SAT Solvers, Isomorph-free Generation, and the Quest for the Minimum Kochen–Specker System

Curtis Bright
University of Windsor

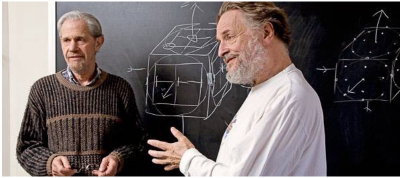
joint work with Zhengyu Li and Vijay Ganesh (Waterloo)

February 7, 2023

UBC Discrete Math Seminar

The Free Will Theorem

In 2006, mathematicians John Conway and Simon Kochen proved the *Free Will Theorem*—if observers have free will then so do quantum particles.[1]

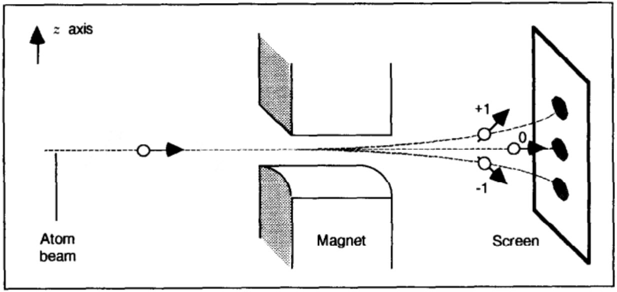


The proof crucially relies on a finite configuration of three dimensional vectors called a Kochen–Specker (KS) system.

[1] J. Conway, S. Kochen. The Free Will Theorem. *Foundations of Physics*, 2006.

The Stern–Gerlach Experiment (1922)

Shoot an atom of orthohelium through a magnetic field:

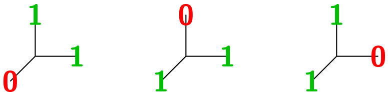


The *spin* of the atom (in this particular direction) is $+1$, -1, or 0.

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**.

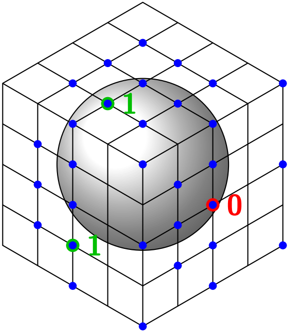


The 101 conspiracy

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

The KS Theorem (1967)

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

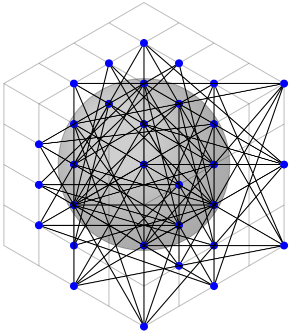


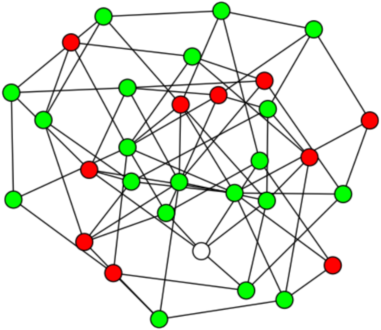
31 vector KS system of Conway and Kochen

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.[2]

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

The search space on 21-vertex graphs is huge, and the computation took 75 CPU years using the best graph enumeration algorithms.[3]

---

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

[3] 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:[4]

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 while the graph is generated vertex-by-vertex.

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

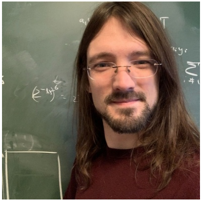
Graph Enumeration

The computer algebra library *nauty* can enumerate all graphs of a given size satisfying properties 1–2.

Property 3 seems difficult to incorporate **during** the generation; instead, it is 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.

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

Example: Is $(x \lor y) \land (\neg x \lor \neg y)$ satisfiable?

SAT to the Rescue

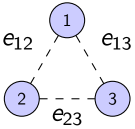
Satisfiability (SAT) solvers take a formula in Boolean logic and try to solve it, i.e., find an assignment that makes it true.

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

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

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:

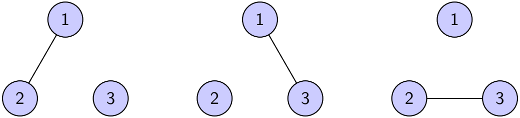


$$\begin{bmatrix} 0 & e_{12} & e_{13} \\ e_{12} & 0 & e_{23} \\ e_{13} & e_{23} & 0 \end{bmatrix}$$

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

SAT in Practice

The SAT approach outperformed nauty's graph enumeration approach—**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.[5]
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).[6]

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

[6] 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.[7,8]

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

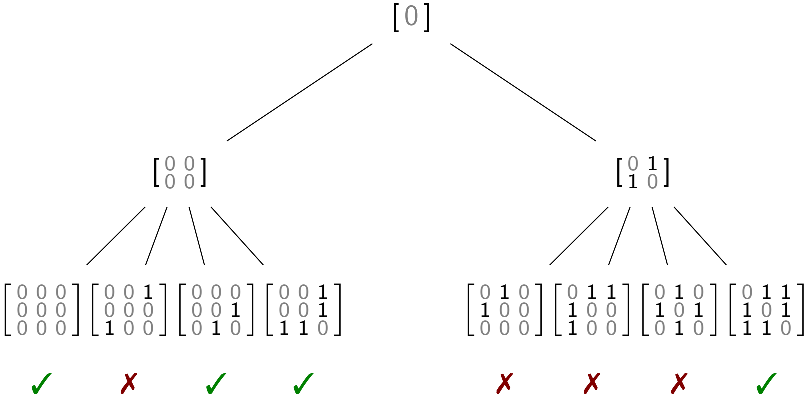
[8] R. Read. Every one a winner or how to avoid isomorphism search when cataloguing combinatorial configurations. *Annals of Discrete Mathematics*, 1978.

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.	$\begin{bmatrix} 1 & 0 & 0 \end{bmatrix} >_{\text{lex}}$	$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix} >_{\text{lex}}$	$\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$
Canonical?	✗	✗	✓

Orderly Generation of Graphs



Canonical testing introduces overhead, but every negative test prunes a large part of the search space.

Orderly Generation in Practice

Each canonical test is independent, making the method easy to parallelize.

Verifying a matrix is *noncanonical* is often fast—it requires finding a single permutation of the vertices giving a lex-smaller matrix.

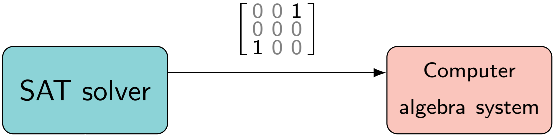
There have been only a few attempts at combining isomorph-free generation and SAT solving.[9,10]

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

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

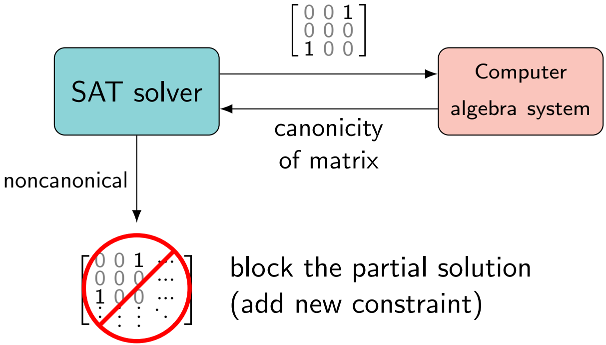
Orderly Generation in SAT

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)...



block the partial solution
(add new constraint)

KS Search Results

The time it takes to run an exhaustive search for KS graphs of a given order *n* using SAT-based orderly generation:

n	Time	Speedup over SAT	Speedup over nauty
17	0.35 m	36.1	83.2
18	2.27 m	31.6	232.7
19	17.43 m	675.9	639.7
20	130.71 m	timeout	timeout
21	1,566.05 m	timeout	timeout

The order 21 case was resolved in under a day on a single desktop, while the previous approach used 300 desktops for three months.

KS Search Results

The time it takes to run an exhaustive search for KS graphs of a given order n using SAT-based orderly generation:

n	Time	Speedup over SAT	Speedup over nauty
17	0.35 m	36.1	83.2
18	2.27 m	31.6	232.7
19	17.43 m	675.9	639.7
20	130.71 m	timeout	timeout
21	1,566.05 m	timeout	timeout

The order 21 case was resolved in under a day on a single desktop, while the previous approach used 300 desktops for three months.

The order 22 case was resolved in 3.4 CPU months. No KS system was found, so a KS system *must have at least 23 directions*.

A Promising Future

SAT-based isomorph-free generation can produce exponential speedups over pure SAT or computer algebra approaches.

The approach can be applied to many combinatorial generation problems. Please reach out if you are interested in using it in your own work.

Thank You!

`curtisbright.com`