisted of 26 talks. Among these there were five 80-minute tutorials on core topics of the seminar:

- proof complexity (Paul Beame),
- conflict-driven clause learning (CDCL) SAT solvers (João Marques-Silva),

- proof systems connected to SAT solving (Sam Buss),
- preprocessing and inprocessing (Matti Järvisalo),
- SAT and SMT (Nikolaj Bjørner).

Throughout, the tutorials were well-received as a means of introducing the topics and creating a common frame of reference for participants from the different communities.

There were also nine slighly shorter survey talks of 50 minutes which were intended to give overviews of a number of important topics for the seminar:

- semialgebraic proof systems (Albert Atserias),
- pseudo-Boolean constraints and CDCL (Daniel Le Berre),
- Gröbner bases (Manuel Kauers),
- SAT-enabled verification of state transition systems, (Karem Sakallah),
- SAT and computational complexity (Ryan Williams)
- the (strong) exponential time hypothesis and consequences (Ryan Williams),
- SAT and parameterized complexity (Stefan Szeider),
- QBF solving (Nina Narodytska),
- random satisfiability (Dimitris Achlioptas).

Most tutorials and survey talks were scheduled early in the week, to create a conducive atmosphere for collaboration on open problems later in the week. The rest of the talks were 25-minute presentations on recent research of the participants. The time between lunch and afternoon coffee was left for self-organized collaborations and discussions, and there was no schedule on Wednesday afternoon.

Based on polling of participants before the seminar week, it was decided to have an open problem session on Monday evening, and on Wednesday evening there was a panel discussion. The organizing committee also considered the option of having a poster session to give more researchers the opportunity to present recent research results, but the feedback in the participant poll was negative and so this idea was dropped.

2 Table of Contents

Some Open Problems

3 Overview of Presentations

In this section we list the titles and abstracts of all presentations given during the seminar.

3.1 An introduction to proof complexity

Paul Beame (University of Washington – Seattle, US)

We give an overview of proof complexity including its basic definitions and many examples of natural and widely-studied propositional proof systems including inference systems using logical formulas, circuits, polynomials, and linear and polynomial inequalities. We also show how every complete SAT solver also yields a propositional proof system. We describe many of the known relationships between propositional proof systems and known bounds on their efficiency. We show some of the key techniques for bounding the lengths of propositional proofs, including relationships between their size, width, and degree and we show how this is related to forms of graph expansion of their input formulas. Finally, we describe a number of classes of natural examples of formulas that are hard to prove in various proof systems.

3.2 Tutorial on conflict-driven clause learning (CDCL) SAT solvers

João Marques-Silva (INESC-ID – Lisboa, PT)

Conflict-driven clause learning (CDCL) SAT solvers represent the de facto standard solver in practical problem solving with SAT, being used in the most visible and most successful practical applications of SAT. This tutorial will give an overview of the key concepts and techniques used in modern CDCL SAT solvers.

3.3 An Introduction to Semialgebraic Proofs: Basic Definitions and Results

Albert Atserias (UPC – Barcelona, ES)

Boolean satisfiability is a special case of integer linear programming, so we can hope to integrate some of their methods to SAT solvers. We will go over well-studied semialgebraic techniques, namely Gomory-Chvátal cuts and lift-and-project methods, and present some cases where they beat a resolution-based approach, as well as some lower bounds.

3.4 Handling Pseudo-Boolean constraints in a CDCL solver: a practical survey

Daniel Le Berre (CNRS – Lens, FR)

CDCL solvers have been quickly extended to handle arbitrary constraints. Doing so while preserving the original proof system of the solver does not require much changes to the solver.

Extending the proof system of the solver is however much more challenging. The talk will emphasize the extension of the CDCL architecture to the so called "generalized resolution" proof system, which lies between resolution and cutting planes proof systems, to handle Pseudo-Boolean constraints.

It will especially point out the strong requirements of the CDCL architecture on the proof system used for conflict analysis. Gory details about the constraints derived by such extended CDCL solver on benchmarks such as pigeon hole formulas will highlight both the strength and weaknesses of the resulting solver.

3.5 Gröbner bases

Manuel Kauers (Universität Linz, AT)

We explain what Gröbner bases are, why they are interesting, and how they are computed. The focus of the talk is on computational aspects. We will therefore not say much about how Gröbner bases can be used for solving all sorts of problems in commutative algebra. Instead, after discussing the classical Buchberger algorithm for computing Gröbner basis, we will try to sketch the underlying ideas of more recent algorithms.

3.6 Tutorial on proof systems connected to SAT solving

Sam Buss (University of California – San Diego, US)

Most SAT solvers implicitly generate refutation in the resolution proof system. We review this connection and characterize the shape of proofs generated by a CDCL solver. We introduce proof systems weaker than resolution that model these proofs.

3.7 Tutorial on preprocessing and inprocessing

Matti Järvisalo (University of Helsinki, FI)

This tutorial aims at covering (i) some of the most important preprocessing techniques used today in practice in conjunction with SAT solvers, and (ii) a generic "inprocessing" proof system capturing the deductions made by inprocessing SAT solvers that interleave CDCL search and preprocessing steps during search.

3.8 An Empirical Understanding of Conflict-Driven Clause-Learning SAT Solvers

Vijay Ganesh (University of Waterloo, CA)

Modern conflict-driven clause-learning (CDCL) Boolean SAT solvers routinely solve very large industrial SAT instances in relatively short periods of time. This phenomenon has stumped both theoreticians and practitioners since Boolean satisfiability is an NP-complete problem widely believed to be intractable. It is clear that these solvers somehow exploit the structure of real-world instances. However, to-date there have been few results that precisely characterize this structure, or shed any light on why these SAT solvers are so efficient.

In this talk, I will present results that provide a deeper empirical understanding of why CDCL SAT solvers are so efficient. First, we provide evidence that industrial SAT instances have "good community structure", and that this correlates more strongly with the running time of SAT solvers than traditional complexity-theoretic measures of SAT instance size such as number of clauses, variables or clause-variable ratio. Second, we characterize the famous VSIDS branching heuristic through a set of behavioral invariants that we discovered through a rigorous scientific process. These invariants include the following: First, VSIDS picks high-centrality bridge variables in the community structure of SAT instances much more often than other variables. Second, the multiplicative decay in VSIDS acts as a exponential moving average (EMA). Third, VSIDS is spatially and temporal focused (localized) with respect to the community structure of the SAT instance. We believe that the net effect of these behaviors of VSIDS is that it essentially enables the CDCL SAT solver to carry out a divide-and-conquer strategy by separating and then solving the communities of an instance.

Finally, I will present an abstract model of a SAT solver as an "active learner with deductive corrective feedback" that we believe is an accurate and analyzable mathematical model of CDCL solvers. I will also provide evidence that many successful techniques in formal verification and, more broadly, in software engineering can be abstractly modeled as "reinforcement learners with deductive corrective feedback".

3.9 MaxSAT Solving with SAT Oracles

João Marques-Silva (INESC-ID – Lisboa, PT)

Given an unsatisfiable formula, the maximum satisfiability problem (MaxSAT) is to identify a maximal subset of clauses that can be simultaneously satisfied. MaxSAT finds a growing number of practical applications, that include fault localization in software, design debugging in hardware, different applications in bioinformatics, timetabling and scheduling problems, among many others. For practical purposes, the most effective algorithms are based on iterative identification and relaxation of unsatisfiable subformulas using SAT solvers as oracles. This talk gives a brief overview of MaxSAT algorithms based on SAT oracles, and highlights what are currently the most effective techniques.

3.10 SAT-Enabled Verification of State Transition Systems

Karem A. Sakallah (University of Michigan – Ann Arbor, US)

The sequential behavior of complex artifacts, such as a hardware design or a software programs, is commonly captured by modeling the artifact as a formal state transition system. Given a desired (safety) property on the states of such a system, an important verification challenge is to determine whether all states reachable from a given (safe) initial state are safe and, if not, to produce an execution trace leading from the initial state to an unsafe state. Algorithmic approaches for solving this problem, in contrast to interactive theorem proving or proof checking methods, are what is referred to in the literature as model checking (MC).

In this talk I will briefly survey the evolution of MC over the last 30+ years highlighting the critical role SAT technology played in scaling MC to transition systems with exponentially-sized state spaces. I will also describe two specific applications, one in hardware and one in software, to illustrate the architecture of a scalable SAT-based verification environment.

3.11 Machine learning for SAT

Holger H. Hoos (University of British Columbia – Vancouver, CA)

In this presentation I will explain how machine learning methods can be used to automatically configure, select, combine and assess SAT solvers. I will briefly cover algorithm configuration techniques, such as SMAC (as used in the recent Configurable SAT Solver Challenges), automated algorithm selectors, such as SATzilla, automatic techniques for constructing parallel solver portfolios and finally, an interesting approach for assessing the scaling of solver performance with instance size that recently produced evidence that SLS-based SAT solvers like WalkSAT have running time polynomial in instance size for phase transition random-3-SAT instances.

3.12 How SAT Solvers Could (And Do) Prove Lower Bounds + (S)ETH and A survey of Consequences

Ryan Williams (Stanford University, US)

This is a merger of two tutorial talks: one by me on SAT algorithms and connections to computational complexity theory, and one by Mohan (cancelled) on the Exponential Time Hypothesis and the Strong Exponential Time Hypothesis.

3.13 A Survey on Parameterized Complexity and SAT

Stefan Szeider (TU Wien, AT)

In this talk I will discuss basic concepts of parameterized complexity (such as fixed-parameter tractability, reductions, hardness, and kernelization) and survey parameterized complexity results related to satisfiability (SAT). The focus will be on laying out what kind of questions can be asked and not on technical details.

3.14 From SAT to SMT – a Tutorial

Nikolaj S. Bjørner (Microsoft Corporation – Redmond, US)

Satisfiability Modulo Theories (SMT) solvers are used in many modern program verification, analysis and testing tools. They owe their scale and efficiency thanks to advances in search algorithms underlying modern SAT solvers and first-order theorem provers. They owe their versatility in software development applications thanks to specialized algorithms supporting theories, such as numbers and algebraic data-types, of relevance for software engineering.

This tutorial introduces algorithmic principles of SMT solving, taking as basis modern SAT solvers and integration with specialized theory solvers and quantifier reasoning. We detail some of the algorithms used for main theories used in current SMT solvers and survey newer theories and approaches to integrating solvers. The tutorial also outlines some application scenarios where SMT solvers have found use, including program verification, network analysis, symbolic model checking, test-case generation, and white-box fuzzing.

3.15 Survey on QBF solving

Nina Narodytska (Carnegie Mellon University, US)

Quantified Boolean formulas are a natural extension of propositional formulas with universal and existential quantifiers. QBF solvers are used in solving many problems in knowledge representation and reasoning, automated planning, and computer aided design.

In this talk, I will introduce the QBF problem and survey state-of-the-art techniques used in QBF solving. Then I will focus on a recent and successful approach that is based on the counterexample-guided abstraction refinement (CEGAR) paradigm. This approach proved very effective on a large number of industrial families of benchmarks.

3.16 QBF proof complexity

Olaf Beyersdorff (University of Leeds, GB)

In this talk we give an overview of the relatively young field of QBF proof complexity. We explain the main resolution-based proof systems for QBF, modelling CDCL and expansion-based solving. As our main contribution we exhibit a new and elegant proof technique for showing lower bounds in QBF proof systems based on strategy extraction. This technique provides a direct transfer of circuit lower bounds to lengths of proofs lower bounds. We use our method to show the hardness of a natural class of parity formulas for Q-resolution. Our lower bounds imply new exponential separations between two different types of resolution-based QBF calculi: proof systems for CDCL-based solvers and proof systems for expansion-based solvers. The relations between proof systems from the two different classes were not known before.

3.17 Parallel SAT Solving or To Share or Not To Share

Armin Biere (Universität Linz, AT)

We give a brief introduction into the problem and the current state-of-the-art of parallel SAT solving, mostly from a practical point of view. The talk continues with discussing current challenges.

3.18 Linear Temporal Logic Satisfiability Checking

Kristin Yvonne Rozier (University of Cincinnati, US)

Formal verification techniques are growing increasingly vital for the development of safety-critical software and hardware. Techniques such as requirements-based design and model checking have been successfully used to verify systems for air traffic control, airplane separation assurance, autopilots, logic designs, medical devices, and other functions that ensure human safety. Formal behavioral specifications written early in the system-design process and communicated across all design phases increase the efficiency, consistency, and quality of the system under development. We argue that to prevent introducing design or verification errors, it is crucial to test specifications for satisfiability.

In 2007, we established LTL satisfiability checking as a sanity check: each system requirement, its negation, and the set of all requirements should be checked for satisfiability before being utilized for other tasks, such as property-based system design or system verification via model checking. We demonstrated that LTL satisfiability checking reduces to model checking; an extensive experimental evaluation proved that for LTL satisfiability checking, the symbolic approach is superior to the explicit approach. However, the performance of the symbolic approach critically depends on the encoding of the formula. Since 1994, there had been essentially no new progress in encoding LTL formulas as symbolic automata for BDD-based analysis. We introduced a set of 30 symbolic automata encodings, demonstrating that a portfolio approach utilizing these encodings translates to significant, sometimes exponential, improvement over the standard encoding for symbolic LTL satisfiability checking. In recent years, LTL satisfiability checking has taken off, with others inventing exciting new methods to scale with increasingly complex systems. We revisit the benchmarks for LTL satisfiability checking that have become the de facto industry standard and examine the encoding methods that have led to leaps in performance. We highlight the past and present, and look to the future of LTL satisfiability checking, a sanity check that now has an established place in the development cycles of safety-critical systems.

3.19 Resolution Proofs of Bounded Width

Christoph Berkholz (KTH Royal Institute of Technology, SE)

The talk focuses on the structure and complexity of resolution refutations of bounded width (where every clause contains at most k literals).

Such refutations can be found in time $n^{O(k)}$ by exhaustively deriving all possible clauses with at most k literals. We show that this upper bound is tight by proving a matching lower bound. Furthermore, deciding whether there exists a resolution refutation of bounded width is EXPTIME-complete, whereas the same problem for regular resolution is PSPACE-complete.

We will also discuss the structure of bounded width refutations in terms of classical proof complexity measures such as resolution depth, (treelike) resolution size and clause space.

3.20 An Ultimate Trade-Off in Propositional Proof Complexity

Alexander Razborov (University of Chicago, US)

Trade-off results in complexity theory follow this general pattern: a task is exhibited that is easy with respect to a chosen complexity meeasure but becomes much harder after requiring that the protocol is efficient with respect to another, normally very different, measure. In most cases, "much harder" means "as hard as an average task of comparable size" without imposing any restrictions on the protocol.

In this talk we exhibit an unusually strong trade-off result between width and tree-like resolution proof size that significantly deviates from this pattern. Namely, we construct unsatisfiable k-CNFs that possess refutations of very small width $O(k)$ but such that any tree-like resolutation refutation of even mildly sublinear width $n^{1-\epsilon}/k$ must be of double exponential size $\exp(n^{\Omega(k)})$. This is exponentially larger than the trivial 2^n size bound to which all unsatisfiable CNFs with n variables are entitled.

3.21 Narrow Proofs May Be Maximally Long

Massimo Lauria (KTH Royal Institute of Technology, SE)

We prove that there are 3-CNF formulas over n variables refutable in resolution in width w that require resolution proofs of size $n^{\Omega(w)}$. This shows that the simple counting argument that any formula refutable in width w must have a proof in size $n^{O(w)}$ is essentially tight. Moreover, our lower bound extends even to polynomial calculus resolution (PCR), Sherali-Adams and Lasserre/Sums-of-Squares, implying that the corresponding size upper bounds in terms of degree are tight as well.

3.22 A Survey of Random Satisfiability

Dimitris Achlioptas (University of California – Santa Cruz, US)

Given a CNF formula F, let $S(F)$ denote its set of satisfying assignments. We consider a random k-CNF formula F on n variables, constructed by adding m random clauses one by one, each clause selected uniformly at random among all $2^k\binom{n}{k}$ possible clauses. The talk will give a survey of results about random satisfiability by narrating the "video" of $S(F)$ as clauses are added. We will see that two important phase transitions occur (neither of which is the satisfiability transition) and emphasis will be placed on their potential algorithmic implications. No familiarity with random satisfiability will be assumed.

3.23 Space and Random CNFs

Ilario Bonacina (University of Rome "La Sapienza", IT)

We will see some space lower bounds in Resolution and Polynomial Calculus Resolution (PCR) for random k-CNFs. More precisely about random 3-CNFs: a quadratic lower bound for the total space needed in Resolution to refute such formulas and a linear lower bound for monomial space in PCR.

3.24 Improving and Evaluating a Hybrid Approach to Max-SAT Solving

Jessica Davies (IST Austria – Klosterneuburg, AT)

MaxHS is a recent approach to solving Max-SAT that utilizes a hybrid algorithm that exploits both a SAT solver and an IP solver as black-boxes. This approach has a number of attractive properties, but in the recent Max-SAT Evaluations it has not performed as well as other purely SAT-based solvers. In this paper we examine a current implementation of MaxHS and find a number of improvements. With these improvements implemented we compare the performance of the approach to other approaches for solving Max-SAT. Our results indicate that the hybrid approach remains a promising direction for further research.

3.25 Bit-Vectors: Complexity and Decision Procedures

Andreas Fröhlich (Universität Linz, AT)

Bit-vectors are important for many practical applications in verification. We discuss theory and practice by giving complexity results and presenting several alternative decision procedures.

4 Some Open Problems

Before the seminar, the organizers collected a list of open problems from the participants that could potentially be discussed during the open problem session Monday evening and at other times during the week. All submitted problems were collected at the webpage http://www.csc.kth.se/~jakobn/dagstuhl15171/openproblems.php. Many of these problems were indeed discussed during the Monday evening problem session, and in addition other problems were raised there as well.

Below follows a hopefully representative selection of these open problems. The list is basically unsorted except it is (roughly) in chronological order of submission. Some partially overlapping problems have been merged. The full list of problems is still available at http://www.csc.kth.se/~jakobn/dagstuhl15171/openproblems.php. One suggestion put forward during the seminar week was to collect these and other research problems on a wiki-style website to stimulate research. This seems like a very attractive idea, and is something that might be implemented in the future.

4.1 Minimum variable space and minimum depth of resolution refutations

Alexander Razborov (University of Chicago, US)

Can it be the case that minimum variable space is equivalent, up to a polynomial and $\log n$ factors, to the minimum depth of resolution refutations? This is true if we additionally normalize variable space by log of the proof length, therefore an equivalent form of our question is this: does there exist a strong ultimate tradeoff between variable space and proof length?

4.2 Exact counting for k-SAT

Ryan Williams (Stanford University, US); (originally from Rahul Santhanam)

It is known that k-SAT with n variables and m clauses can be solved in about $2^{n-n/O(k)} \cdot poly(m)$ time, and these are the best known running times for tthe worst case. For computing the number of k-SAT solutions, there is a randomized algorithm of Impagliazzo, Matthews, and Paturi (SODA'12) running in $2^{n-n/O(k)} \cdot poly(m)$ time, and a deterministic algorithm of Beame, Impagliazzo, and Srinivasan (CCC'12) running in worse time.

Is there a deterministic worst-case $\#k$-SAT algorithm running in $2^{n-n/O(k)} \cdot poly(m)$ time? (Give an algorithm, or evidence against its existence.)

4.3 Optimality of Regular Resolution?

Alasdair Urquhart (University of Toronto, CA)

- Show that for well known examples such as the pigeonhole principle (PHP) and Tseitin formulas, regular resolution is optimal. This conjecture seems very plausible to me, but I don't see how to approach it at the moment.
- More generally, you can ask: Can you give general conditions on a set of clauses that ensure that regular resolution is optimal? In general, the examples separating general and unrestricted resolution have a rather artificial appearance, where we add "spoiler variables" to mess up any regular refutation.
- A closely related problem that may be more accessible is this: for the same set of examples, show that the regular width and the unrestricted width of a refutation are the same. Are there general conditions that ensure this equality?

4.4 How and why does VSIDS work? (Full simulation of resolution by CDCL with heuristics?)

Alexandra Goultiaeva (Google Waterloo, CA), Armin Biere (Universität Linz, AT), and Vijay Ganesh (University of Waterloo, CA)

The variable scoring scheme VSIDS (variable state independent decaying sum) introduced by Chaff and its modern variants is crucial for the speed of CDCL solvers. There is almost no empirical investigation on how it really works, and further no theoretical explanation why it is working.

In particular, it has been proven that CDCL SAT solvers p-simulate resolution. The order of decisions is assumed to be arbitrary, i.e., the proof shows that (if a short resolution proof exists) there exists a sequence of decisions that would allow the solver to find a short resolution proof. I.e, a result that either:

- shows that whenever a short resolution proof exists, there will always be a sequence of decisions that respects VSIDS ordering and allows the solver to find a short resolution proof, or
- shows a counterexample where a short resolution proof exists but a solver respecting VSIDS ordering (regardless of tie-breaking) can never find a short proof.

4.5 Learning definitions through extended resolution

Armin Biere (Universität Linz, AT)

There has been attempts to shrink learned clauses by introducing definitions in the sense of extended resolutions, which however in practice has not really been effective. It is unclear whether these newly introduce literals are can really be used in the search process and shrink proofs. The question is whether it is possible to come up with a more general but practical scheme to introduce definitions, which allow to shrink proof size and improve SAT solving in practice too.

4.6 Limits of portfolio based parallel SAT solving

Armin Biere (Universität Linz, AT)

Portfolio based SAT solving is the dominating approach in the parallel application track of the SAT competition. However, the improvements we saw in the last two years are apparently based on using better sharing schemes for learned clauses, thus kind of implicit work splitting. From a practical point of view it is first of all still unclear how much of the success of solvers like Penelope or Plingeling can be contributed to the portfolio idea and how much is due to splitting the work. As the number of compute units is increased it is conjectured that the relative contribution of the portfolio part will saturate. Does this happen and when?

4.7 What is the relationship, if any, between cluster analysis and survey propagation on application SAT instances?

Allen Van Gelder (University of California – Santa Cruz, US)

There are several recent works on cluster analysis (AKA community structure) of application SAT instances. They seem to focus on connections between clause learning, VSIDS, and the page-rank algorithm.

- What new idea is needed for survey propagation to be useful on application SAT instances?
- Is survey propagation useful somehow on unsatisfiable application SAT instances? Can certain behavior suggest the application SAT instance is unsat and give evidence?
- Can cluster analysis on application SAT instances give a hint or prediction whether the application SAT instance is unsat or sat?

4.8 Why does conflict-driven search work so well? (How do CDCL solvers exploit the structure of real-world instances?)

Karem Sakallah (University of Michigan – Ann Arbor, US), Sharad Malik (Princeton University, US), and Vijay Ganesh (University of Waterloo, CA)

Empirical evaluation of solver performance suggests that the two most important features of modern SAT solvers are *conflict-driven clause learning* and *conflict-driven branching*. Tracing the execution of a modern conflict-driven solver seems to show that the solver is aggressively trying to falsify the formula (looking for conflicts) and only when that fails does it yield a satisfying assignment. This strategy seems to work quite well on a very diverse set of benchmarks. Why? Can we characterize when it does not work? What problem structure causes an aggressive falsification approach to fail? What other strategies can we envision to complement conflict-driven search?

4.9 How to cut directed paths in a dag (related to the complexity of CircuitSAT)

Edward A. Hirsch (Steklov Institute – St. Petersburg, RU)

Consider directed acyclic graphs with vertices of indegree at most two (that is, Boolean circuits). Prove (or disprove) that for every $\epsilon > 0$ there is a constant $K = K(\epsilon)$ such that for every n large enough in every such dag with n vertices there is a subset of vertices of size at most $\epsilon \cdot n$ such that its removal (with incident edges) leaves no directed paths of length more than K.

4.10 How Total Space and Monomial Space relate with other complexity measures?

Ilario Bonacina (University of Rome "La Sapienza", IT)

Given an unsatisfiable CNF ϕ let's see a refutation of it in Resolution (res. PCR) as a sequence of memory configurations, i.e. set of clauses (res. polynomials) such that each memory configuration is obtained from the previous one either (i) removing some clause (resp. polynomial), or (ii) adding some clause from ϕ, or (iii) inferring some consequence applying the inference rules to something in memory.

$\mathit{MSpace}_{PCR}(\phi \vdash \bot) \geq m$ means that for every PCR refutation π of ϕ (according to the previous model) there must be some memory configuration in π in which at least m distinct monomials appear (maybe in several places in the polynomials in that memory configuration).

TSpace$(\phi \vdash \perp) \geq m$ means that for every (Res/PCR) refutation π of ϕ (according to the previous model) there must be some memory configuration in π in which the total number of occurrences of literals in that memory configuration is at least m.

So the questions are the following:

- Is it the case that given a k-CNF ϕ, $\mathit{TSpace}_{RES}(\phi \vdash \perp) = \Omega((\mathit{width}(\phi \vdash \perp) - k)^2)$?
- Is it the case that given a k-CNF ϕ, $\mathit{MSpace}_{PCR}(\phi \vdash \perp) = \Omega(\mathit{degree}(\phi \vdash \perp) - k)$?
- Is there any k-CNF ϕ in n variables and $n^{O(1)}$ clauses such that $\mathit{TSpace}_{PCR}(\phi \vdash \perp) = \Omega(n^2)$? It should be true w.h.p. for random k-CNFs for any $k \geq 3$ and with clause density a constant above the unsatisfiability threshold.

4.11 Random k-SAT

Dimitris Achlioptas (University of California – Santa Cruz, US)

- The sat/unsat threshold for random 10-SAT is provably > 700. Solve random 10-SAT instances with 100,000 variables of density 600 (or greater). (hard)
- The sat/unsat threshold for random 6-SAT is provably > 40. Solve random 6-SAT instances with 100,000 variables of density 35 (or greater). (not easy)
- The mixture of $(1-\epsilon)n$ random 2-clauses and $(2/3)n$ random 3-clauses (on the same variables) is satisfiable with high probability, for every $\epsilon > 0$. Prove that $2/3$ is best possible. That is, prove that for every $\delta > 0$, there exists $\epsilon > 0$ such that such a mixture is unsatisfiable. (hard)

4.12 The complexity of the parity principle in semi-algebraic systems

Paul Beame (University of Washington – Seattle, US)

Determine the complexity of the parity principle (also known as the mod 2 counting principle, or the matching principle on K_{2n+1}) in semi-algebraic systems, especially LS and LS+:

This has a variable for each edge of the complete graph on an odd number of vertices. In clausal form this has clauses like the bijective pigeonhole for each vertex but it is easy to derive $\sum_{i\neq j} x_{ij} = 1$ in small size in these systems. (In LS it takes degree $\Omega(n)$ to derive this but it is only quadratic size. In LS+ there is a rank one derivation.)

In cutting planes it is easy to derive a contradiction from this since one can add all of the equations to get $2\sum_{i,j:i\neq j} x_{ij} = 2n+1$ and rounding in both directions yields a contradiction. However, it is not clear how to simulate this "division by 2" in any semi-algebraic system. This is related to the Knapsack problem considered by Grigoriev. He showed that if a sum of m variables is an odd number that is near the middle of the interval $[0, m]$ then Positivstellensatz degree is large. Using the methods of Kojevnikov and Itsyksen this yields tree-like size lower bounds for LS. The differences here are that there are $\binom{2n+1}{2}$ variables and the $2n+1$ bound is nowhere near the middle of the range $[0, \binom{m}{2}]$, and we have separate equations for subset of the variables.

5 Panel Discussion

On Wednesday evening there was a panel discussion with Paul Beame, Nikolaj Bjørner, Sam Buss, Sharad Malik, Karem Sakallah, and Stefan Szeider serving as members of the panel. The panel members opened the discussion with a short "keynote remark" each (of around 4–5 minutes), after which followed a discussion of a little bit more than an hour between panel members and all participants present. The purpose of the panel was to discuss question such as promising and/or important future research directions, how (and if) we should get more interaction between practitioners and theoreticians doing SAT-related research, or whatever else the seminar participants wanted to talk about.

The panel discussion brought out several socio-scientific issues at the forefront of satisfiability research. Three of the most memorable issues were:

1. Some researchers lamented over the laser-like focus of many on the SAT competitions; they felt that not enough attention is being paid to the long-term scientific goal of understanding of why SAT is solvable in practice. Others argued in response that the SAT competitions are fun and community-building; they help motivate people to do worthy work with good intentions.
2. Related to the subject of competitions, a few researchers objected to their format, again based on scientific disagreement. There is still a rift between those designing "classical" CDCL-based SAT solvers, and those who use machine learning techniques to design algorithm "portfolios" selecting such SAT solvers to run on instances, and the SAT competition has developed rules to isolate the latter group from the rest of the solver submissions. The question of whether re-designing the competition in this way will positively (or negatively) influence further research is intriguing; it certainly was not resolved by this panel discussion.
3. Related to the subject of understanding SAT, there was extensive speculation by many parties on why SAT solvers tend to work so well in practice. Some pointed to the variable choice heuristics of solvers; some pointed to the clause learning of solvers; some posited that there must be inherent structure in most real-world SAT instances. Some asked (controversially) *if and why we should expect be able to understand SAT solvers at all:* SAT code and SAT instances solved in practice are so complex that perhaps humans simply cannot know, or cannot rigorously explain why practical SAT instances are solved so efficiently.

All in all, the thought-provoking discussion highlighted the diversity of attitudes and ideas that people bring to SAT research.

6 Examples of Outcomes of the Workshop

It is still a bit too early for any concrete publications to have resulted from the seminar, but participants have reported that the following papers, in different stages of preparation, were significantly influenced by discussions during the seminar:

- Albert Atserias, Massimo Lauria, and Jakob Nordström. **Narrow Proofs May Be Maximally Long.** Journal version in submissions, 2015.
- Armin Biere and Andreas Fröhlich. **Evaluating CDCL Variable Scoring Schemes.** To appear in *Proceedings of SAT'15,* September 2015.

- Armin Biere and Andreas Fröhlich. **SAT Solving and Stock Market Analysis.** Manuscript in preparation, 2015.
- Oliver Kullmann and João Marques-Silva. **Computing maximal autarkies with few and simple oracle queries.** To appear in *Proceedings of SAT'15,* September 2015.
- Massimo Lauria and Jakob Nordström. **Tight Size-Degree Bounds for Sums-of-Squares Proofs.** In *Proceedings of CCC'15,* June 2015.
- Jia Hui Liang, Vijay Ganesh, Ed Zulkoski, Atulan Zaman, Krzysztof Czarnecki. **Understanding VSIDS Branching Heuristics in Conflict-Driven Clause-Learning SAT Solvers.** Manuscript in submission, 2015.
- Jakob Nordström. **On the Interplay Between Proof Complexity and SAT Solving.** *ACM SIGLOG News,* July 2015.
- Mladen Mikša and Jakob Nordström. **A Generalized Method for Proving Polynomial Calculus Degree Lower Bounds.** In *Proceedings of CCC'15,* June 2015.

Making the connection to the panel discussion which we report on in Section 5, the Dagstuhl seminar week played an important role in stimulating a research project focused on a comprehensive empirical study to better understand the impact on performance of different features in modern CDCL SAT solvers. In joint work, Laurent Simon, João Marques-Silva, and Karem Sakallah have collected all non-random benchmarks from all SAT competitions and races (2002 to 2014) and instrumented both Minisat and Glucose to enable and disable their various options in order to pinpoint the effect of each option or combination of options on performance. The plan is to make this data available on a public website and provide extensive analysis of the data in a paper that is currently under preparation.

Other participants of the seminar have reported about at least six concrete research projects that resulted to a large part from contacts during the week at Dagstuhl. Since many of these projects are still in a start-up phase it would seem slightly premature to list concrete participants, but it can be mentioned that these projects involve researchers from INESC-ID Lisboa, Johannes Kepler University, KTH Royal Institute of Technology, Microsoft Research, Princeton University, RWTH Aachen, Swansea University, Universitat Politécnica de Catalunya, and University of Washington in various constellations. Several of these projects involves interdisciplinary research with both applied and theoretical components, and many seminar participants mentioned explicitly that the mix of theoreticians and practitioners at the seminar played a decisive role in making this happen.

7 Evaluation by Participants

In addition to the traditional Dagstuhl evaluation after the seminar, the organizing committee also arranged for a separate evaluation which specific questions about different aspects of the seminar. Below follows a summary of the answers – the full results are available at http://www.csc.kth.se/~jakobn/dagstuhl15171/evaluation.php.

In the post-seminar survey, the participants identified two major aspects of the seminar they enjoyed most: the networking opportunities between theoreticians and practitioners that the environment of Dagstuhl provided, and the high quality of the tutorial talks selected by the organizers. Many reported that they learned a substantial amount from the seminar talks.

However, the seminar was not immune from some negative feedback. Some found the tutorials too elementary, and felt there was not enough focus on talks with new results. Some felt that there should have been talks on the applications and general impact of SAT in science and engineering. Some participants felt there was not enough proof complexity and others felt there was too much. A few did not like that some of the schedule extended into the late evening (which was the case for the Monday evening open problem session and the Wednesday evening panel discussion).

The seminar participants were polled before the seminar about some different aspects of the planning, and based on the results of this poll it was decided to have an open problem session on the first day. It the post-seminar survey, this decision was viewed favourably: 48% felt of respondents it was “definitely the right decision” and 40% felt it was “probably” the right decision. Some felt that the open problem session had too many problems, many of which were either too vague to fully grasp or too specific to be interesting; perhaps a “curated” open problem session would have been more effective.

Also based on results of the pre-seminar poll, we decided not to have poster session, and an overwhelming majority felt this was the right decision in hindsight as well. Nevertheless, some did wish that there had been more opportunities to recreate “what happens at a poster session”: structured informal discussions about SAT research among many participants.

In general, much of the feedback contained the sentiment that more time for “guided” extended discussions among the entire group would have been useful. This is interesting when placed in the context of the feedback on the panel discussion (which was an intentionally guided discussion of SAT issues). Only slightly more than half of the respondents to the post-seminar survey felt that the panel was either “definitely” or “probably” a good idea with hindsight. Some enjoyed the panel, but others did not find the discussion fruitful. One participant, noting the abundance of experts at the seminar, suggested that a “town hall style” meeting (where everyone had the same chance to state their views) might have fared better.

All in all, the feedback from the participants was overwhelmingly positive. Many called the experience “great” or “fantastic” and thought the seminar had been “superbly organized” with “outstanding” talks. One participant even wrote that “[t]his was hands down the best Dagstuhl I have ever attended, and I have attended 10 so far”, and another respondent noted that “I and other people remarked that it seemed we could easily continue into a second week – people were refreshed rather than exhausted by the end of the seminar.” Many participants look forward to returning to Dagstuhl: in the post-seminar evaluation, 72% said they would definitely come again if invited to a similar seminar, and 20% said they would probably come again.

Participants

- Erika Abraham
RWTH Aachen, DE
- Albert Atserias
UPC – Barcelona, ES
- Gilles Audemard
CNRS – Lens, FR
- Paul Beame
University of Washington – Seattle, US
- Christoph Berkholz
KTH Royal Institute of Technology, SE
- Olaf Beyersdorff
University of Leeds, GB
- Armin Biere
Universität Linz, AT
- Nikolaj S. Bjørner
Microsoft Corporation – Redmond, US
- Ilario Bonacina
University of Rome "La Sapienza", IT
- Sam Buss
University of California – San Diego, US
- Amin Coja-Oghlan
Goethe-Universität – Frankfurt a. M., DE
- Jessica Davies
IST Austria – Klosterneuburg, AT
- Holger Dell
Universität des Saarlandes, DE
- Jan Elffers
KTH Royal Institute of Technology, SE
- John Franco
University of Cincinnati, US
- Andreas Fröhlich
Universität Linz, AT
- Vijay Ganesh
University of Waterloo, CA
- Alexandra Goultiaeva
Google Waterloo, CA
- Martin Grohe
RWTH Aachen, DE
- Daniel Große
Universität Bremen, DE
- Edward A. Hirsch
Steklov Institute – St. Petersburg, RU
- Holger H. Hoos
University of British Columbia – Vancouver, CA
- Matti Järvisalo
University of Helsinki, FI
- Mikoláš Janota
INESC-ID – Lisboa, PT
- Manuel Kauers
Universität Linz, AT
- Oliver Kullmann
Swansea University, GB
- Massimo Lauria
KTH Royal Institute of Technology, SE
- Daniel Le Berre
CNRS – Lens, FR
- Sharad Malik
Princeton University, US
- João Marques-Silva
INESC-ID – Lisboa, PT
- Nina Narodytska
Carnegie Mellon University, US
- Jakob Nordström
KTH Royal Institute of Technology, SE
- Yakau Novikau
OneSpin Solutions – München, DE
- Albert Oliveras
UPC – Barcelona, ES
- Pavel Pudlák
Academy of Sciences – Prague, CZ
- Alexander Razborov
University of Chicago, US
- Kristin Yvonne Rozier
University of Cincinnati, US
- Karem A. Sakallah
University of Michigan – Ann Arbor, US
- Martina Seidl
Universität Linz, AT
- Laurent Simon
University of Bordeaux, FR
- Niklas Sörensson
Mentor Graphics – Göteborg, SE
- Stefan Szeider
TU Wien, AT
- Alasdair Urquhart
University of Toronto, CA
- Allen Van Gelder
University of California – Santa Cruz, US
- Marc Vinyals
KTH Royal Institute of Technology, SE
- Magnus Wahlström
Royal Holloway University of London, GB
- Ryan Williams
Stanford University, US