ISSN 2186-7437

NII Shonan Meeting Report

No. 180

The Art of SAT

Nikolaj Bjørner
Marijn Heule
Tachio Terauchi

October 2-5, 2023

National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-Ku, Tokyo, Japan

The Art of SAT

Organizers:
Nikolaj Bjørner (Microsoft Research)
Marijn Heule (Carnegie Mellon University)
Tachio Terauchi (Waseda University

October 26, 2023

Background and introduction

Modern Satisfiability (SAT) solvers have in the past decades reached a stunning level of maturity, impacting a vast array of applications in Computer Science and Mathematics. Developing efficient SAT technologies draws on deep scientific results and amounts to an art pursued by a highly sophisticated community around SAT solvers. The solutions and proofs of combinatorial problems produced by SAT solvers reveal patterns that can provide mathematical insights and result in artistic images. In order to stimulate advances in SAT solving, radically new innovations are needed. To leverage the advances in SAT solving, expansive collaborations are sought. The seminar “Art of SAT” was held at Shonan conference center October 2-5 bringing together experts from the international SAT community and practitioners and theoreticians from Japan and neighboring countries. It fostered stronger connections and a broadened the research community around SAT and automated reasoning.

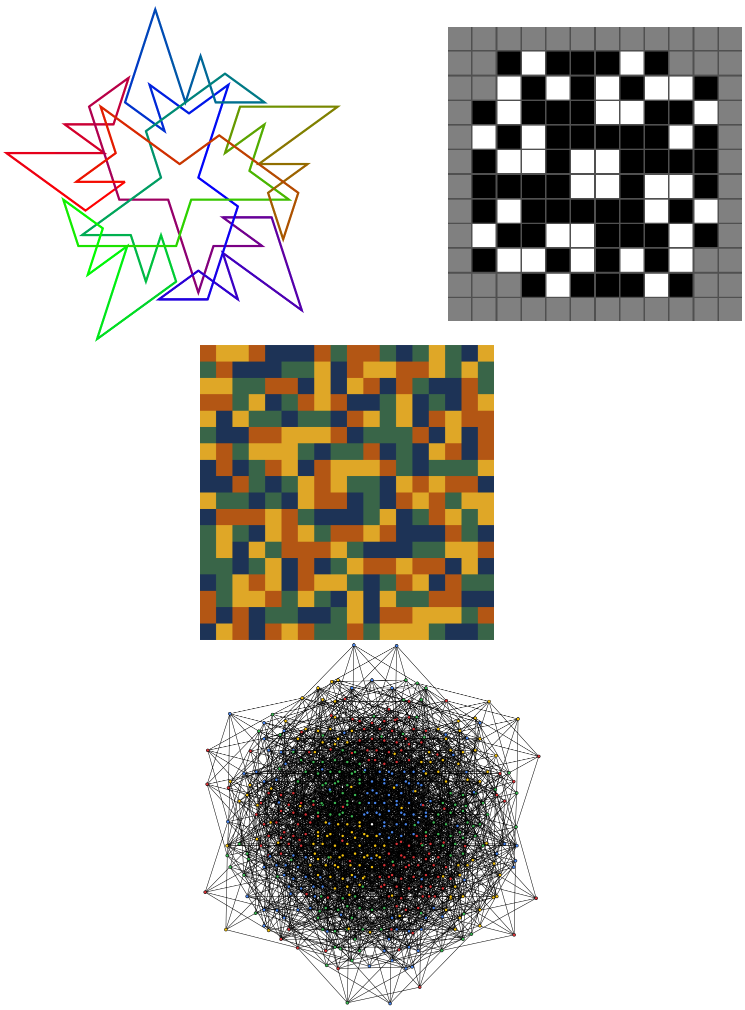

Figure 1: Art obtained via SAT solving of combinatorial problems

The Art of SAT

Building high-performance SAT solvers is an art that requires deep expert knowledge and access to ideas that have been developed within small communities.

Extracting Information from SAT Results (Or: Art *from* SAT)

In recent years, SAT solvers have been used to solve long-standing open problems in mathematics, including the Pythagoreans triples problem, Erdős discrepancy problem, Keller's conjecture and the chromatic number of the plane. Some mathematicians argue that these solutions don't contribute to understanding. However, the solutions and proofs produced by SAT solvers reveal interesting patterns and can provide mathematical insights and result in artistic images. Also, there has been much work on using solutions and proofs produced by SAT solvers for various applications in computer science, such as using interpolants computed from the unsatisfiability proofs produced by SAT solvers for program verification and synthesis. A goal of the seminar at Shonan has been to explore which useful information can be extracted from SAT solutions and proofs from various applications.

International Synergies and Opportunities

Tools built around SAT solvers abound and the seminar included several users of SAT and SMT technologies.

Active areas pursued by the international community of SAT researchers include, to mention a few, using lookahead search techniques to speed up parallel SAT solving; using machine learning to configure and tune SAT solvers; integrate specialized solvers for Boolean functions, such as pseudo-Boolean inequalities and parity constraints; new inference mechanisms that extends the power of SAT solvers beyond the limitations of resolution. The seminar included experts in these domains where they shared their insights, inspired and strengthened research in SAT solving.

Overview of the meeting

The meeting was organized as a combination of seminar presentations centered around the topics of Art of SAT, and discussions of challenge problems. It brought together experts in SAT solving from Japan, Taiwan, Europe and North America. Anchored by a keynote by Armin Biere on igniting the SAT revolution through breakthroughs in SAT solving technologies, the topics of the seminar extended to using SAT solvers for establishing theorems in combinatorial mathematics, encoding techniques into SAT, industrial planning, program verification and synthesis and SMT solving. The confluences offered by the composition of participants and their backgrounds led to numerous exchanges and advances.

The program together with several of the slides used during the seminar is available online:

`https://nikolajbjorner.github.io/ShonanArtOfSAT/program.html`

Overview of Talks

The Art of Igniting the SAT Revolution

Armin Biere

This talk expanded and presented additional preliminary results related to our POS'23 presentation on the SAT Museum, where the virtual SAT Solver Museum is an effort towards preserving historical SAT solvers, by collecting and porting their source code to modern compilers and evaluating them on representative benchmark sets on the same hardware. This allows us to compare historic and modern solvers in the same environment.

A SAT Solver + Computer Algebra Attack on the Minimum Kochen–Specker Problem

Vijay Ganesh

Two of the most fundamental theorems in quantum foundations are the Kochen-Specker and the Conway-Kochen "Free Will" Theorems. Both these theorems rely on the existence of a finite object called the KS vector system. Despite several decades of effort by leading physicists and mathematicians, the minimum KS system problem for 3-dimensions remains unresolved. In this talk, we present a SAT solver + Computer Algebra (CAS) attack on this problem, and provide a verified computational proof that there are no KS systems of size 23 or below. To the extent that we know, ours is the first attack on a problem in quantum foundations via a combination of a SAT solver and CAS system.

Second-Order Quantified Boolean Logic

Jie-Hong Roland Jiang

Second-order quantified Boolean formulas (SOQBFs) generalize quantified Boolean formulas (QBFs) by admitting second-order quantifiers on function variables in addition to first-order quantifiers on atomic variables. Recent endeavors establish that the complexity of SOQBF satisfiability corresponds to the exponential-time hierarchy (EXPH), similar to that of QBF satisfiability corresponding to the polynomial-time hierarchy (PH). This fact reveals the succinct expression power of SOQBFs in encoding decision problems not efficiently doable by QBFs. In this paper, we investigate the second-order quantified Boolean logic with the following main results: First, we present a procedure of quantifier elimination converting SOQBFs to QBFs and a game interpretation of SOQBF semantics. Second, we devise a sound and complete refutation-proof system for SOQBF. Third, we develop an algorithm for countermodel extraction from a refutation proof. Finally, we show potential applications of SOQBFs in system design and multi-agent planning. With these advances, we anticipate practical tools for development.

Modular Primal-Dual Fixpoint Logic Solving for Temporal Verification

Hiroshi Unno

We present a novel approach to deciding the validity of formulas in first-order fixpoint logic with background theories and arbitrarily nested inductive and co-inductive predicates defining least and greatest fixpoints. Our approach is constraint-based, and reduces the validity checking problem of the given first-order-fixpoint logic formula (formally, an instance in a language called µCLP) to a constraint satisfaction problem for a recently introduced predicate constraint language.

Coupled with an existing sound-and-relatively-complete solver for the constraint language, this novel reduction alone already gives a sound and relatively complete method for deciding µCLP validity, but we further improve it to a novel modular primal-dual method. The key observations are (1) µCLP is closed under complement such that each (co-)inductive predicate in the original primal instance has a corresponding (co-)inductive predicate representing its complement in the dual instance obtained by taking the standard De Morgan's dual of the primal instance, and (2) partial solutions for (co-)inductive predicates synthesized during the constraint solving process of the primal side can be used as sound upper-bounds of the corresponding (co-)inductive predicates in the dual side, and vice versa. By solving the primal and dual problems in parallel and exchanging each others' partial solutions as sound bounds, the two processes mutually reduce each others' solution spaces, thus enabling rapid convergence. The approach is also modular in that the bounds are synthesized and exchanged at granularity of individual (co-)inductive predicates.

We demonstrate the utility of our novel fixpoint logic solving by encoding a wide variety of temporal verification problems in µCLP, including termination/non-termination, LTL, CTL, and even the full modal µ-calculus model checking of infinite state programs. The encodings exploit the modularity in both the program and the property by expressing each loops and (recursive) functions in the program and sub-formulas of the property as individual (possibly nested) (co-)inductive predicates. Together with our novel modular primal-dual µCLP solving, we obtain a novel approach to efficiently solving a wide range of temporal verification problems.

The Art of Symmetry Breaking: Isomorph-Free Generation of Combinatorial Objects with SAT Modulo Symmetries

Stefan Szeider

SAT modulo Symmetries (SMS) is a framework for the exhaustive isomorph-free generation of combinatorial objects with a prescribed property. SMS relies on the tight integration of a CDCL SAT solver with a custom dynamic symmetry-breaking algorithm that iteratively refines an ordered partition of the generated object's elements. SMS utilizes the IPASIR-UP interface and hence can work with any solver implements this interface. SMS supports DRAT proofs for the SAT solver's reasoning and offline verification of the symmetry-breaking clauses, and thus provides an additional layer of confidence in the obtained results. This talk will discuss the basic concepts of SMS and review some recent applications on graphs, digraphs, hypergraphs, and matroids. Based on joint work with Katalin Fazekas, Markus Kichweger, Tomas Peitl, Manfred Scheucher, Tianwei Zhang. Documentation on *readthedocs* (`https://sat-modulo-symmetries.readthedocs.io/`).

Satisfiability and Term Rewriting

Akihisa Yamada

SAT and SMT are central tools in automatically proving termination of term rewrite systems (TRSs). In this talk, I would like to 1. explain the use of interactive SMT solving in my termination tool NaTT, and present ideas for further improvement, for which I am not sure how to encode as standard SMT problems. 2. investigate a DPLL(T)-like approach for "satisfiability modulo rewriting" (also called (in)feasibility). Such problems appear in termination and confluence analysis of (conditional) TRSs, but so far only expensive tree-automata-based techniques or basic syntactic techniques are known.

Repairing DoS Vulnerability of Real-World Regexes

Tachio Terauchi

There has been much work on synthesizing and repairing regular expressions (*regexes* for short) from examples. These *programming-by-example* (PBE) methods help the users write regexes by letting them reflect their intention by examples. However, the existing methods may generate regexes whose matching may take super-linear time and are vulnerable to regex denial of service (ReDoS) attacks. This paper presents the *first* PBE repair method that is guaranteed to generate only invulnerable regexes. Importantly, our method can handle *real-world regexes* containing *lookarounds* and *backreferences*. Due to the extensions, the existing formal definitions of ReDoS vulnerabilities that only consider pure regexes are insufficient. Therefore, we first give a novel *formal semantics and complexity of backtracking matching algorithms for real-world regexes*, and with them, give the *first formal definition of ReDoS vulnerability for real-world regexes*. Next, we present a novel condition called *real-world strong 1-unambiguity* that is sufficient for guaranteeing the invulnerability of real-world regexes, and formalize the corresponding PBE repair problem. Finally, we present an algorithm that solves the repair problem. The algorithm builds on and extends the previous PBE methods to handle the real-world extensions and with constraints to enforce the real-world strong 1-unambiguity condition.

IPASIR-UP: User Propagators for CDCL

Katalin Fazekas

Modern SAT solvers are frequently embedded as sub-reasoning engines into more complex tools for addressing problems beyond the Boolean satisfiability problem. Examples include solvers for Satisfiability Modulo Theories (SMT), combinatorial optimization, model enumeration and counting. In such use cases, the SAT solver is often able to provide relevant information beyond the satisfiability answer. Further, domain knowledge of the embedding system (e.g., symmetry properties or theory axioms) can be beneficial for the CDCL search, but cannot be efficiently represented in clausal form.

In this talk, I will present IPASIR-UP, a general interface that allows to inspect and influence the internal behaviour of CDCL SAT solvers. The interface captures the most essential functionalities that are sufficient to simplify and improve use cases that require a more fine-grained interaction with the SAT solver than provided via the standard IPASIR interface. First I will illustrate the main functionalities provided by IPASIR-UP. Following that, I will briefly describe some open challenges in using IPASIR-UP in combination with some complex crucial features of modern SAT solvers, such as inprocessing and proof production. At the end, I will shortly present possible ways to address some of these challenges.

This is based on joint work with Aina Niemetz, Mathias Preiner, Markus Kirchweger, Stefan Szeider, and Armin Biere.

CaDiCaL(T): CaDiCaL as CDCL(T) Engine in cvc5

Mathias Preiner

cvc5 is a state-of-the-art SMT solver based on the CDCL(T) framework, which tightly integrates theory solvers with a CDCL SAT solver at its core. The CDCL(T) framework requires a tight integration with the SAT solver in a way that allows the theory layer to interact with the SAT solver during search, i.e., in an online fashion. For this purpose, cvc5 integrates a heavily customized version of MiniSat 2.2.0, which was extended to allow this online interaction.

In this talk, I will discuss the integration of CaDiCaL as CDCL(T) SAT engine in cvc5 via the recently proposed IPASIR-UP interface. IPASIR-UP is an interface for interactive incremental SAT solving for interacting with a CDCL SAT solver during search. I will explain in detail how the notifications and callbacks of IPASIR-UP are used to establish the communication between cvc5's theory solvers and CaDiCaL and how we use it to inspect and influence the CDCL search. I will conclude with an evaluation on SMT-LIB and discuss some challenges we encountered on incremental SMT-LIB problems.

Bit-Blasting Meets Local Search in Bitwuzla

Aina Niemetz

Reasoning about quantifier-free bit-vector constraints in Satisfiability Modulo Theories (SMT) has been an ongoing challenge for many years, especially for large bit-widths. Current state-of-the-art for bit-precise reasoning is a technique called bit-blasting, where bit-vector constraints are eagerly translated into propositional logic (SAT). Bit-blasting is very efficient in practice but does not generally scale well for large bit-widths due to fact that the translation is in general exponential in the size of the input formula, which potentially (and in practice) overwhelms the underlying SAT solver. For these instances, we need alternative approaches for bit-precise reasoning that do not rely on translations to the SAT level. Such an alternative approach is our propagation-based local search procedure, which relies on propagating target values from top-level constraints towards the inputs while utilizing so-called invertibility conditions. Invertibility conditions precisely characterize when bit-vector constraints are invertible, a core concept of our approach. Our procedure is, as expected for local search, incomplete in the sense that it can only determine satisfiability but was shown to be effective on hard satisfiable instances, in particular in combination with bit-blasting in a sequential portfolio setting. In this talk, I will talk about the strengths and potential weaknesses of this approach, how to address these weaknesses, and the challenge to combine local search with bit-blasting, as implemented in our new SMT solver Bitwuzla, in a more hybrid manner that allows information sharing between the two approaches.

Hamiltonian Cycle Reconfiguration with Answer Set Programming

Mutsunori Banbara

The Hamiltonian cycle reconfiguration problem is defined as determining, for a given Hamiltonian cycle problem and two among its feasible solutions, whether one is reachable from another via a sequence of feasible solutions subject to certain transition constraints. We develop an approach to solving the Hamiltonian cycle reconfiguration problem based on Answer Set Programming (ASP). Our approach relies on a high-level ASP encoding and delegates both the grounding and solving tasks to an ASP-based solver. To show the effectiveness of our approach, we conduct experiments on the benchmark set of Flinders Hamiltonian Cycle Project.

CoRe Challenge 2022/2023: International Competition for Combinatorial Reconfiguration

Takehide Soh

Combinatorial Reconfiguration is a novel algorithmic concept that provides mathematical models and analysis for "transformations over state spaces." Its appearance ranges from theory to applications. However, its technical achievements are hard to access. Thus, it is required to found a common infrastructure for utilizing and applying the algorithmic technology of combinatorial reconfiguration. See this website for more backgrounds. Combinatorial Reconfiguration Challenge (CoRe Challenge 2022/2023) is a competition aiming for practically exploring the combinatorial reconfiguration. CoRe Challenge 2022/2023 targets the Independent Set Reconfiguration (ISR) problem. The ISR problem is one of the most well-studied reconfiguration problems. Theoretically, the problem is PSPACE-complete, which implies that there exist instances such that even a shortest reconfiguration sequence requires a super polynomial steps.

SAT-Based Explicit LTLf Satisfiability Checking... and Beyond

Kristin Yvonne Rozier

Driven by the many pertinent applications in robotics, planning, and AI-based applications, we present a SAT-based framework for LTLf (Linear Temporal Logic on Finite Traces) satisfiability checking, first published in AAAI 2019. We use propositional SAT-solving techniques to construct a transition system for the input LTLf formula; satisfiability checking then reduces to a path-search problem over this transition system. Furthermore, we introduce CDLSC (Conflict-Driven LTLf Satisfiability Checking), a novel algorithm that leverages information produced by propositional SAT solvers from both satisfiability and unsatisfiability results. Experimental evaluations show that CDLSC significantly outperforms all other existing approaches for LTLf satisfiability checking, by demonstrating an approximate four-fold speed-up compared to the second-best solver. We then highlight enduring challenges for temporal logic satisfiability, including other temporal logics like LTL and MLTL. We conclude with an overview of currently-open research questions.

The Art of Encoding Happy Endings

Marijn Heule

In 1970s Erdős asked whether, for every integer k, every set of sufficiently many points in the plane in general position contains a k-hole, that is, a subset of k points spanning a convex polygon which contains no other points of the set. Denote by $h(k)$ the minimum number of points needed to guarantee the existence of k-hole, if it exists. While $h(3) = 3$ and $h(4) = 5$ are straightforward, Harborth (1978) proved $h(5) = 10$, and Horton (1983) constructed arbitrary large sets without 7-holes. The existence of 6-holes, however, remained open until 2006, when Gerken and Nicolas independently proved that $h(6)$ is finite.

We use SAT to show that every set of 30 points yields a 6-hole. The key contributions are a compact $O(n^4)$ encoding and a partitioning of the problem that allows us to have linear time speedups even when using 1000s of cores. Together with the set of 29 points without 6-holes constructed by Overmars (2002), this determines the last remaining value $h(6) = 30$.

A one-size-fits-all proof logging system?

Jakob Nordström

We propose a unified proof logging system for decision problems, optimization problems, model enumeration problems, and problem reformulations, and semantics for composition of such proofs. We discuss some of the challenges in designing such a proof system and the choices made to overcome them. We base our system on pseudo-Boolean reasoning with 0-1 integer linear programs, which is a superset of conjunctive normal form (CNF). This is the only proof system that can currently support all enhanced SAT solving techniques for decision problems, and the language of 0-1 linear inequalities also makes it convenient to reason about linear objective functions. However, to maintain equisatisfiability of decision problems restrictions for how contraints are deleted must be imposed, and this is also crucial to preserve the optimal value of optimization problem instances. In addition, for projected model enumeration, the use of strengthening rules such as RAT and pseudo-Boolean generalizations of RAT must be carefully restricted in order not to change the set of solutions. Finally, precise definitions are needed of what it would mean to rewrite the objective function and/or constraints of a problem instance while keeping it "the same", and what the semantics should be for proof compositions if a sequence of solvers operations on (translations of) the same problem instance and emit separate proof logs.

This is based on joint work with Bart Bogaerts, Stephan Gocht, Ciaran McCreesh, Magnus O. Myreen, Andy Oertel, and Yong Kiam Tan.

Lemur: Integrating Large Language Models in Automated Program Verification

Nina Narodytska

The demonstrated code-understanding capability of LLMs raises the question of whether they can be used for automated program verification, a task that typically demands high-level abstract reasoning about program properties that is challenging for verification tools. We propose a general methodology to combine the power of LLMs and automated reasoners for automated program verification. We formally describe this methodology as a set of derivation rules and prove its soundness. We instantiate the calculus as a sound automated verification procedure, which led to practical improvements on a set of synthetic and competition benchmarks.

Optimal Scheduling for a Vertical Transport Machine with MaxSAT

Miyuki Koshimura

This study considers optimal scheduling for a vertical transport machine (VTM) with MaxSAT. The VTM is a part of the automated warehouse which stores lots of baggage in shelves. Many pieces of baggage arriving from various locations are transported to designated floors by conveyors and VTM, and stored on shelves on that floor. The movement of the baggage in the VTM is described as a set of Boolean formulas, and those costs are represented by a set of weighted Boolean formulas. The MaxSAT solver finds out an optimal scheduling as a model satisfying the formulas with the minimum cost. Experimental results show that the proposed method can solve the practical problems in reasonable time.

BNN verification dataset for Max-SAT Evaluation 2020

Masahiro Sakai

Deep neural networks achieved impressive performance in various tasks, but formal verification of their behavior is a challenging task. [Narodytska+, AAAI-18] proposed a method to verify a specific type of neural network called binarized neural networks (BNNs) using a SAT solver. Among other properties, they checked the (ϵ, p)-robustness of a neural network for an input x, i.e. absence of a perturbation τ such that $||\tau||p \leq \epsilon$ and the neural network misclassifies $x + \tau$. We extend their approach and consider an optimization problem that minimizes $||\tau||$ instead of giving the upper bound ϵ a priori. The optimization problem can be encoded as partial weighted Max-SAT, and we submitted problem instances to Max-SAT Evaluation 2020. In this talk, I will introduce our encoding and report some results.

Incremental Maximum Satisfiability: Interfaces, Solvers and Applications

Matti Järvisalo

Enabling incremental computations on the level of constraint optimization remains a noticeable challenge. Boolean satisfiability (SAT) solvers allow for incremental computations, which is key to efficient employment of SAT solvers iteratively for developing complex decision and optimization procedures, including maximum satisfiability (MaxSAT) solvers. However, while incremental computations have been identified to have great potential in speeding up especially so-called unsatisfiability-based MaxSAT solving, enabling incremental computations in MaxSAT has remained until very recently to most extent unexplored. In this talk, I will give an overview of recent developments in incremental MaxSAT, including an API supporting the development of incremental solvers and their applications; the new Incremental Track of MaxSAT Evaluations; current availability of incremental MaxSAT solvers with some more details on SAT-IP hybrid approach of implicit hitting sets implemented in iMaxHS; and a glance at recent successful applications of incremental MaxSAT solving.

Trace-Guided Inductive Synthesis of Recursive Functional Programs

Roopsha Samanta

We propose a novel trace-guided approach to tackle the challenges of ambiguity and generalization in synthesis of recursive functional programs from examples. Our approach augments the search space of programs with recursion traces consisting of sequences of recursive subcalls of programs. Our method is based on a new version space algebra (VSA) for succinct representation and efficient manipulation of pairs of recursion traces and programs that are consistent with each other. We have implemented this approach in a tool called SyRup and evaluate it on benchmarks from prior work. Our evaluation demonstrates that SyRup not only requires fewer examples to achieve a certain success rate than existing synthesizers, but is also less sensitive to the quality of the examples.

Higher-order model checking and its similarity (?) with SAT solving.

Naoki Kobayashi

In this talk, I first give a brief overview of how my past work on automated program verification used SAT/SMT solving. I then provide a brief introduction to higher-order model checking (which is a "higher-order extension" of finite state model checking), and discuss the similarities and differences between higher-order model checking and SAT solving.

List of Participants

- Mutsunori Banbara, Nagoya University
- Armin Biere, Johannes Kepler University
- Nikolaj Bjørner, Microsoft Research
- Sam Buss, University of California, San Diego
- Katalin Fazekas, TU Wien
- Vijay Ganesh, Waterloo University
- Marijn Heule, Carnegie Mellon University
- Katsumi Inoue, NII
- Matti Järvisalo, University of Helsinki
- Jie-Hong Jiang, National Taiwan University
- Naoki Kobayashi, University of Tokyo
- Miyuki Koshimura, Kyushu University
- Oliver Kullmann, Swansea University
- Shin-ichi Minato, Kyoto University
- Nina Narodytska, Vmware
- Aina Niemetz, Stanford University
- Jakob Nordström, University of Copenhagen
- Mathias Preiner, Stanford University
- Kristin Yvonne Rozier, Iowa State
- Masahiro Sakai, Preferred Networks
- Roopsha Samanta, Purdue University
- Stefan Szeider, TU Wien
- Soh Takahide, Kobe University
- Terauchi Terauchi, Waseda University
- Sophie Tourret, INRIA
- Hiroshi Unno, University of Tsukuba
- Akihisa Yamada, National Institute of Informatics

Meeting Schedule

Check-in Day: October 1 (Sun)

- Welcome Banquet

Day1: October 2 (Mon)

Morning 1:

- 9-10:00: Participants introduce themselves.
- 10:00-10:30: break

Morning 2:

- 10:30-11:30 Armin Biere **The Art of igniting the SAT revolution**
- 11:30-12:00 Vijay Ganesh: **A SAT Solver + Computer Algebra Attack on the Minimum Kochen–Specker Problem**

Afternoon 1: Topic Higher- order SAT.

- 14:00-14:30 Jie-Hong Roland Jiang **Second-order Boolean QBF**
- 14:30-15:00 Hiroshi Unno **Modular Primal-Dual Fixpoint Logic Solving for Temporal Verification**
- 15:00-15:30 Sam Buss **Seeking hard SAT instances for constant depth Frege**
- 15:30-16:00 break

Afternoon 2: Encoding SAT

- 16:00-16:30 Stefan Szeider **isomph-free SAT modulo symmetries**
- 16:30-17:00 Akihisa Yamada **Satisfiability and Term Rewriting**
- 17:00-17:30 Tachio Terauchi **Synthesizing regular expressions**
- 17:30-18:00 Discussion

Evening: Demos and social interaction.

Day2: October 3 (Tue)

Morning 1: Topic session on building SAT services on top of SAT solvers.

- 9-9:30: Katalin Fazekas **IPASIR-UP: User Propagators for CDCL**
- 9:30-10:00 Mathias Preiner **CaDiCaL(T): CaDiCaL as CDCL(T) Engine in cvc5**
- 10:00-10:30 Aina Niemetz **Bit-Blasting Meets Local Search in Bitwuzla**

Morning 2: Applications of SAT

- 11:00-11:30 Mutsunori Banbara **Hamiltonian Cycle Reconfiguration with Answer Set Programming**
- 11:30-12:00 Takehide Soh **CoRe Challenge 2022/2023: International Competition for Combinatorial Reconfiguration**

Group Photo

Afternoon 1: Topic session on encoding (part II).

- 14:00-14:30 Kristin Yvonne Rozier **SAT based explicit LTLf satisfiability checking**
- 14:30-15:10 Marijn Heule **The art of encoding happy endings**
- 15:10-15:40 Nina Narodytska **Invariant generation using LLMs**
- 15:40-16:00 break

Afternoon 2: Topic session on MaxSAT. Breakout session for collaborations.

- 16:00-16:30 Miyuki Koshimura **Optimal Scheduling for a Vertical Transport Machine with MaxSAT**
- 16:30-17:00 Masahiro Sakai **BNN verification dataset for MaxSAT Evaluation 2020**
- 17:00-18:00 Discussions and Collaborations

Evening: Social interaction.

Day3: October 4 (Wed)

Morning 1: Certificates, Search, and Beyond

- 9-9:45: Jakob Nordström **A one-size-fits-all proof logging system?**
- 9:45-10:15 Shin-ichi Minato **The Art of Counting Graphs**

Morning 2:

- 11:00-11:30 Matti Järvisalo **Incremental MaxSAT**
- 11:30-12:00 Oliver Kullmann **Transforming QBFs Using Biclique Covers**

Afternoon: excursion and main banquet

Day4: October 5 (Thu)

Morning 1:

- 9:15-10:15 Discussion on challenge problems
  - Marijn Heule Simple to generate, hard to solve SAT problems
  - Stefan Szeider Coloring maps between the Earth and Moon
  - Oliver Kullmann Automating Cube and Conquer

- 10:15-10:45 Break

Morning 2:

- 10:45-11:10 Roopsha Samanta **Trace-Guided Inductive Synthesis of Recursive Functional Programs**
- 11:10-11:15 Nikolaj Bjørner **Introduction to the z3guide**
- 11:15-11:35 Naoki Kobayashi **Higher-order model checking and its similarity (?) with SAT solving**
- 11:35-11:40 Jie-Hong Roland Jiang **SAT 2024 conference announcement**

Summary of discussions

The seminar used a session for discussing challenge problems. It was led by Marijn Heule, Stefan Szeider and Oliver Kullmann.

Marijn Heule Simple to generate, hard to solve SAT problems. Most wanted Folkman graph. Finding smaller and smaller Folkman graphs has been an active research area for 70 years. Erdos awarded researchers twice for their improvements. In an interesting twist, we found some graphs that we expect to have the Folkman properties, but the SAT instances are too hard. This is somewhat surprising as they are relatively small and very structured. These graphs are excellent challenges for SAT solvers.

Stefan Szeider Coloring maps between the Earth and Moon. The Earth-Moon coloring problem, https://en.wikipedia.org/wiki/Earth-Moon_problem, considers k-colorability for maps over other topologies than the planar field. It is generally unsolved and the discussion centered around using SAT solving to establish bounds.

Oliver Kullmann Automating Cube and Conquer. Cube and Conquer is used to partition sub-goals to a point where the goals are easily solved using CDCL. It has been successful for solving several combinatorial problems. However, the state-of-art techniques require domain tuned splitting heuristics and weights. The discussion centered around drawing inspiration from heuristics for weighted non-binary search trees and quantum-inspired optimization.

Summary of new findings

SAT solving Starting with a historic comparison of modern SAT solving advances, a clear set of recent transformative innovations were identified. The integration of local search with SAT solving was extended to Bit-vector domains, where preliminary experiments reported in the seminar showed a need to identify how to control phase caching in light of local search assignments. An integration of IPASIR-UP with cvc5 demonstrated advances. It also established how SMT solver heuristics influenced design choices for handling local assumptions used for incremental solving.

SAT and Computer Algebra and proofs Integrations of CAD systems for enumerating graphs modulo symmetries were discussed in a series of contributed talks. They present an exciting area for integration of extended symbolic solving capabilities with SAT engines with the aim at setting long-standing open conjectures in combinatorics.

Beyond SAT Quantified Boolean Logics were extended beyond first-order by a system that handles second order quantification over predicates. Fixed-point solving was extended to dual least and greatest fixed point equations. Higher-order model checking was connected to intrisics of SAT solving advances. Each of the reported advances illustrated uses of core SAT techniques beyond propositional SAT.

SAT and applications SAT solvers were used in rewriting termination engines, in synthesizing regular expressions, Hamiltonian cycle problems, integration of LLMs for invariant synthesis, trace-guided synthesis of recursive functions, and encoding LTLf satisfiability.

SAT and competitions Advances in SAT solving are incentiviced by competitions evaluating previous and new solvers. Among the competitions discussed were the competitions on Combinatorial Reconfiguration, path enumerations, incremental MaxSAT, and MaxSAT solving for Binarized Neural Networks.

SAT and encodings A cross cutting theme of the seminar was methods for encoding (combinatorial) problems into SAT. It was elaborated in a presentation covering the k-hole problem. A very different manifestation of encodings emerged in a discussion about finding minimal bi-cliques for simplifying QBF problems.

MaxSAT Several discussions included SAT in MaxSAT engines. New results and a competition of MaxSAT solvers were discussed. Complementarily, a use of MaxSAT solving for vertical transport machines was presented along with BNN verification datasets.

Identified issues and future directions

The seminar "Art of SAT" highlighted the significant progress in the development of Satisfiability (SAT) solvers and their impact on various fields in Computer Science and Mathematics. However, several issues and future directions were identified:

Need for Innovation: While SAT technologies have matured, there is a need for radically new innovations to stimulate further advances in SAT solving.

Collaboration: Expansive collaborations are sought to leverage the advances in SAT solving3. This includes bringing together experts from the international SAT community and practitioners and theoreticians from various countries.

Understanding Patterns: The solutions and proofs produced by SAT solvers reveal patterns that can provide mathematical insights. There is a potential to further explore these patterns for a deeper understanding of combinatorial problems.

Community Building: Events like the "Art of SAT" seminar help in fostering stronger connections and broadening the research community around SAT and automated reasoning. More such events could be beneficial for the field.