Schedule for Dagstuhl workshop 22411 "Theory and Practice of SAT and Combinatorial Solving"

Time (ca)	MONDAY OCT 10	TUESDAY OCT 11	WEDNESDAY OCT 12	THURSDAY OCT 13	FRIDAY OCT 14	Time (ca)
08:45	**8:45-9:00:** Welcome					08:45
09:00	**9:00-9:55**	**9:00-9:55**	**9:00-9:55**	**9:00-9:55**	**9:00-9:30**	09:00
09:15	Beyersdorff	Mahajan	Kovács and Suda	Fleury	Meel: Designing samplers	09:15
09:30	Theory & practice of SAT	QBF solving and complexity	First-order theorem proving	Verified solvers	**9:30-10:00**	09:30
09:45	**9:55-10:50**	**9:55-10:50**	**9:55-10:50**	**9:55-10:50**	Böhm: QBF CDCL vs resolution	09:45
10:00	Biere	McCreesh	Gleixner	Tan	**10:00-10:30**	10:00
10:15	Trusting SAT solvers	Constraint programming	Mixed integer programming	Verified proof checkers	Vinyals: Limits of 1UIP learning	10:15
10:30					***COFFEE BREAK***	10:30
10:45	***COFFEE BREAK***	***COFFEE BREAK***	***COFFEE BREAK***	***COFFEE BREAK***		10:45
11:00	**11:10-12:05**	**11:10-12:05**	**11:10-12:05**	**11:00-11:30**	**11:00-11:30**	11:00
11:15	Williams	Bjørner	Helmert	Oertel: Certified CNF translation	Fleury: Evaluating solver options	11:15
11:30	Complexity and SAT	Satisfiability modulo theories	Automated planning	**11:30-12:00**	**11:30-12:00**	11:30
11:45				Bogaerts: Certified symmetries	Ganesh: SAT + computer algebra	11:45
12:00						12:00
12:15	***LUNCH***	***LUNCH***	***LUNCH***	***LUNCH***	***LUNCH***	12:15
12:30						12:30
12:45					***WORKSHOP ENDS***	12:45
13:00						13:00
13:15						13:15
13:30					***LEGEND***	13:30
13:45					*Long talk (55 min)*	13:45
14:00	**14:00-14:55**		***WEDNESDAY***		*Short talk (30 min)*	14:00
14:15	Nordström				*Other*	14:15
14:30	Proof complexity and		***AFTERNOON***			14:30
14:45	SAT solving					14:45
15:00	***COFFEE & CAKE***	**15:00-15:30**	***FREE***	**15:00-15:30**		15:00
15:15		Schidler et al.: SLIM part I		Yolcu: Exponential separations		15:15
15:30	**15:40-16:35**	**15:30-16:00**		**15:30-16:00**		15:30
15:45	de Rezende	Schidler et al.: SLIM part II		Berg: Reundancy in MaxSAT		15:45
16:00	Efficient proof search	***COFFEE & CAKE***		***COFFEE & CAKE***		16:00
16:15						16:15
16:30	**16:35-17:30**	**16:30-17:00**		**16:30-17:00**		16:30
16:45	Kaufmann	Järvisalo: PB optimization		Heule: Packing chromatic number		16:45
17:00	Algebraic methods for	**17:00-17:30**		**17:00-18:00**		17:00
17:15	circuit verification	Fazekas: Extending CDCL		Open problem session		17:15
17:30	**17:30-18:00**	**17:30-18:00**				17:30
17:45	Presentation of participants	Li: Understanding CDCL				17:45
18:00	**DINNER**	**DINNER**	**DINNER**	**DINNER**		18:00
18:15						18:15

Dagstuhl Workshop 22411:
Theory and Practice of SAT and Combinatorial Solving
October 9–14, 2022

GENERAL INFORMATION

- Please be aware that you will need the username and password of your door account (i.e., the credentials you used to register for the seminar) while in Dagstuhl.
- The Dagstuhl reception is located in the facility opposite the manor house. The reception is open from 15:00 to 19:00 on Sunday and from 8:00 to 16:00 on other days. If it is closed when you arrive, please use the access code in your voucher to enter the building and then follow the self-service check-in procedure described at the reception.
- Departure is on Friday October 14. Dagstuhl kindly asks you to clear your room by 9:00 and to pay your bill for accommodation, meals and your private expenses on the day of your departure before lunch.

MEALS

- Breakfast is served from 7:30 to 8:45.
- Lunch is served at 12:15.
- Dinner is served at 18:00.
- During the morning break, coffee and tea are available outside the seminar room.
- In the afternoon, coffee and cake are served between 15:00 and 16:00 in the dinner hall.

SCHEDULE

Sunday October 9

15:00	Dagstuhl reception opens
18:00–	Buffet dinner
20:00–	Informal gathering in the lounge in the castle (if desired) Beverages and a small assortment of snacks are available on a cash honour system.

Monday October 10

07:30–08:45	Breakfast
08:45–09:00	Welcome
09:00–09:55	Olaf Beyersdorff *Theory and Practice of SAT Solving*
09:55–10:50	Armin Biere *Trusting SAT Solvers*
10:50–11:10	Coffee break
11:10–12:05	Ryan Williams *Around the Complexity of SAT*
12:15	Lunch
14:00–14:55	Jakob Nordström *Proof complexity and SAT solving*
15:00–15:40	Coffee and cake
15:40–16:35	Susanna F. de Rezende *Theoretical Barriers for Efficient Proof Search (a Survey)*
16:35–17:30	Daniela Kaufmann *Exploring Algebraic Methods for Circuit Verification*
17:30–18:00	Presentation of participants
18:00	Dinner

Tuesday October 11

07:30–08:45	Breakfast
09:00–09:55	Meena Mahajan *Quantified Boolean Formulas: Solving and Proof Complexity*
09:55–10:50	Ciaran McCreesh *How Constraint Programming Isn't Like SAT*
10:50–11:10	Coffee break
11:10–12:05	Nikolaj Bjørner *An Introduction to SMT with Proofs*
12:15	Lunch
	Afternoon break
15:00–16:00	Andre Schidler, Friedrich Slivovski, and Stefan Szeider *Scalable optimization with SAT-based local improvement (SLIM)*
16:00–16:30	Coffee and cake
16:30–17:00	Matti Järvisalo *Pseudo-Boolean Optimization by Implicit Hitting Sets*
17:00–17:30	Katalin Fazekas *On Design Decisions of Extending CDCL with External Propagators*
17:30–18:00	Chunxiao (Ian) Li *Towards a Deeper Understanding of Modern CDCL SAT Solvers*
18:00	Dinner

Wednesday October 12

07:30–08:45	Breakfast
09:00–09:55	Laura Kovács and Martin Suda *First-Order Theorem Proving*
09:55–10:50	Ambros Gleixner *Algorithmic Mixed Integer Programming: Between Exactness and Performance in Theory and Practice*
10:50–11:10	Coffee break
11:10–12:05	Malte Helmert *Introduction to Automated Planning*
12:15	Lunch
	Free afternoon
18:00	Dinner

Thursday October 13

07:30–08:45	Breakfast
09:00–09:55	Mathias Fleury *Verifying Solvers: How Much Do You Want to Prove?*
09:55–10:50	Yong Kiam Tan *The Last Mile in Trustworthy Automated Reasoning*
10:50–11:00	Coffee break
11:00–11:30	Andy Oertel *Certified CNF Translations for Pseudo-Boolean Solving*
11:30–12:00	Bart Bogaerts *On Proof Logging and Symmetry Handling*
12:15	Lunch
	Afternoon break
15:00–15:30	Emre Yolcu *Exponential separations using guarded extension variables*
15:30–16:00	Jeremias Berg *Clause Redundancy and Preprocessing in Maximum Satisfiability*
16:00–16:30	Coffee and cake
16:30–17:00	Marijn Heule *The Packing Chromatic Number of the Infinite Square Grid is 15*
17:00–18:00	Open problem session
18:00	Dinner

Friday October 14

7:30–8:45	Breakfast
9:00–9:30	Kuldeep Meel *Designing Samplers is Easy: The Boon of Testers*
9:30–10:00	Benjamin Böhm *CDCL vs resolution: the picture in QBF*
10:00–10:30	Marc Vinyals *Theoretical limits of 1UIP Learning*
10:30–11:00	Coffee break
11:00–11:30	Mathias Fleury and Armin Biere *Discussion: How to combine and compare options in solvers?*
11:30–12:00	Vijay Ganesh *A SAT Solver + Computer Algebra Attack on the Minimum Kochen-Specker Problem*
12:15	Lunch

Jeremias Berg: Clause Redundancy and Preprocessing in Maximum Satisfiability

The study of clause redundancy in Boolean satisfiability (SAT) has proven significant in various terms, from fundamental insights into preprocessing and inprocessing to the development of practical proof checkers and new types of strong proof systems. I will present our recent work on liftings of the recently-proposed notion of propagation redundancy — based on a semantic implication relationship between formulas — in the context of maximum satisfiability (MaxSAT), where of interest are reasoning techniques that preserve optimal cost (in contrast to preserving satisfiability in the realm of SAT). We establish the strongest MaxSAT-lifting of propagation redundancy allows for changing in a controlled way the set of minimal correction sets in MaxSAT. This ability is key in succinctly expressing MaxSAT reasoning techniques and allows for obtaining correctness proofs in a uniform way for MaxSAT reasoning techniques very generally. I will also highlight some interesting directions for future work.

Olaf Beyersdorff: Theory and Practice of SAT Solving

This talk will provide a survey on the relations between proof complexity and SAT solving. What can proof complexity tell us about the strength and limitations of SAT solving? Why should practitioners be interested in proof complexity results and why should theorists study SAT solving? What have we achieved in the past 25 years and which problems remain open?

Armin Biere: Trusting SAT Solvers

Many critical applications crucially rely on the correctness of SAT solvers. Particularly in the context of formal verification, the claim by a SAT solver that a formula is unsatisfiable corresponds to a safety or security property to hold, and thus needs to be trusted. In order to increase the level of trust an exciting development in this century was to let SAT solvers produce certificates, i.e., by tracing proofs of unsatisfiability, which can independently be checked. In the last ten years this direction of research gained substantial momentum, e.g., solvers in the main track of the SAT competition are required to produce such certificates and industrial applications of SAT solvers require that feature too. In this talk we review this quarter of century of research in certifying the result of SAT solvers, discuss briefly alternatives, including testing approaches and verifying the SAT solver directly, mention exciting research on new proof systems produced in this context as well as how these ideas extend beyond formulas in conjunctive normal form.

Nikolaj Bjørner: An Introduction to SMT with Proofs

The talk provides an overview of selected current trends in SMT solving theories and techniques. An active area of discussion in the SMT community is around proof formats for SMT solvers. I give an introduction to current approaches pursued in solvers such as Z3, CVC5, VeriT and SMTInterpol.

Benjamin Böhm: CDCL vs resolution: the picture in QBF

This talk will cover the relations between QBF resolution and QCDCL solving algorithms. Modelling QCDCL as proof systems we show that QCDCL and Q-Resolution are incomparable. We also introduce new versions of QCDCL that turn out to be stronger than the classic models.

This talk is based on a couple of recent papers (joint with Olaf Beyersdorff and Tomas Peitl, which appeared in ITCS'21, SAT'21, SAT'22 and IJCAI'22).

Bart Bogaerts: On Proof Logging and Symmetry Handling

In this talk, we take a deep-dive in the fascinating world of symmetry handling for Boolean Satisfiability, by reviewing static and dynamic techniques. We focus on proof logging techniques for these symmetry handling methods. We end with some open problems and challenges.

Susanna F. de Rezende: Theoretical Barriers for Efficient Proof Search (a Survey)

The proof search problem is a central question in automated theorem proving and SAT solving. Clearly, if a propositional tautology F does not have a short (polynomial size) proof in a proof system P, any algorithm that searches for P-proofs of F will necessarily take super-polynomial time. But can proofs of "easy" formulas, i.e., those that have polynomial size proofs, be found in polynomial time? This question motivates the study of automatability of proof systems. In this talk, we give an overview of known non-automatability results, focusing on the more recent ones, and present some of the main ideas used to obtain them.

Katalin Fazekas: On Design Decisions of Extending CDCL with External Propagators

Solving combinatorial problems often combines SAT solving with different reasoning techniques. An external propagator can interpret the partial assignment built by the SAT solver during search and, based on such different reasoning methods, can construct clauses that are propagating or conflicting under that assignment. The use of such external propagators allows to directly guide the search of the SAT solver into a preferred direction, which can make problem solving in several problem domains more efficient (consider e.g. dynamic symmetry breaking). However, both the efficient combination of external propagators with the complex features of modern SAT solvers (e.g. proof logging and inprocessing), and the theoretical understanding of such combined reasoning methods are open problems.

This talk, based on current work in progress, presents some of the design decisions that must be considered when external propagation is combined with modern CDCL solvers. We describe some challenges and formulate both practical and theoretical open questions about how to implement external propagation in CDCL in the presence of current SAT solver features such as proof logging, clause database reduction and inprocessing.

Mathias Fleury: Verifying Solvers: How Much Do You Want to Prove?

In this talk, I present the two main approaches to verify solvers: partial verification (usually bottom-up from code to the specification) and complete verification (usually top-down from the specification towards the code). The former approach present many similarities to verify checkers, whereas the latter starts with a full formalization of underlying algorithm. I compare the approaches and show where the main challenges are.

Mathias Fleury and Armin Biere: Discussion: How to combine and compare options in solvers?

Comparing options between solvers is a complicated task. There are three main ways: runtime options (with the risk of not understanding requirements between features), compile-time option (with the issue of testing and making the code very complicated), or different versions of the source code. A second question is how to measure the performance without implementing the most advanced version. This (short) talk should serve as a basis for discussion on how to organize the development of a solver with various options.

Vijay Ganesh: A SAT Solver + Computer Algebra Attack on the Minimum Kochen-Specker Problem

One of the most fundamental results in the foundations of quantum mechanics is the Kochen–Specker (KS) theorem, a ‘no-go’ theorem which states that contextuality is an essential feature of any hidden-variable theory. The theorem hinges on the existence of a mathematical object called a KS vector system. Although the existence of a KS vector system was first established by Kochen and Specker, the problem of the minimum size of such a system has stubbornly remained open for over 50 years. In this paper, we present a new method based on a combination of a SAT solver and a computer algebra system (CAS) to address this problem. We improve the lower bound on the minimum number of vectors in a KS system from 22 to 23 and improve the efficiency of the search by a factor of over 1000 when compared to the most recent computational methods. Finding a minimum KS system would simplify experimental tests of the KS theorem and have direct applications in quantum information processing, specifically in the security of quantum cryptographic protocols based on complementarity, zero-error classical communication, and dimension witnessing.

Ambros Gleixner: Algorithmic Mixed Integer Programming: Between Exactness and Performance in Theory and Practice

Today’s state-of-the-art solvers for the general class of mixed integer programs exhibit both exact and heuristic properties. Both aspects are crucial for their lasting relevance in academic and industrial practice. In this talk, we give an overview of methods implemented and successfully used in mixed-integer programming solvers and try to point out connections to satisfiability solving and pseudo-Boolean optimization. We conclude by outlining our efforts to address the ubiquitous use of floating-point arithmetic in virtually all fast mixed integer programming solvers and report advances in performant roundoff-error-free MIP solving with proof logging.

Malte Helmert: Introduction to Automated Planning

In my talk, I will introduce the classical planning problem and explain its relevance to the seminar by contrasting it with SAT. For those that haven’t seen planning before, I hope to provide some basic understanding of the problem and why it is of interest. For those familiar with planning, I hope to give one or two additional perspectives on the problem and its complexity. Time permitting, I will also update the seminar participants on research in the planning community that tackles the main motivating questions of the seminar, in particular discussing results and open challenges for certifying planning algorithms.

Marijn Heule: The Packing Chromatic Number of the Infinite Square Grid is 15

A packing k-coloring of a graph $G = (V, E)$ is a mapping from V to $\{1, \ldots, k\}$ such that any pair of vertices u, v that receive the same color c must be at distance greater than c in G. Arguably the most fundamental problem regarding packing colorings is to determine the packing chromatic number of the infinite square grid. Various works in the last 20 years improved the bounds of this problem. We finally solve it and show that the answer is 15. A crucial part of our solution is a novel encoding that reduces the runtime by a factor of 30. Moreover, we construct and validate a DRAT proof of unsatisfiability for the direct encoding of the problem. This proof includes the symmetry-breaking and reencoding techniques that we applied.

Matti Järvisalo: Pseudo-Boolean Optimization by Implicit Hitting Sets

Recent developments in applying and extending Boolean satisfiability (SAT) based techniques have resulted in new types of approaches to pseudo-Boolean optimization (PBO), complementary to the more classical integer programming techniques. In this paper, we develop the first approach to pseudo-Boolean optimization based on instantiating the so-called implicit hitting set (IHS) approach, motivated by the success of IHS implementations for maximum satisfiability (MaxSAT). In particular, we harness recent advances in native reasoning techniques for pseudo-Boolean constraints, which enable efficiently identifying inconsistent assignments over subsets of objective function variables (i.e. unsatisfiable cores in the context of PBO), as a basis for developing a native IHS approach to PBO, and study the impact of various search techniques applicable in the context of IHS for PBO.

This talk is based on the following papers:

- Improvements to the Implicit Hitting Set Approach to Pseudo-Boolean Optimization. Pavel Smirnov, Jeremias Berg, and Matti Järvisalo. SAT 2022
- Pseudo-Boolean Optimization by Implicit Hitting Sets. Pavel Smirnov, Jeremias Berg, and Matti Järvisalo. CP 2021

Daniela Kaufmann: Exploring Algebraic Methods for Circuit Verification

Digital circuits are widely utilized in computers, because they provide models for various digital components and arithmetic operations. To avoid problems like the infamous Pentium FDIV bug, it is critical to ensure that these circuits are correct. Formal verification can be used to evaluate whether a circuit meets a given specification. Arithmetic circuits, in particular integer multipliers, pose a challenge to current verification approaches. Techniques that rely solely on SAT solving or decision diagrams appear incapable of tackling this problem in an acceptable period of time. In practice, circuit verification still requires a substantial amount of manual labor.

In this talk, we will demonstrate an automated verification technique that is based on algebraic reasoning and is currently considered to be one of the most successful verification methods for circuit verification. In this approach the circuit is modeled as a set of polynomial equations that is implied by the circuit. For a correct circuit, we must demonstrate that the specification is implied by the polynomial representation of the given circuit. However, some sections of the multiplier, such as final stage adders, are difficult to check using simply computer algebra. To address this issue, we will provide a hybrid solution that blends SAT and computer algebra.

But who verifies the verifier? The ability to independently generate and check proof certificates boosts confidence in the outcomes of automated reasoning tools. We present an algebraic proof calculus that allows us to obtain certificates as a by-product of circuit verification and that can be efficiently verified with our independent proof checking tools.

Laura Kovács and Martin Suda: First-Order Theorem Proving

First-order theorem proving is undergoing a rapid development thanks to its successful use in software analysis, formal verification, IT security, symbolic computation, theorem proving in mathematics, and other related areas. Breakthrough results in all areas of theorem proving have been obtained, including improvements in theory, implementation, and the development of powerful theorem proving tools.

This talks serves as a mini-tutorial on the theory and practice of first-order theorem proving. We introduce the core concepts of automating first-order theorem proving in first-order logic with equality. We will discuss the resolution and superposition calculus, introduce the saturation principle, present various algorithms implementing redundancy elimination, and demonstrate how these concepts are implemented in our Vampire theorem prover. We survey practical consideration for making saturation efficient and focus on applications of SAT solvers in the efficient automation of first-order reasoning.

Chunxiao (Ian) Li: Towards a Deeper Understanding of Modern CDCL SAT Solvers

Understanding why state-of-the-art SAT solvers are empirically successful has been a long standing question since the beginning of solver research. An ideal answer to the question should be both empirically verifiable, and in the same time can be theoretically analyzed. To shed light on this difficult problem, I will present a novel concept for SAT formulas and proofs, namely the hierarchical community structure (HCS). Empirically we show that hierarchical community structure can be used to distinguish industrial formulas from random, crafted and crypto formulas. And theoretically, we prove size upper bounds parameterized in the HCS structure. I will also discuss some recent developments in the proof locality through the lens of HCS.

Meena Mahajan: Quantified Boolean Formulas: Solving and Proof Complexity

QBF solving brings many new challenges and has thrown up many innovative approaches and heuristics. QBF proof complexity explores the theoretical underpinnings of these approaches rigorously, explains relative strengths of different approaches, exposes limitations, and suggests new approaches. This talk will survey some of the developments in the area.

Ciaran McCreesh: How Constraint Programming Isn't Like SAT

This talk provides an overview of modern constraint programming and how it differs from SAT solving, both in technology and terminology. I'll give an introduction to how the CP community thinks and speaks, starting with modelling and reformulation; then constraints, propagation, and lazy clause generation; and finally, search. Next we'll take a closer look at the all-different constraint: I'll explain how it's propagated and why CNF can't do the same thing, and then we'll look at whether stronger propagation is actually a good idea in practice. I'll conclude with a look at three exciting research topics: proof logging, belief propagation, and parallel search.

Kuldeep Meel: Designing Samplers is Easy: The Boon of Testers

Uniform sampling is a fundamental problem with a wide variety of applications. Recently, Chakraborty and Meel designed the first scalable sampling tester, Barbarik, based on a grey-box sampling technique for testing if the distribution, according to which the given sampler is sampling, is close to the uniform or far from uniform. The availability of Barbarik has the potential to spur the development of samplers and testing techniques such that developers can design sampling methods that can be accepted by Barbarik even though these samplers may not be amenable to a detailed mathematical analysis.

In this work, we present the realization of this aforementioned promise. Based on the flexibility offered by CryptoMiniSat, we design a sampler CMSGen that promises the achievement of the sweet spot of the quality of distributions and runtime performance. In particular, CMSGen achieves significant runtime performance improvement over the existing samplers. We conduct two case studies, and demonstrate that the usage of CMSGen leads to significant runtime improvements in the context of combinatorial testing and functional synthesis.

(Joint work with Priyanka Golia, Mate Soos, and Sourav Chakraborty)

Jakob Nordström: Proof Complexity and SAT Solving

This talk is intended to give an overview of proof complexity and connections to Boolean satisfiability (SAT) solving. The focus will be on proof systems (and corresponding algorithms) such as resolution (DPLL and conflict-driven clause learning), Nullstellensatz and polynomial calculus (linear algebra and Gröbner basis computations), and cutting planes (pseudo-Boolean solving and 0-1 integer linear programming). We will also discuss briefly proof systems such as Sherali-Adams and sums of squares (linear programming and semidefinite programming hierarchies), stabbing planes (0-1 ILP), and extended resolution (SAT pre- and inprocessing).

Andy Oertel: Certified CNF Translations for Pseudo-Boolean Solving

The dramatic improvements in Boolean satisfiability (SAT) solving since the turn of the millennium have made it possible to leverage state-of-the-art conflict-driven clause learning (CDCL) solvers for many combinatorial problems in academia and industry, and the use of proof logging has played a crucial role in increasing the confidence that the results these solvers produce are correct. However, the fact that SAT proof logging is performed in conjunctive normal form (CNF) clausal format means that it has not been possible to extend guarantees of correctness to the use of SAT solvers for more expressive combinatorial paradigms, where the first step is an unverified translation of the input to CNF.

In this work, we show how cutting-planes-based reasoning can provide proof logging for solvers that translate pseudo-Boolean (a.k.a. 0-1 integer linear) decision problems to CNF and then run CDCL. To support a wide range of encodings, we provide a uniform and easily extensible framework for proof logging of CNF translations. We are hopeful that this is just a first step towards providing a unified proof logging approach that will also extend to maximum satisfiability (MaxSAT) solving and pseudo-Boolean optimization in general.

This is joint work with Stephan Gocht, Ruben Martins and Jakob Nordström.

Andre Schidler, Friedrich Slivovski, and Stefan Szeider: Scalable optimization with SAT-based local improvement (SLIM)

SAT-based local improvement (SLIM) is an optimization metaheuristic. It repeatedly employs SAT-based solvers to local versions of the problem instance at hand, gradually improving a heuristically computed initial global solution. SLIM has been successfully instantiated for several problems, including graph decomposition and coloring, decision tree induction, Bayesian network structure learning, and circuit synthesis.

Yong Kiam Tan: The Last Mile in Trustworthy Automated Reasoning

State-of-the-art automated reasoning tools are complex and highly-optimized pieces of software. This complexity can lead to an increased risk of soundness-critical bugs, which may affect the trustworthiness of automatically generated results. To remedy this state of affairs, many tools now generate proof logs (or proof certificates), which can be independently checked for correctness.

This talk is about the "last mile" in highly trustworthy automated reasoning—the development of efficient, formally verified proof checkers that are capable of soundly scrutinizing proof logs for various theories. I will survey theories, proof systems, and proof checkers that have been formalized in proof assistants, including my work with various collaborators on verifying proof checkers using HOL4 and CakeML. Looking ahead, I speculate that today's foundational software verification tools are well-suited to handle tougher challenges in end-to-end verification of proof checkers. For example, 1) building common infrastructure to ease verification of new proof systems and/or efficient proof checkers; 2) developing a unified proof checker that seamlessly handles proofs from different theories; or 3) verifying proof checkers for proof systems that feature probabilistic or interactive elements.

Marc Vinyals: Theoretical limits of 1UIP Learning

Even though CDCL can reproduce resolution proofs with at most a polynomial overhead, it is not clear how large that overhead needs to be, or if one is needed at all. We investigate the role that learning schemes play in this simulation by focusing on syntactical properties of proofs generated by CDCL solvers that employ the standard 1UIP learning scheme. In particular we show that proofs of this kind can simulate resolution proofs with at most a linear overhead, but there also exist formulas where such overhead is necessary or, more precisely, that there exist formulas with resolution proofs of linear length that require quadratic CDCL proofs.

Ryan Williams: Around the Complexity of SAT

I'll do my best to give a gentle overview of the state of the art in worst-case SAT solving, as of October 2022. We'll talk briefly about the algorithms, and also about impossibility conjectures. We'll see the Exponential Time Hypothesis (ETH), the Strong ETH, and even the Super Strong ETH, which are increasingly stronger assertions about how hard SAT is to solve in the worst case. We'll see some of the vast implications of these hypotheses, and discuss their potential truth (or falsity).

Emre Yolcu: Exponential separations using guarded extension variables

We study the complexity of proof systems augmenting resolution with inference rules that allow, given a formula F in conjunctive normal form, deriving clauses that are not necessarily logically implied by F but whose addition to F preserves satisfiability. When the derived clauses are allowed to introduce variables not occurring in F, the systems we consider become equivalent to extended resolution. We are concerned with the versions of these systems "without new variables." They are called BC-, RAT-, SBC-, and GER-, denoting respectively blocked clauses, resolution asymmetric tautologies, set-blocked clauses, and generalized extended resolution. Each of these systems formalizes some restricted version of the ability to make assumptions that hold "without loss of generality," which is commonly used informally to simplify or shorten proofs. Except for SBC-, these systems are known to be exponentially weaker than extended resolution. They are, however, all equivalent to it under a relaxed notion of simulation that allows the translation of the formula along with the proof when moving between proof systems. By taking advantage of this fact, we construct formulas that separate RAT- from GER- and vice versa. With the same strategy, we also separate SBC- from RAT-. Additionally, we give polynomial-size SBC- proofs of the pigeonhole principle, which separates SBC- from GER- by a previously known lower bound. These results also separate the three systems from BC- since they all simulate it. We thus give an almost complete picture of their relative strengths.