Report from Dagstuhl Seminar 23261

# SAT Encodings and Beyond

# Marijn J. H. Heule<sup>\*1</sup>, Inês Lynce<sup>\*2</sup>, Stefan Szeider<sup>\*3</sup>, and Andre Schidler<sup>†4</sup>

- 1 Carnegie Mellon University Pittsburgh, US. marijn@cmu.edu
- 2 University of Lisbon, PT. ines.lynce@tecnico.ulisboa.pt
- 3 TU Wien, AT. stefan@szeider.net
- 4 TU Wien, AT. aschidler@ac.tuwien.ac.at

#### — Abstract -

This report documents the program and the outcomes of Dagstuhl Seminar 23261 "SAT Encodings and Beyond." The seminar facilitated an intense examination and discussion of current results and challenges related to encodings for SAT and related solving paradigms. The seminar featured presentations and group work that provided theoretical, practical, and industrial viewpoints. The goal was to foster more profound insights and advancements in encoding techniques, which are pivotal in enhancing solvers' efficiency.

Seminar June 25–30, 2023 – https://www.dagstuhl.de/23261

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Constraint and logic programming; Theory of computation  $\rightarrow$  Discrete optimization

**Keywords and phrases** constraint propagation, lower and upper bounds, problem formulation, propositional satisfiability, symmetry breaking

Digital Object Identifier 10.4230/DagRep.13.6.106

### 1 Executive Summary

Marijn J. H. Heule (Carnegie Mellon University - Pittsburgh, US) Inês Lynce (University of Lisbon, PT) Stefan Szeider (TU Wien, AT)

License ☺ Creative Commons BY 4.0 International license © Marijn J. H. Heule, Inês Lynce, and Stefan Szeider

The propositional satisfiability problem (SAT) is one of the most fundamental problems in computer science. As the first problem shown to be NP-complete by the Cook-Levin Theorem, SAT remains a fundamental benchmark problem for complexity theory. In contrast to its theoretical hardness, research over the last 20 years has successfully designed and engineered powerful algorithms for the SAT problem, called SAT solvers, that are surprisingly efficient on problem instances that arise from real-world applications. However, to solve a problem with a SAT solver or a related tool, one must first formulate the problem in terms of propositional logic to be digestible by the solver. This translation from the original problem to propositional logic is often called a SAT encoding. The encoding itself is often the crucial part that determines whether the solver can solve the problem efficiently, making the encoding techniques at least as important as the solving techniques. Hence, much effort has been put into researching efficient encoding techniques.

Except where otherwise noted, content of this report is licensed under a Creative Commons BY 4.0 International license SAT Encodings and Beyond, *Dagstuhl Reports*, Vol. 13, Issue 6, pp. 106–122 Editors: Marijn J. H. Heule, Inês Lynce, Stefan Szeider, and Andre Schidler DAGSTUHL Dagstuhl Reports Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

<sup>\*</sup> Editor / Organizer

<sup>&</sup>lt;sup>†</sup> Editorial Assistant / Collector

### Marijn J. H. Heule, Inês Lynce, Stefan Szeider, and Andre Schidler

Other previous scientific meetings primarily focused on solving techniques, not on encodings. Hence, this Dagstuhl Seminar provided an overdue opportunity for an in-depth discussion of the state-of-the-art of encodings and future challenges and research avenues. When planning the seminar, we identified the following five critical topics.

**The Effectiveness of Encodings.** Current challenging research questions are new encodings for global constraints, theoretical lower and upper bounds on encoding size for global constraints, and new methods for symmetry breaking. Topics of interest are general principles of problem reformulations and their impact on the effectiveness of encodings.

**The Complexity of Encodings.** Although state-of-the-art SAT solvers can deal with millions of clauses and variables, the size of the original instance must be significantly smaller since the encoding often causes a polynomial (often cubic or worse) size blow-up. Which methods can overcome these limitations?

**Encoding Tools.** To fully exploit the power of SAT solvers, researchers have designed high-level languages that are amenable to describing constraints and developed compilers for converting constraints into CNF. Exciting topics for discussions include the questions of how to obtain an optimal hybridization of encodings and how to decompose global constraints.

**Lazy Encodings.** An interesting approach to SAT encodings is to start with an incomplete under-constrained encoding and add clauses to it once a solution has been found that violates properties that are not considered by the encoding. SAT modulo Theories and Lazy Clause Generation are among the approaches utilizing eager encodings.

**Verifying Encodings.** Trust in the correctness of SAT-solving results increased significantly in the last couple of years as all top-tier solvers can produce proofs of unsatisfiability that can be validated using efficient and formally verified tools. An interesting topic is how the encoding part of the toolchain can be sufficiently validated.

**Beyond SAT.** The success of SAT solving has spawned the development of efficient solvers for problems that are more general than SAT, including MaxSAT, QBF-SAT, PB, ASP, and CP. These more general problems require new encoding techniques.

We invited key researchers to cover these topics and were happy that most of the people we wished for accepted the invitation. Hence, we could approach participants individually to solicit longer survey talks to cover these topics by top experts. Shorter, focused talks complemented these longer survey-like talks. The talks covered various encoding aspects for particular solving paradigms, including SAT, CP, ASP, MaxSAT, and QBF.

We were delighted to have the *industrial perspective* covered by Andreas Falkner (Siemens AG), who presented challenges in industrial product configuration.

Other talks were devoted to symmetry-breaking techniques that boost SAT-based combinatorial search, which included a live demo of the SMS tool; another focus of several talks was the verification of results obtained via encodings. Some talks explored the theoretical limits of encodings and the connection between computer algebra systems and SAT encodings.

In addition to the talks, we had an *open-problems and challenges* session and dedicated time for group work. The posed problems asked for desirable properties for proof logging, how encodings can ensure that propagation on a high level implies propagation on a low level, how encodings for enumeration and counting can be established, how one can measure the usefulness of auxiliary variables in encodings, how to verify that an encoding is correct, and the exact computational complexity of minimal resolution proof length (in binary). Also, efficient encodings for several concrete problems were posed, including Golumb Rulers,

23261

the Connect-4 game, the metric dimension of hypercubes, the independent configuration problem, problems related to Steiner Triples, line arrangements with a limited number of triangles, and block designs that appear in product configuration. We formed working groups to tackle some of these problems and had a brief session where progress on these problems was reported and discussed.

Overall, we are pleased with the outcome of the seminar. We have met our objectives and started a highly stimulating discussion and exchange of ideas, covering the state of the art and future challenges. Still, it also became clear that encodings are a far-reaching topic that leaves many challenging open questions for future work. So, a follow-up Dagstuhl Seminar in the future is strongly indicated.

# 2 Table of Contents

Executive Summary Marijn J. H. Heule, Inês Lynce, and Stefan Szeider		
Verified encodings for SAT solvers Cayden Codel and Marijn J. H. Heule	1	
Breaking Symmetries when Solving Hard Combinatorial Problems Michael Codish	2	
FPT-reductions to SAT – And SAT encodings for problems in FPT Ronald de Haan	2	
Challenges in industrial product configuration Andreas Falkner	3	
Reasoning-Enabling Encodings Marijn J. H. Heule	3	
SAT-Based Judgment Aggregation Matti Järvisalo	4	
SAT encodings from a Contraint Programming perspective: Why and Why not <i>George Katsirelos</i>	4	
Combining SAT and Computer Algebra for Circuit Verification Daniela Kaufmann	5	
Isomorph-Free Generation of Combinatorial Objects With SAT Modulo Symmetries Markus Kirchweger and Stefan Szeider	5	
Automatic Tabulation in Constraint Models         Zeynep Kiziltan	5	
An iterative university course timetabling tool with MaxSAT Inês Lunce	6	
Some connections between encodings and circuits Stefan Menael	6	
Certified CNF Translations for Pseudo-Boolean Solving Andy Oertel	6	
Co-Certificate Learning with SAT Modulo Symmetries Tomáš Peitl, Markus Kirchweger, and Stefan Szeider	7	
Exact resolution complexity Tomáš Peitl and Stefan Szeider	7	
SAT-based Local Improvement Method Vaidyanathan Peruvemba Ramaswamy, Andre Schidler, and Stefan Szeider 11	9	
Structures from Combinatorial Geometry and their Encodings Manfred Scheucher	0	

# 110 23261 – SAT Encodings and Beyond

	Solutions of Quantified Boolean Formulas Martina Seidl and Sibylle Möhle	120
	Encoding MiniZinc for SAT, MaxSAT and QUBO Guido Tack	120
	Encodings of Collatz-like problems into termination of string rewriting <i>Emre Yolcu</i>	121
Pa	articipants	122

## **3** Overview of Talks

### 3.1 SAT and Computer Algebra

Curtis Bright (University of Windsor, CA)

License 
 © Creative Commons BY 4.0 International license
 © Curtis Bright

 Main reference Curtis Bright, Ilias S. Kotsireas, Vijay Ganesh: "When satisfiability solving meets symbolic computation", Commun. ACM, Vol. 65(7), pp. 64–72, 2022.
 URL https://doi.org//10.1145/3500921

Combining satisfiability (SAT) solvers with computer algebra systems (CASs) progress has enabled progress on problems requiring search and sophisticated mathematics [1]. In this talk, I will outline problems I have worked on in which the SAT+CAS method outperformed pure SAT or pure CAS approaches by orders of magnitude. For example, the SAT+CAS method found the first Williamson matrices of order 70 [2], certified the nonexistence of finite projective planes of order 10 [3], demonstrated a Kochen–Specker vector system in three dimensions must have size at least 24 [4, 5], and has improved certain side-channel attacks on integer factorization [6].

### References

- C. Bright, I. Kotsireas, V. Ganesh. When Satisfiability Checking Meets Symbolic Computation. Communications of the ACM, 2022.
- 2 C. Bright, I. Kotsireas, V. Ganesh. Applying Computer Algebra Systems with SAT Solvers to the Williamson Conjecture. Journal of Symbolic Computation, 2020.
- 3 C. Bright, K. Cheung, B. Stevens, I. Kotsireas, V. Ganesh. A SAT-based Resolution of Lam's Problem. AAAI 2021.
- 4 Z. Li, C. Bright, V. Ganesh. An SC-Square Approach to the Minimum Kochen–Specker Problem, SC-Square Workshop, 2022.
- 5 Z. Li, C. Bright, V. Ganesh. A SAT Solver + Computer Algebra Attack on the Minimum Kochen–Specker Problem, arXiv:2306.13319, 2023.
- 6 Y. Ajani, C. Bright. A Hybrid SAT and Lattice Reduction Approach for Integer Factorization, SC-Square Workshop, 2023.

### 3.2 Verified encodings for SAT solvers

Cayden Codel (Carnegie Mellon University - Pittsburgh, US) and Marijn J. H. Heule (Carnegie Mellon University - Pittsburgh, US)

License 
 Creative Commons BY 4.0 International license
 Cayden Codel and Marijn J. H. Heule

 Joint work of Cayden R. Codel, Jeremy Avigad, Marijn J. H. Heule
 Main reference Cayden R. Codel, Jeremy Avigad, Marijn J. H. Heule: "Verified Encodings for SAT Solvers," to appear in FMCAD 2023.
 URL http://crcodel.com/research/fmcad2023.pdf

SAT is a powerful tool for solving a wide array of problems, but many problems are not expressed in propositional logic and must instead be encoded into SAT. These encodings are often subtle, and implementations are error-prone. Formal correctness proofs are needed to ensure that implementations are bug-free.

In this talk, we present a library for formally verifying SAT encodings, written using the Lean interactive theorem prover. Our library currently contains verified encodings for the parity, at-most-one, and at-most-k constraints. It also contains methods of generating fresh

### 112 23261 – SAT Encodings and Beyond

variable names and combining sub-encodings to form more complex ones, such as one for encoding a valid Sudoku board. The proofs in our library are general, and so this library serves as a basis for future encoding efforts.

### 3.3 Breaking Symmetries when Solving Hard Combinatorial Problems

Michael Codish (Ben Gurion University - Beer Sheva, IL)

License  $\textcircled{\textbf{ co}}$  Creative Commons BY 4.0 International license  $\textcircled{\textbf{ co}}$  Michael Codish

Many hard combinatorial problems involve huge numbers of symmetries which derive from isomorphic representations of objects in the search space. Restricting search to avoid symmetries – aka "symmetry breaking" – makes a big difference when trying to solve such problems.

Symmetry breaking in constraint programming is often achieved by introducing symmetry breaking constraints which are satisfied by at least one member of each isomorphism class. Complete symmetry breaking constraints are satisfied by exactly one member from each class and other symmetry breaks are called partial.

In this talk I will focus mainly on breaking symmetries in graph search problems. The search for complete symmetry breaking constraints for graph search problems is itself a hard problem and it is unknown if there exists a complete symmetry breaking constraint that is polynomial in the size of the graph.

In computer science when the problem is hard – we typically follow one or more from three alternatives: (1) clever brute force computation, (2) approximation algorithms, or (3) identifying special cases where the problem is easier.

This talk will focus on how each of these three alternatives comes into play when solving hard graph search problems.

### 3.4 FPT-reductions to SAT – And SAT encodings for problems in FPT

Ronald de Haan (University of Amsterdam, NL)

In this talk, we will discuss some results and some research directions that connect the theory of parameterized complexity and the theory of encodings and solvers for SAT and related problems such as ASP.

The talk can be divided into roughly two parts. The first part addresses the (im)possibility of fixed-parameter tractable encodings into SAT. This makes sense for problems whose (classical) complexity is beyond NP, and so one does not hope for polynomial-time encodings. For suitable choices of parameters, the problem could be encoded in fpt-time to SAT. We will give a few examples of cases where this is possible, and we present a parameterized complexity toolbox that can be used to assess the (im)possibility of fpt-time encodings into SAT.

The second part addresses an ongoing research direction, that revolves around encoding fpt-time solvable problems into SAT in such a way that CDCL solvers are guaranteed to run in fixed-parameter tractable time. For some problems, one can do this in such a way that this works with any branching heuristic, and for some problems the choice of branching heuristic makes a difference. We will present some examples illustrating this (for SAT and ASP), and then we raise some open research questions in this arena.

### 3.5 Challenges in industrial product configuration

Andreas Falkner (Siemens AG - Wien, AT)

License	Creative Commons BY 4.0 International license
	© Andreas Falkner
Main reference	Andreas A. Falkner, Alois Haselböck, Gerfried Krames, Gottfried Schenner, Herwig Schreiner,
	Richard Taupe: "Solver Requirements for Interactive Configuration", J. Univers. Comput. Sci.,
	Vol. 26(3), pp. 343–373, 2020.
URL	http://www.jucs.org/jucs_26_3/solver_requirements_for_interactive
Main reference	Andreas A. Falkner, Gerhard Friedrich, Konstantin Schekotihin, Richard Taupe, Erich Christian
	Teppan: "Industrial Applications of Answer Set Programming", Künstliche Intell., Vol. 32(2-3),
	pp. 165–176, 2018.
URL	https://doi.org//10.1007/S13218-018-0548-6
Main reference	Andreas A. Falkner, Gerhard Friedrich, Alois Haselböck, Gottfried Schenner, Herwig Schreiner:
	"Twenty-Five Years of Successful Application of Constraint Technologies at Siemens", AI Mag.,
	Vol. 37(4), pp. 67–80, 2016.
URL	https://doi.org//10.1609/AIMAG.V37I4.2688
Main reference	Andreas A. Falkner, Anna Ryabokon, Gottfried Schenner, Kostyantyn M. Shchekotykhin: "OOASP:
	Connecting Object-Oriented and Logic Programming", in Proc. of the Logic Programming and
	Nonmonotonic Reasoning - 13th International Conference, LPNMR 2015, Lexington, KY, USA,
	September 27-30, 2015. Proceedings, Lecture Notes in Computer Science, Vol. 9345, pp. 332–345,
	Springer, 2015.
URL	https://doi.org//10.1007/978-3-319-23264-5_28
Main reference	Deepak Dhungana, Andreas A. Falkner, Alois Haselböck: "Generation of conjoint domain models for
	system-of-systems", in Proc. of the Generative Programming: Concepts and Experiences, GPCE'13,
	Indianapolis, IN, USA - October 27 - 28, 2013, pp. 159–168, ACM, 2013.
URL	https://doi.org//10.1145/2517208.2517224
Main reference	Andreas A. Falkner, Alois Haselbock, Gottfried Schenner, Herwig Schreiner: "Modeling and solving
	technical product configuration problems", Artif. Intell. Eng. Des. Anal. Manuf., Vol. 25(2),
	pp. 115–129, 2011.
	https://doi.org//10.101//S089000410000570
Main reference	Deepak Dhungana, Andreas A. Falkner, Alois Haselbock: "Configuration of Cardinality-Based
	reature models using Generative Constraint Satisfaction", in Proc. of the 37th EUROMICRO
	Conference on Software Engineering and Advanced Applications, SEAA 2011, Ould, Finland, August
וסון	but - September 2, 2011, pp. 100–103, IEEE Computer Society, 2011.
UKL	https://doi.org//10.1109/SEAA.2011.24

Product configuration has been among the first successful applications of symbolic AI, e.g. translating feature models with cross-tree constraints to SAT encodings and finding consistent solutions. Despite the high maturity of state-of-the-art tools, encoding remains challenging in practice: dynamic size (i.e. unbounded multiplicities of variables and constraints), OO-like inheritance (for clearer knowledge representation), open domains, merging of subsystem encodings (from distributed authors), multi-dimensional optimization, reconfiguration and knowledge evolution, explanations and recommendations, debugging, etc.

## 3.6 Reasoning-Enabling Encodings

Marijn J. H. Heule (Carnegie Mellon University - Pittsburgh, US)

License ☺ Creative Commons BY 4.0 International license ◎ Marijn J. H. Heule

A common approach in automated reasoning is to encode a given problem into propositional logic and then solve the resulting formula with a satisfiability (SAT) solver. As the quality of the encoding has a big impact on solver performance, it is no coincidence that solvers are

### 114 23261 – SAT Encodings and Beyond

highly successful in the field of hardware verification: digital electronic circuits have a direct encoding into propositional logic which is often adequate for solving. However, the same is not true for many other applications. Sophisticated encodings may be required to efficiently solve some problems using SAT solvers. This talk will focus on sophisticated encodings of some hard-combinatorial problems for which a straightforward encoding is ineffective. We will first describe general techniques to produce high-quality encodings. Afterward, we will present encodings for specific problems: edge-matching puzzles, Hamiltonian cycles, and packing colorings.

### 3.7 SAT-Based Judgment Aggregation

Matti Järvisalo (University of Helsinki, FI)

License 

 © Creative Commons BY 4.0 International license
 © Matti Järvisalo

 Joint work of Ari Conati, Andreas Niskanen, Matti Järvisalo
 Main reference Ari Conati, Andreas Niskanen, Matti Järvisalo: "SAT-Based Judgment Aggregation", in Proc. of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS '23, p. 1412–1420, International Foundation for Autonomous Agents and Multiagent Systems, 2023.
 URL https://dl.acm.org/doi/abs/10.5555/3545946.3598792

Judgment aggregation (JA) offers a generic formal logical framework for modeling various settings where agents must reach joint agreements through aggregating the preferences, judgments, or beliefs of individual agents by social choice mechanisms. In this work, we develop practical JA algorithms for outcome determination by harnessing Boolean satisfiability (SAT) based solvers as the underlying reasoning engines, leveraging on their ability to efficiently reason over logical representations incrementally. Concretely, we provide algorithms for outcome determination under a range of aggregation rules, using natural choices of SAT-based techniques adhering to the computational complexity of the problem for the individual rules. We also implement and empirically evaluate the approach using both synthetic and PrefLib data, showing that the approach can scale significantly beyond recently proposed alternative algorithms for JA.

# 3.8 SAT encodings from a Contraint Programming perspective: Why and Why not

George Katsirelos (INRAE - Palaiseau, FR)

Encoding constraints to CNF is an attractive option for CP solvers, especially in the context of clause learning. However, it is not always a preferable or even feasible option, depending on our requirements. In this talk, I will go over some cases where SAT encodings have been used successfully in CP solvers, as well as some cases where it does not work out as well. I will point out some theoretical work for why this is the case, as well as some practical reasons for it.

### 3.9 Combining SAT and Computer Algebra for Circuit Verification

Daniela Kaufmann (TU Wien, AT)

Verifying multiplier circuits is an important problem which in practice still requires substantial manual effort. In this talk, I will demonstrate that encoding the entire problem into SAT is not the ideal strategy, nor is using a pure algebraic encoding. We use a combination of SAT and computer algebra in our method to significantly improve automated verification of integer multipliers.

## 3.10 Isomorph-Free Generation of Combinatorial Objects With SAT Modulo Symmetries

Markus Kirchweger (TU Wien, AT) and Stefan Szeider (TU Wien, AT)

License 
 Creative Commons BY 4.0 International license
 Markus Kirchweger and Stefan Szeider

 Main reference Markus Kirchweger, Stefan Szeider: "SAT Modulo Symmetries for Graph Generation", in Proc. of the 27th International Conference on Principles and Practice of Constraint Programming, CP 2021, Montpellier, France (Virtual Conference), October 25-29, 2021, LIPIcs, Vol. 210, pp. 34:1–34:16, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

URL https://doi.org//10.4230/LIPICS.CP.2021.34

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 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 of its applications on graphs, digraphs, hypergraphs, and matroids. At the end of the talk, we will give a live demo of the tool.

### 3.11 Automatic Tabulation in Constraint Models

Zeynep Kiziltan (University of Bologna, IT)

 License 

 Creative Commons BY 4.0 International license
 Zeynep Kiziltan

 Joint work of 

 Özgür Akgün, Ian P. Gent, Christopher Jefferson, Zeynep Kiziltan, Ian Miguel, Peter Nightingale, András Z. Salamon, Felix Ulrich-Oltean

 Main reference 

 Özgür Akgün, Ian P. Gent, Christopher Jefferson, Zeynep Kiziltan, Ian Miguel, Peter Nightingale, András Z. Salamon, Felix Ulrich-Oltean: "Automatic Tabulation in Constraint Models", CoRR, Vol. abs/2202.13250, 2022.

 URL https://arxiv.org/abs/2202.13250

The performance of a constraint model can often be improved by converting a sub-problem into a single table constraint, which is referred to as tabulation. In this talk, I will describe an automatic tabulation approach, implemented in Savile Row which is a constraint model reformulation tool. Savile Row takes as input a model described in the high-level solverindependent modelling language Essence Prime and has backends for CP, SAT and ILP solvers. Our approach to automatic tabulation deploys heuristics to discover opportunities for tabulation and uses a specific propagator or an encoding for the generated table constraint depending on the chosen backend solver.

### 3.12 An iterative university course timetabling tool with MaxSAT

Inês Lynce (University of Lisbon, PT)

License 

 © Creative Commons BY 4.0 International license
 © Inês Lynce

 Joint work of Inês Lynce, Alexandre Lemos, Pedro T. Monteiro
 Main reference Alexandre Lemos, Pedro T. Monteiro, Inês Lynce: "Introducing UniCorT: an iterative university course timetabling tool with MaxSAT", J. Sched., Vol. 25(4), pp. 371–390, 2022.
 URL https://doi.org//10.1007/S10951-021-00695-6

This work describes the UniCorT tool designed to solve university course timetabling problems specifically tailored for the 2019 International Timetabling Competition (ITC). The proposed approach includes pre-processing, the use of a maximum satisfiability (MaxSAT) solver and a local search procedure. The impact of a handful of techniques in the quality of the solution and the execution time is evaluated. We take into account different pre-processing techniques and CNF encodings, as well as the combination with a local search procedure. The success of our tool is attested by having been ranked among the five finalists of the ITC 2019 competition.

### 3.13 Some connections between encodings and circuits

Stefan Mengel (CNRS, CRIL - Lens, FR)

License ☺ Creative Commons BY 4.0 International license © Stefan Mengel

In the literature, there are several results making SAT-encodings and Boolean circuits. In particular, it is known that different classes of circuits correspond tightly to encodings with specific properties, e.g. restricted (tree/clique-)width or propagation strength. In this talk, I will survey some of these connections and point out some open questions.

### 3.14 Certified CNF Translations for Pseudo-Boolean Solving

Andy Oertel (Lund University, SE)

Licen	se 🐵 Creative Commons BY 4.0 International license
	© Andy Oertel
Joint work	of Stephan Gocht, Ruben Martins, Jakob Nordström, Andy Oertel
Main referen	ce Stephan Gocht, Ruben Martins, Jakob Nordström, Andy Oertel: "Certified CNF Translations for
	Pseudo-Boolean Solving", in Proc. of the 25th International Conference on Theory and Applications
	of Satisfiability Testing, SAT 2022, August 2-5, 2022, Haifa, Israel, LIPIcs, Vol. 236, pp. 16:1–16:25
	Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.
U	RL https://doi.org//10.4230/LIPICS.SAT.2022.16

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

### Marijn J. H. Heule, Inês Lynce, Stefan Szeider, and Andre Schidler

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 published at SAT'22.

### 3.15 Co-Certificate Learning with SAT Modulo Symmetries

Tomáš Peitl (TU Wien, AT), Markus Kirchweger (TU Wien, AT), and Stefan Szeider (TU Wien, AT)

License 

 © Creative Commons BY 4.0 International license
 © Tomáš Peitl, Markus Kirchweger, and Stefan Szeider

 Main reference Markus Kirchweger, Tomás Peitl, Stefan Szeider: "Co-Certificate Learning with SAT Modulo Symmetries", in Proc. of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China, pp. 1944–1953, ijcai.org, 2023.

 URL https://doi.org/10.24963/IJCAI.2023/216

We present a new SAT-based method for generating all graphs up to isomorphism that satisfy a given co-NP property. Our method extends the SAT Modulo Symmetry (SMS) framework with a technique that we call co-certificate learning. If SMS generates a candidate graph that violates the given co-NP property, we obtain a certificate for this violation, i.e., a "co-certificate" for the co-NP property. The co-certificate gives rise to a clause that the SAT solver serving as SMS's backend learns as part of its CDCL procedure. We demonstrate that SMS plus co-certificate learning is a powerful method that allows us to improve the best-known lower bound on the size of Kochen-Specker vector systems, a problem that is central to the foundations of quantum mechanics and has been studied for over half a century. Our approach is orders of magnitude faster and scales significantly better than a recently proposed SAT-based method.

### 3.16 Exact resolution complexity

Tomáš Peitl (TU Wien, AT) and Stefan Szeider (TU Wien, AT)

License 
 © Creative Commons BY 4.0 International license
 © Tomáš Peitl and Stefan Szeider

 Main reference Tomás Peitl, Stefan Szeider: "Finding the Hardest Formulas for Resolution", J. Artif. Intell. Res., Vol. 72, pp. 69–97, 2021.
 URL https://doi.org//10.1613/JAIR.1.12589

This talk is based on two papers about computing shortest resolution proofs of formulas in CNF, in which we investigate encodings to compute shortest proofs of minimally unsatisfiable formulas, and of hitting formulas in particular, we compute the hardest formulas (and the hardest hitting formulas) with a small number of clauses, and discuss related questions. The abstracts of the two papers follow.

1. A CNF formula is harder than another CNF formula with the same number of clauses if it requires a longer resolution proof. In this paper, we introduce resolution hardness numbers; they give for m=1,2,... the length of a shortest proof of a hardest formula on m clauses. We compute the first 10 resolution hardness numbers, along with the corresponding hardest formulas. To achieve this, we devise a candidate filtering and symmetry breaking search scheme for limiting the number of potential candidates for hardest formulas, and an efficient SAT encoding for computing a shortest resolution proof of a given candidate formula.

2. Hitting formulas, introduced by Iwama, are an unusual class of propositional CNF formulas. Not only is their satisfiability decidable in polynomial time, but even their models can be counted in closed form. This stands in stark contrast with other polynomial-time decidable classes, which usually have algorithms based on backtracking and resolution and for which model counting remains hard, like 2-SAT and Horn-SAT. However, those resolution-based algorithms usually easily imply an upper bound on resolution complexity, which is missing for hitting formulas. Are hitting formulas hard for resolution? In this paper we take the first steps towards answering this question. We show that the resolution complexity of hitting formulas is dominated by so-called irreducible hitting formulas, first studied by Kullmann and Zhao, that cannot be composed of smaller hitting formulas. However, by definition, large irreducible unsatisfiable hitting formulas are difficult to construct; it is not even known whether infinitely many exist. Building upon our theoretical results, we implement an efficient algorithm on top of the Nauty software package to enumerate all irreducible unsatisfiable hitting formulas with up to 14 clauses. We also determine the exact resolution complexity of the generated hitting formulas with up to 13 clauses by extending a known SAT encoding for our purposes. Our experimental results suggest that hitting formulas are indeed hard for resolution.

## 3.17 SAT-based Local Improvement Method

Vaidyanathan Peruvemba Ramaswamy (TU Wien, AT), Andre Schidler (TU Wien, AT), Stefan Szeider (TU Wien, AT)

License	O Creative Commons BY 4.0 International license
	$\ensuremath{\mathbb{C}}$ Vaidyanathan Peruvemba Ramaswamy, Andre Schidler, and Stefan Szeider
Joint work of	Peruvenba Ramaswamy Vaidyanathan, Franz Xaver Reichl, Andre Schidler, Friedrich Slivovsky,
	Stefan Szeider
Main reference	Valdyanathan Peruvemba Ramaswamy, Stefan Szeider: "Learning Fast-Inference Bayesian
	Networks", in Proc. of the Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6 14, 2021
	Conference on Neural finormation rocessing Systems 2021, Neurin S 2021, December 0-14, 2021, within a pr 17852-17863 2021
LIRI	https://proceedings.neurins.cc/paper/2021/hash/94e70705efae423efda1088614128d0b-
0.12	Abstract.html
Main reference	Vaidyanathan Peruvemba Ramaswamy, Stefan Szeider: "Turbocharging Treewidth-Bounded
	Bayesian Network Structure Learning", in Proc. of the Thirty-Fifth AAAI Conference on Artificial
	Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial
	Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence,
	EAAI 2021, Virtual Event, February 2-9, 2021, pp. 3895–3903, AAAI Press, 2021.
URL	https://doi.org//10.1609/AAAI.V3515.16508
Main reference	Valdyanathan Peruvemba Ramaswamy, Stefan Szeider: "Learning large Bayesian networks with
	expert constraints, in Froc. of the Uncertainty in Artificial Intelligence, Thruster Uncertainty - Egnth Conference
	Proceedings of Machine Learning Research Vol 180 np 1592–1601 PMLR 2022
URL	https://proceedings.mlr.press/v180/peruvemba-ranaswamv22a.htm]
Main reference	Franz-Xaver Reichl, Friedrich Slivovsky, Stefan Szeider: "Circuit Minimization with QBF-Based
	Exact Synthesis", in Proc. of the Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI
	2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023,
	Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington,
	DC, USA, February 7-14, 2023, pp. 4087–4094, AAAI Press, 2023.
URL	https://doi.org//10.1609/AAAI.V3714.25524
Main reference	Andre Schldier, Stefan Szeider: "SAI-based Decision Tree Learning for Large Data Sets", in Proc. of
	In provide A publications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on
	Educational Advances in Artificial Intelligence, FAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence FAAI 2021 Virtual Event February 2-9, 2021
	pp. 3904–3912. AAAI Press. 2021.
URL	https://doi.org//10.1609/AAAI.V35I5.16509
Main reference	André Schidler, Stefan Szeider: "SAT-Boosted Tabu Search for Coloring Massive Graphs", ACM J.
	Exp. Algorithmics, Vol. 28, Association for Computing Machinery, 2023.

URL https://doi.org//10.1145/3603112

The SAT-based Local Improvement Method (SLIM) framework has yielded several competitive heuristics for a wide variety of problems such as treewidth, branchwidth, treedepth, decision trees, graph coloring, circuit minimization, etc. SLIM starts off with an initial heuristic solution and then repeatedly replaces small local parts with an improved version. The improved version is found by solving a SAT/MaxSAT/SMT encoding of the local part. This encoding must also ensure that the improved local part is still compatible with the rest of the global solution. We call this property 'replacement consistency', and this is the key challenge in each SLIM instantiation. SLIM capitalizes on the scalability of the initial heuristic algorithm and the power of modern SAT solvers to produce heuristic solutions of higher quality than simpler local search techniques. In this talk, we give an overview of the SLIM framework and then discuss some case-studies demonstrating the application of SLIM.

### 3.18 Structures from Combinatorial Geometry and their Encodings

Manfred Scheucher (TU Berlin, DE)

License 
 © Creative Commons BY 4.0 International license
 © Manfred Scheucher

 Main reference Manfred Scheucher: "Two disjoint 5-holes in point sets", Comput. Geom., Vol. 91, p. 101670, 2020.
 URL https://doi.org//10.1016/J.COMGEO.2020.101670

Point and lines are fundamental entities from geometry. We discuss Erdös-Szekeres type problems on point sets and the underlying combinatorics of point configurations and their dual structure: arrangements of lines. By slightly relaxing the geometric restrictions ("lines" dont have to be straight), we obtain so-called pseudopoint configurations and arrangements of pseudolines. While the original settings cannot be axiomized via finitely many forbidden subconfigurations unless P=NP=ETR, there are indeed purely combinatorial descriptions for "pseudo" settings which allow to make investigations using computer assistance, and in particular, using SAT.

### 3.19 Solutions of Quantified Boolean Formulas

Martina Seidl (Johannes Kepler Universität Linz, AT) and Sibylle Möhle (MPI für Informatik - Saarbrücken, DE)

```
License 
Creative Commons BY 4.0 International license
Martina Seidl and Sibylle Möhle
Joint work of Martina Seidl, Sibylle Möhle, Andreas Plank
```

In this talk, we will have a closer look at solutions of quantified Boolean formulas (QBFs), i.e., the tree models of true QBFs and the tree counter-models of false QBFs and their representations as Boolean functions. These models and counter-models are of practical interest as they contain the solutions to the application problems encoded as QBFs. We will discuss how well-understood concepts from SAT like model enumeration and model counting transfer to QBF and their (counter-)models.

### 3.20 Encoding MiniZinc for SAT, MaxSAT and QUBO

Guido Tack (Monash University - Clayton, AU)

The MiniZinc modelling language lets users express their constraint satisfaction and optimisation problems in a high-level, solver-independent way. MiniZinc supports a range of decision variable types (integer, Boolean, set, float), container types (sets, arrays, tuples, records), and a large number of pre-defined predicates and functions for typical problem domains such as scheduling, packing, rostering, network problems and many others. A MiniZinc program (usually called a "model") typically represents an entire problem class, which can be turned into a concrete problem instance by supplying values for the parameters defined by the program. MiniZinc can translate problem instances into input suitable for a variety of back-end solving formalisms, including CP (Constraint Programming), MIP (Mixed Integer Linear Programming), and SAT/MaxSAT (Boolean Satisfiability). The MiniZinc system consists of a generic, back-end independent compiler/interpreter implemented in C++, and back-end specific libraries of encodings expressed in the MiniZinc language itself. This talk will cover the basic architecture of the MiniZinc translation process, and then focus on the encodings for SAT, MaxSAT and QUBO, before giving a brief outlook on the next major version of MiniZinc that is currently under development.

# 3.21 Encodings of Collatz-like problems into termination of string rewriting

Emre Yolcu (Carnegie Mellon University - Pittsburgh, US)

License  $\bigcirc$  Creative Commons BY 4.0 International license

© Emre Yolcu Joint work of Emre Yolcu, Scott Aaronson, Marijn J. H. Heule

Main reference Emre Yolcu, Scott Aaronson, Marijn J. H. Heule: "An Automated Approach to the Collatz

Conjecture", J. Autom. Reason., Vol. 67(2), p. 15, 2023.

URL https://doi.org//10.1007/S10817-022-09658-8

I will describe two different ways of encoding Collatz-like problems into termination of string rewriting: one using a unary representation of integers and another using a mixed-base representation. When integers are represented in unary, the termination problem that corresponds to the Collatz conjecture (or even its simpler variants) does not admit proofs via natural matrix interpretations, a method widely used in proving termination of rewriting. I will sketch a proof of this impossibility result and then show a few instances where simply changing the encoding results in the termination problem becoming easy to solve (even automatically) via matrix interpretations.

### Participants

Carlos Ansotegui University of Lleida, ES Jeremias Berg University of Helsinki, FI Olaf Beyersdorff Friedrich-Schiller-Universität Jena, DE Armin Biere Universität Freiburg, DE Curtis Bright University of Windsor, CA Cayden Codel Carnegie Mellon University -Pittsburgh, US Michael Codish Ben Gurion University -Beer Sheva, IL Ronald de Haan University of Amsterdam, NL Emir Demirovic TU Delft, NL Andreas Falkner Siemens AG – Wien, AT Johannes Klaus Fichte Linköping University, SE María Andreína Francisco Rodríguez Uppsala University, SE Marijn J. H. Heule Carnegie Mellon University -Pittsburgh, US Matti Järvisalo University of Helsinki, FI

 Mikoláš Janota Czech Technical University -Prague, CZ George Katsirelos INRAE – Palaiseau, FR Daniela Kaufmann TU Wien, AT Markus Kirchweger TU Wien, AT Zeynep Kiziltan University of Bologna, IT Inês Lynce University of Lisbon, PT Vasco Manquinho INESC-ID - Lisbon, PT Valentin Mayer-Eichberger Isotronic - Berlin, DE Ciaran McCreesh University of Glasgow, GB Stefan Mengel CNRS, CRIL – Lens, FR Sibylle Möhle MPI für Informatik -Saarbrücken, DE Jakob Nordström University of Copenhagen, DK & Lund University, SE Andy Oertel Lund University, SE Sebastian Ordyniak University of Leeds, GB Tomáš Peitl TU Wien, AT

Vaidyanathan Peruvemba Ramaswamy TU Wien, AT Jussi Rintanen Aalto University, FI Torsten Schaub Universität Potsdam, DE Manfred Scheucher TU Berlin, DE Andre Schidler TU Wien, AT Martina Seidl Johannes Kepler Universität Linz. AT Carsten Sinz Hochschule Karlsruhe, DE Takehide Soh Kobe University, JP Stefan Szeider TU Wien, AT Guido Tack Monash University -Clayton, AU Hélène Verhaeghe KU Leuven, BE Hai Xia TU Wien, AT Emre Yolcu Carnegie Mellon University -Pittsburgh, US Tianwei Zhang TU Wien, AT

