SAT Solving + Isomorph-free Generation

. . . and the Quest for the Minimum Kochen–Specker System

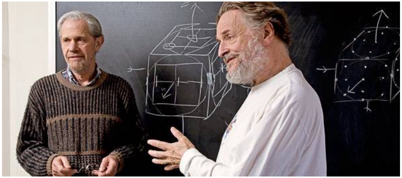
Curtis Bright
University of Windsor

May 14, 2023

Quantum Computing Academic Assembly

The Free Will Theorem

In 2006, Conway and Kochen proved the *Free Will Theorem*—if humans have have free will then so do quantum particles.[1] The assumption that humans have free will has since been removed.[2]

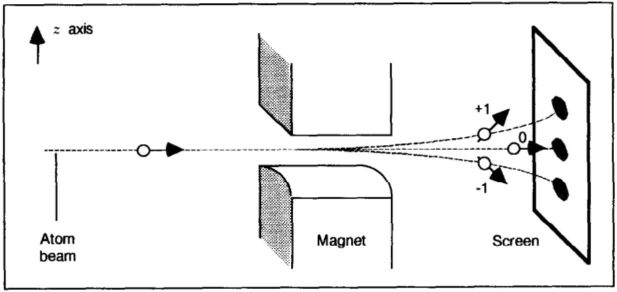


The proof relies on a finite configuration of three dimensional vectors called a Kochen–Specker (KS) system.

[1] J. Conway, S. Kochen. The Free Will Theorem. *Foundations of Physics*, 2006.
[2] S. Kochen. On the Free Will Theorem. Preprint, 2022.

The Stern–Gerlach Experiment (1922)

Shoot an atom of orthohelium through a magnetic field:



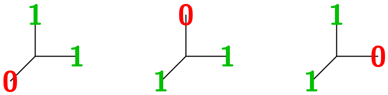
The *spin* of the atom (in this particular direction) is $+1$, -1, or 0.

Image Source: S. Weinert

The SPIN Axiom

Suppose the ±1 beams are combined producing the “squared” spin. This is **1** if the particle deflects and **0** otherwise.

The squared spin in any three mutually orthogonal directions will be **0 in exactly one of these directions**.

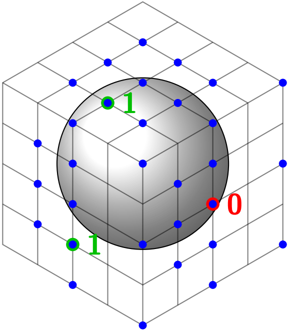


The 101 conspiracy

In particular, two orthogonal directions cannot both have a squared spin of **0**.

The KS Theorem (1967)

It is impossible to assign $\{\mathbf{0}, \mathbf{1}\}$ values to the following 31 vectors in a way that maintains the 101 conspiracy.

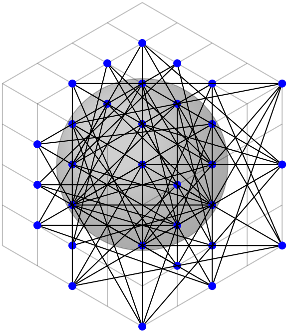


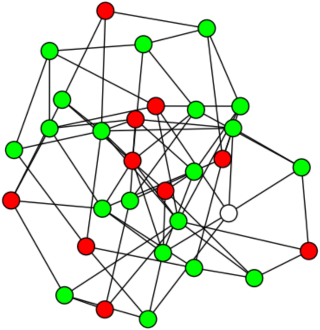
31 vector KS system of Conway and Kochen

The atom *cannot* have a predetermined spin in every direction!

KS Graphs and 101-colourability

Consider the graph formed by a KS system by connecting all pairs of orthogonal vectors:





The property required for the KS theorem is that the graph cannot be *101-coloured* (triangles have exactly one colour-**0** vertex and edges have at most one colour-**0** vertex).

Can We Do Better Than 31 Vectors?

Previously, it was known that at least 22 vectors are required.[3]

This was shown by performing an exhaustive enumeration for all **non**-101-colourable graphs with up to 21 vertices.

The computation took 75 CPU years using the state-of-the-art graph enumeration tool `geng` of `nauty`.[4]

[3] S. Uijlen, B. Westerbaan. A Kochen-Specker System Has at Least 22 Vectors. *New Generation Computing*, 2016.

[4] B. McKay, A. Piperno. Practical Graph Isomorphism, II. *Journal of Symbolic Computation*, 2014.

Properties of KS Graphs

In addition to non-101-colourability, there are a number of restrictive properties a minimal KS graph must satisfy:[5]

1. The graph must be squarefree.
2. The minimum vertex degree of the graph is at least 3.
3. Every vertex in the graph must be part of a triangle.

Previous work exhaustively enumerated graphs with properties 1–2. These are enforced by `geng` as the graph is generated vertex-by-vertex.

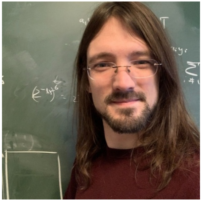
[5] F. Arends. A lower bound on the size of the smallest Kochen-Specker vector system. Master's thesis, Oxford University, 2009.

Graph Enumeration

The triangle constraint (and non-colourability) seem difficult to incorporate **during** the generation; instead, they are used as a filtering condition **after** the generation.

Unfortunately, we could not find an efficient algorithm to restrict the enumeration of graphs to those where every vertex is part of a triangle.

S. Uijlen



B. Westerbaan

SAT to the Rescue

Satisfiability (SAT) solvers take a formula in Boolean logic and try to solve it, i.e., find an assignment that makes it true.

Example: Is $(\ x\ \vee\ y\ ) \wedge (\ \neg x\ \vee\ \neg y\ )$ satisfiable?

SAT to the Rescue

Satisfiability (SAT) solvers take a formula in Boolean logic and try to solve it, i.e., find an assignment that makes it true.

Example: Is $(x \vee y) \wedge (\neg x \vee \neg y)$ satisfiable?
Yes; take x to be true and y to be false.

SAT solvers are used *declaratively*—you state the constraints of your problem, and they search for a solution. They can be amazingly effective, even for problems not arising from logic. Amazon solves a billion SAT problems every day.

A Few Uses of SAT

- 2008 Kouril and Paul determined the sixth van der Waerden number on two colours.
- 2012 Järvisalo, Kaski, Koivisto, and Korhonen found optimal constructions for Boolean circuits.
- 2013 Bundala and Zavodny computed optimal sorting networks for up to sixteen inputs.
- 2014 Konev and Lisitsa solved a special case of the Erdős discrepancy conjecture.
- 2016 Heule, Kullmann, and Marek solved the Boolean Pythagorean triples problem.
- 2020 Heule et al. resolved Keller's conjecture.
- 2021 Scheucher improved bounds on Erdős–Szekeres numbers.
- 2021 Bright et al. gave the first certifiable solution of Lam's Problem.

I have published a short paper on using SAT to solve some simple problems and may be useful as a gentle introduction.[6]

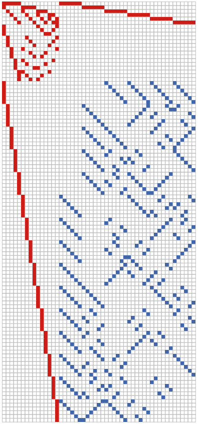
[6] C. Bright, J. Gerhard, I. Kotsireas, V. Ganesh. Effective Problem Solving Using SAT Solvers. *Maple Conference 2019*.

SAT Nonexistence Certificates

A nice benefit of SAT is that when solutions do not exist certificates are generated that can be verified independently.

The lack of verifiable certificates has real consequences. We found discrepancies in both of the independent resolutions of *Lam's problem*.

On the right is a 51-point partial projective plane of order ten asserted to not exist in 2011—but discovered by MathCheck.[7]



[7] C. Bright, K. Cheung, B. Stevens, D. Roy, I. Kotsireas, V. Ganesh. A nonexistence certificate for projective planes of order ten with weight 15 codewords. *Applicable Algebra in Engineering, Communication and Computing*, 2020.

SAT Solvers

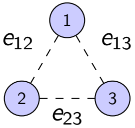
SAT solvers perform well when you have many restrictive constraints—even when those constraints are cumbersome like the triangle constraint and the non-colourability constraint.

SAT solvers use backtracking search and excel at selecting the next variable to branch on that results in a quick conflict.

When they backtrack they learn a short reason for the conflict.

Graphs in SAT

Each edge in a graph is either present or not; say there is an edge between vertices i and j when e_{ij} is true. This gives an adjacency matrix of Boolean variables:



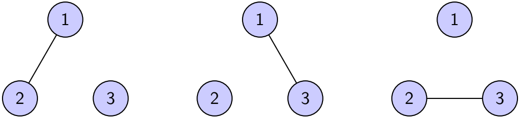
$$\begin{bmatrix} 0 & e_{12} & e_{13} \\ e_{12} & 0 & e_{23} \\ e_{13} & e_{23} & 0 \end{bmatrix}$$

Encoding the squarefree constraint: For each 4-tuple of graph vertices (i, j, k, l), include the constraint

$$\neg(e_{ij} \wedge e_{jk} \wedge e_{kl} \wedge e_{li}).$$

A Problem with SAT

The other KS constraints can also be encoded into SAT, dramatically shrinking the size of the search space—**but** the solver generates many isomorphic copies of the same graph.



$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

In general, an n-vertex graph has $n!$ representations.

SAT Symmetry Breaking

A typical approach is to add "symmetry breaking" constraints that remove as many isomorphic solutions as possible.

For example, lex-order the rows of the adjacency matrix.[8]
However, many distinct isomorphic representations still exist, like

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \text{ and } \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

Instead, we combine SAT with isomorph-free exhaustive generation. This has also been used to certify that projective planes of order 10 do not exist (Lam's problem).[9]

[8] M. Codish, A. Miller, P. Prosser, P. Stuckey. Constraints for symmetry breaking in graph representation. *Constraints*, 2019.

[9] C. Bright, K. Cheung, B. Stevens, I. Kotsireas, V. Ganesh. A SAT-based Resolution of Lam's Problem. *AAAI 2021*.

Orderly Generation

Only “canonical” intermediate objects are recorded. The notion of canonicity is defined so that:

1. Every isomorphism class has exactly one canonical representative.
2. If an object is canonical then it was generated from a canonical object.

Developed independently by Faradžev and Read in 1978.[10,11]

[10] I. Faradžev. Constructive enumeration of combinatorial objects. *Problèmes combinatoires et théorie des graphes*, 1978.

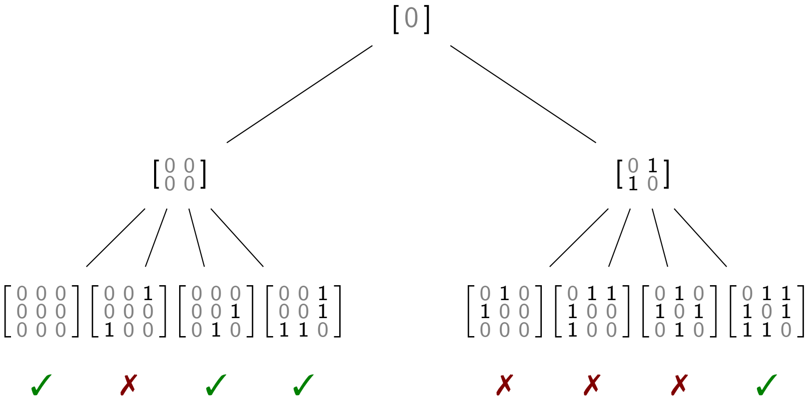
[11] R. Read. Every one a winner or how to avoid isomorphism search when cataloguing combinatorial configurations. *Annals of Discrete Mathematics*, 1978.

Canonicity Example

An adjacency matrix is *canonical* if its "vector representation" is lex-minimal among all matrices in the same isomorphism class.

Adj. matrix	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$
Vector rep.	$[1\ 0\ 0] >_{\mathrm{lex}}$	$[0\ 1\ 0] >_{\mathrm{lex}}$	$[0\ 0\ 1]$
Canonical?	✗	✗	✓

Orderly Generation of Graphs



Canonical testing introduces overhead, but every negative test prunes a large part of the search space (and tests that are negative are usually fast).

Isomorph-free Exhaustive Generation and SAT

I believe there should be more work combining the well-established methods of isomorph-free exhaustive generation with the well-established methods of SAT solving.

There have been a few visionary work along these lines[12,13] and the "SAT modulo symmetry" paradigm incorporates isomorph rejection in a SAT solver.[14]

---

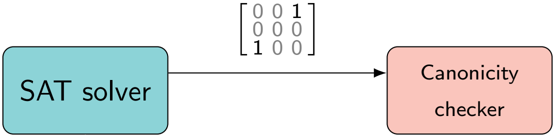
[12] T. Junttila, M. Karppa, P. Kaski, J. Kohonen. An adaptive prefix-assignment technique for symmetry reduction. *Journal of Symbolic Computation*, 2020.

[13] J. Savela, E. Oikarinen, M. Järvisalo. Finding periodic apartments via Boolean satisfiability and orderly generation. *LPAR 2020*.

[14] M. Kirchweger, M. Scheucher, S Szeider. A SAT Attack on Rota's Basis Conjecture. *SAT 2022*.

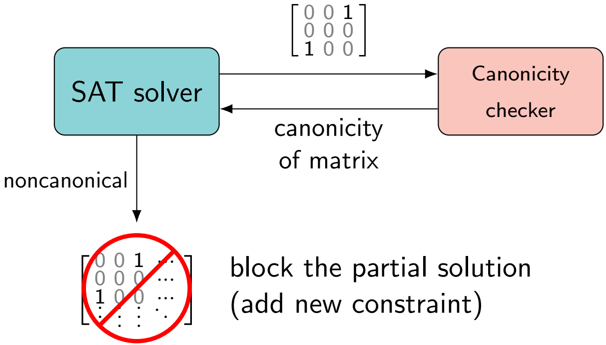
Orderly Generation in SAT

During the search the SAT solver will find partial solutions (complete definitions for the edges in some subgraphs)...



Orderly Generation in SAT

During the search the SAT solver will find partial solutions (complete definitions for the edges in some subgraphs)...



KS Search Results

The time it takes to run an exhaustive search for KS graphs of order n using geng, pure SAT, and SAT + orderly generation:

n	geng	Pure SAT	SAT+O.G.	Speedup
17	25.3 m	8.8 m	0.3 m	$\sim$66x
18	455.6 m	266.6 m	1.7 m	$\sim$209x
19	9,506.4 m	11,705.1 m	8.9 m	$\sim$1,193x
20			83.8 m	
21	$\sim$75 years		$\sim$20 hours	$\sim$32,649x

The order 21 case was solved over 32,000 times faster than the time reported by S. Uijlen and B. Westerbaan (in blue).

The order 22 case was resolved in 19 days. No KS system was found, so a KS system *must have at least 23 directions*.[15]

[15] Z. Li, C. Bright, V. Ganesh. An SC-Square Approach to the Minimum Kochen–Specker Problem. *SC-Square Workshop 2022.*

SAT+CAS Paradigm

The approach can be used for more than just canonicity checking—you can query a computer algebra system (CAS) for mathematical facts that cannot be directly encoded in SAT.[16]

review articles

The science of less-than-brute force.

BY CURTIS BRIGHT, ILIAS KOTSIREAS, AND VIJAY GANESH

When Satisfiability Solving Meets Symbolic Computation

MATHEMATICIANS HAVE LONG been fascinated by objects that exhibit exceptionally nice combinatorial properties. However, it is often difficult to determine whether objects satisfying a given combinatorial property exist. Sometimes, the only feasible method of definitively answering the question of existence is simply to perform a systematic search. A famous example of this is the proof of the *four-color theorem*—the notion that four colors suffice to color the regions of a planar map with adjacent regions colored differently. The theorem has been known to be true since 1977, but every known proof relies on computer calculations in an essential way. Mathematical arguments are used to reduce the search for counterexamples to a finite number of cases, and the cases are then exhaustively checked using a custom-written computer program to rule out any counterexamples.

Independently, computer scientists have made significant progress over the last 50 years on developing general-purpose programs that can automatically solve many kinds of mathematical problems. *Satisfiability solving* and *symbolic computation* are two important branches of computer science that each specialize in solving mathematical problems. Both fields have long histories and have produced impressive tools—satisfiability (SAT) solvers in the former and computer algebra systems (CASs) in the latter. Originally, SAT solvers were designed to solve problems in logic, and CASs were tools to manipulate and simplify algebraic expressions. As we will see, these tools have since found an abundance of new applications outside of these original domains.

Despite their common specialization in solving mathematical problems, the SAT and CAS communities have developed independently of each other. Broadly speaking, the SAT community has focused on effective search methods, while the CAS community has focused on effective mathematical algorithms. Recently, these two communities have started to collaborate in crossover initiatives like the SC-square project. Since the insights of these communities are largely complementary, bringing them together has resulted in new solutions to problems that were out-of-reach of either community separately and has produced advances in problems involving nonlinear real arithmetic, linear integer arithmetic, and Boolean polynomials, to name a few. In this overview, we focus on our own contribution to this ongoing project—a hybrid SAT and CAS system called MathCheck that we have applied to mathematical problems in graph theory, finite geometry, combinatorics, and number theory.

Satisfiability solving. A SAT solver is a program that solves the satisfiability problem from Boolean logic—given a formula in conjunctive normal form, is there an assignment to its variables that makes the expression true? At first glance, SAT solvers seem disconnected from the kinds of problems that most mathematicians and engineers care about. However, stunning progress in applied SAT solving over the last several decades has led to a surprising diversity of applications for SAT solvers—

a www.sc-square.org

b www.uwaterloo.ca/mathcheck

c For more applications of the SAT+CAS paradigm see the overview article appearing in a special issue of the *Journal of Symbolic Computation* devoted to SAT and CAS synergies.

key insights

- Satisfiability (SAT) solving and symbolic computation are fields of computer science with distinguished histories that have developed mostly independently.
- Advances in SAT solving and computer algebra systems (CASs) have led to the development of tools that can solve mathematical search problems significantly larger than ever before—and in a faster, more verifiable way.
- Hybrid "SAT+CAS" systems combine the efficient search and learning routines of SAT solvers with the efficient mathematical algorithms and expressiveness of CASs in order to achieve the best of both worlds.

[16] C. Bright, I. Kotsireas, V. Ganesh. When Satisfiability Checking Meets Symbolic Computation. *Communications of the ACM*, 2022.

A Promising Future

SAT-based isomorph-free generation, and more generally SAT+CAS, can produce exponential speedups over pure SAT or computer algebra.

The approach can be applied to many combinatorial generation problems. If you are interested in using it in your own work I am happy to help.

Thank You!

`curtisbright.com`