SAT and Computer Algebra

Curtis Bright
University of Windsor

June 27, 2023

Dagstuhl Seminar

SAT Encodings and Beyond

Motivation

SAT solvers are great at solving search problems specified by simple constraints (clauses).

Computer algebra systems (CASs) are great at many sophisticated mathematical problems involving little search.

Problems involving *both* sophisticated mathematics and search are good candidates for a SAT+CAS approach.

review articles

DOI:10.1145/3500921

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.[3] 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.[5] 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.[a,2] 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,[19] linear integer arithmetic,[11] and Boolean polynomials,[20] 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[b] that we have applied to mathematical problems in graph theory,[16] finite geometry,[2] combinatorics,[9] and number theory.[1,13]

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[17] has led to a surprising diversity of applications for SAT solvers—

» 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.

[a] www.sc-square.org

[b] www.uwaterloo.ca/mathcheck

[c] For more applications of the SAT+CAS paradigm see the overview article[7] appearing in a special issue of the *Journal of Symbolic Computation* devoted to SAT and CAS synergies.

64 COMMUNICATIONS OF THE ACM | JULY 2022 | VOL. 65 | NO. 7

JULY 2022 | VOL. 65 | NO. 7 | COMMUNICATIONS OF THE ACM 65

Examples of Related Work

There has been a lot of research recently involving SAT and computer algebra or related methods.

A small and incomplete sample:

- Proving the correctness of multiplier circuits (Kaufmann, Biere).
- Finding new algorithms for 3×3 matrix multiplication (Heule, Kauers, Seidl).
- SAT modulo symmetries for generating combinatorial objects in an isomorph-free way (Kirchweger, Peitl, Scheucher, Fazekas, Szeider).
- Making progress on conjectures in geometric group theory (Savela, Oikarinen, Järvisalo).
- Computing directed Ramsey numbers (Neiman, Mackey, Heule).
- Debugging of digital circuits (Mahzoon, Große, Drechsler).

This Talk

I will overview four applications of SAT + CAS that I have worked on from the following publications:

2020 Applying Computer Algebra Systems with SAT Solvers to the Williamson Conjecture, *Journal of Symbolic Computation*.

2021 A SAT-based Resolution of Lam's Problem, *AAAI*.

2022 An SC-Square Approach to the Minimum Kochen–Specker Problem, *7th SC^2 Workshop*.

2023 A Hybrid SAT and Lattice Reduction Approach for Integer Factorization, *8th SC^2 Workshop*.

2020

Williamson Matrices

Applying Computer Algebra Systems with SAT Solvers to the Williamson Conjecture
Journal of Symbolic Computation (Bright, Kotsireas, Ganesh)

Motivation

Consider the problem of sending data over a noisy channel. To be resilient, you and your recipient agree on a set of *codewords* that used to encode the message.

To improve the error-correction properties, you want to make the codewords as “different” as possible.

With *n* binary codewords of length *n* you could try to maximize the bitwise difference between codewords. In the best case, every pair of codewords differ in exactly half their bits.

1	1	1	1
1	1	0	0
1	0	1	0
1	0	0	1

Every pair of codewords differ in 2 bits.

Hadamard Matrices

Represent codewords as $\{\pm 1\}$-vectors and stack them together to form an $n \times n$ matrix:

+	+	+	+
+	+	−	−
+	−	+	−
+	−	−	+

The rows are maximally different when the rows are pairwise orthogonal. This is known as a *Hadamard matrix*.

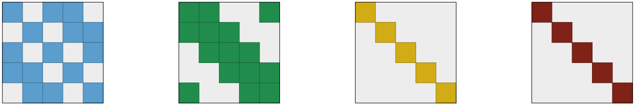
In 1893, Jacques Hadamard proved that n must be a multiple of 4 to be the order of a Hadamard matrix when $n > 2$. But finding a Hadamard matrix of order n can be difficult.

Williamson Matrices

In 1944, John Williamson found a construction for Hadamard matrices of order $4n$ based on searching for symmetric matrices $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D} \in \{\pm 1\}^{n\times n}$ that satisfy the algebraic relationship

$$\boldsymbol{A}^2 + \boldsymbol{B}^2 + \boldsymbol{C}^2 + \boldsymbol{D}^2 = 4n\boldsymbol{I}_{n\times n}.$$

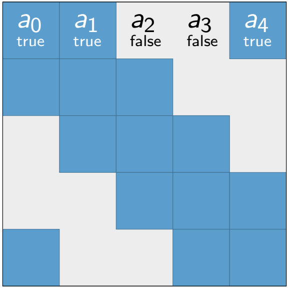
The matrices are also assumed to be *circulant* (each row is a cyclic shift of the previous row).



Williamson matrices of order 5.

Naive SAT encoding

Let the Boolean variable a_i represent the ith entry in the initial row of the matrix $\boldsymbol{A}$ contains a 1.



Using similar variables for $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$, we can express that the off-diagonal entries of $\boldsymbol{A}^2 + \boldsymbol{B}^2 + \boldsymbol{C}^2 + \boldsymbol{D}^2$ are zero using arithmetic circuits (and written in conjunctive normal form).

Computer Algebra Techniques

The naive SAT encoding is not competitive with computer algebraic approaches, because SAT solvers have no conception of the mathematical properties of Williamson matrices.

There are several properties we exploit, but I will just focus on one called the *power spectral density* (PSD). A vector's PSD is the squared magnitudes of its discrete Fourier transform.

That is, if $\boldsymbol{X} := [x_0, \ldots, x_{n-1}]$ and $\omega := \exp(2\pi i/n)$ then

$$\operatorname{PSD}(\boldsymbol{X}, k) := \left| \sum_{j=0}^{n-1} x_j \omega^{jk} \right|^2.$$

PSD Criteria

Theorem. $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$ are the first rows of a set of Williamson matrices of order n if and only if

$$\mathrm{PSD}(\boldsymbol{A}, k) + \mathrm{PSD}(\boldsymbol{B}, k) + \mathrm{PSD}(\boldsymbol{C}, k) + \mathrm{PSD}(\boldsymbol{D}, k) = 4n$$

for $k = 0, \dots, n-1$.

Corollary. PSDs are nonegative, so Williamson matrices satisfy

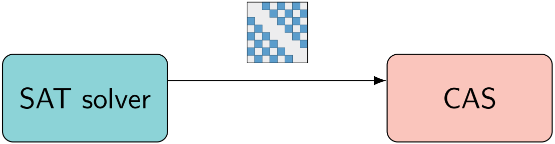
$$\mathrm{PSD}(\boldsymbol{A}, k) \leq 4n \qquad (*)$$

for all $k = 0, \dots, n-1$ (and similarly for $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$).

Encoding $(*)$ into pure SAT would be painful, and no SMT solver that I know of supports the discrete Fourier transform.

SAT+CAS Method

Once the variables encoding a potential Williamson matrix are assigned, the CAS computes its PSD values. . .

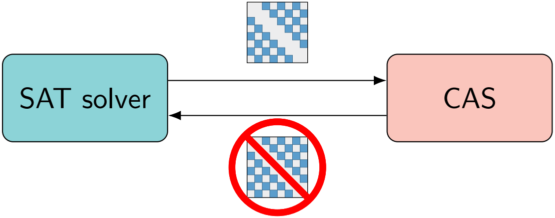


SAT solver

CAS

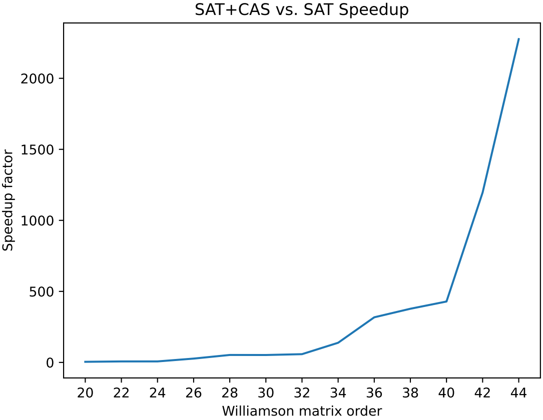
SAT+CAS Method

Once the variables encoding a potential Williamson matrix are assigned, the CAS computes its PSD values...



Since $\mathrm{PSD}(\boldsymbol{A}, 4) = 36 > 4n$, this matrix cannot be Williamson (regardless of $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$), so the CAS produces a conflict clause blocking $\boldsymbol{A}$ and the search continues.

Results



With this approach we exhaustively searched all even orders $n \leq 70$ and found the first known Williamson matrices of order 70 (even orders had previously been searched only up to order 22).

2021

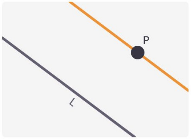
Lam's Problem

A SAT-based Resolution of Lam's Problem
AAAI (Bright, Cheung, Stevens, Kotsireas, Ganesh)

Euclid's Postulates

In 300 B.C., the ancient mathematician Euclid presented five axioms that completely characterized geometry—or so he thought.

His final axiom, the *parallel postulate*, was somewhat more complicated. Could it be derived from the other axioms?



Fifth Axiom

Given a line L and a point P not on L, there is exactly one line through P that is parallel to L.

It took over 2000 years to answer, but the answer is no: there are alternate models of geometry that satisfy Euclid's axioms but **not** the parallel postulate.

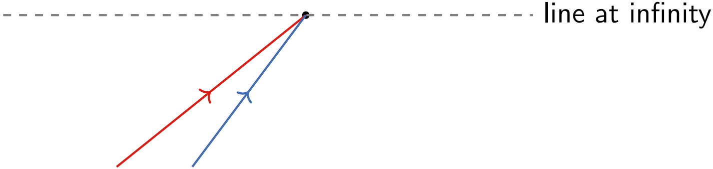
Projective Geometry

For example, consider the following axioms for *projective geometry*:

(1) There is exactly one line between any two points.

(2) Any two lines meet in exactly one point.

The *extended Euclidean plane* is the Euclidean plane with a “line at infinity” added so that formerly parallel lines do now intersect.



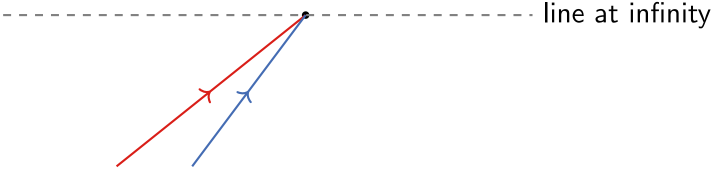
Projective Geometry

For example, consider the following axioms for *projective geometry*:

(1) There is exactly one line between any two points.

(2) Any two lines meet in exactly one point.

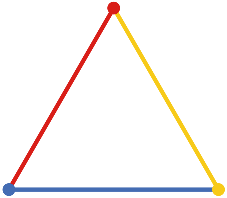
The *extended Euclidean plane* is the Euclidean plane with a “line at infinity” added so that formerly parallel lines do now intersect.

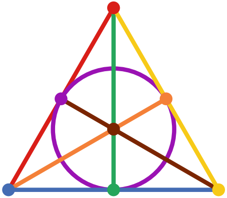


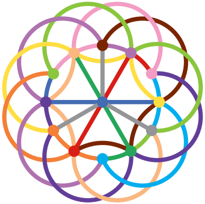
Question: Do finite structures satisfying (1) and (2) exist?

Projective Planes of Orders 1–3

A projective plane is of *order* n if each line contains $n+1$ points.







1	1	0
1	0	1
0	1	1

1	1	0	1	0	0	0
0	1	1	0	1	0	0
0	0	1	1	0	1	0
0	0	0	1	1	0	1
1	0	0	0	1	1	0
0	1	0	0	0	1	1
1	0	1	0	0	0	1

1	0	0	0	1	0	0	0	1	1	0	0	0
0	0	1	1	0	0	0	1	0	1	0	0	0
0	1	0	0	0	1	1	0	0	1	0	0	0
1	0	0	0	0	1	0	1	0	0	1	0	0
0	1	0	1	0	0	0	0	1	0	1	0	0
0	0	1	0	1	0	1	0	0	0	1	0	0
1	0	0	1	0	0	1	0	0	0	0	1	0
0	1	0	0	1	0	0	1	0	0	0	1	0
0	0	1	0	0	1	0	0	1	0	0	1	0
0	0	0	1	1	1	0	0	0	0	0	0	1
0	0	0	0	0	0	1	1	1	0	0	0	1
1	1	1	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	1	1	1	1

Orders of Projective Planes

1	2	3	4	5	6	7	8	9	10
✓	✓	✓	✓	✓	✗	✓	✓	✓	?

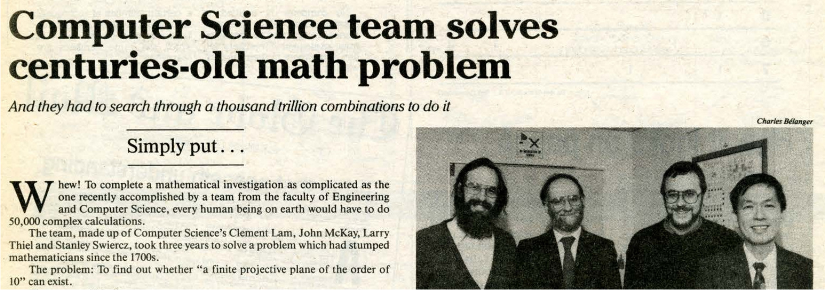
Lam's problem

> *Somehow, this problem has a beauty that fascinates me as well as many other mathematicians.*
>
> Clement Lam

Orders of Projective Planes

1	2	3	4	5	6	7	8	9	10
✓	✓	✓	✓	✓	✗	✓	✓	✓	✗

Lam's problem



Computer Science team solves centuries-old math problem

And they had to search through a thousand trillion combinations to do it

Charles Bélanger

Simply put . . .

Whew! To complete a mathematical investigation as complicated as the one recently accomplished by a team from the faculty of Engineering and Computer Science, every human being on earth would have to do 50,000 complex calculations.

The team, made up of Computer Science's Clement Lam, John McKay, Larry Thiel and Stanley Swiercz, took three years to solve a problem which had stumped mathematicians since the 1700s.

The problem: To find out whether "a finite projective plane of the order of 10" can exist.

Naive SAT Encoding

A projective plane of order n is equivalent to a quad-free binary matrix with $n+1$ ones in each row and column. A *quad* is a rectangle with 1s in the corners:

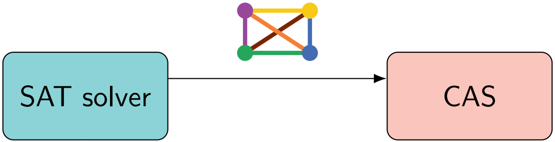
1	0	0	1
0	0	0	0
1	0	0	1

These constraints can be encoded in Boolean logic relatively straightforwardly, but this is not sufficient.

For example, there are a huge number of symmetries in the search (e.g., the rows and columns of a projective plane can be permuted).

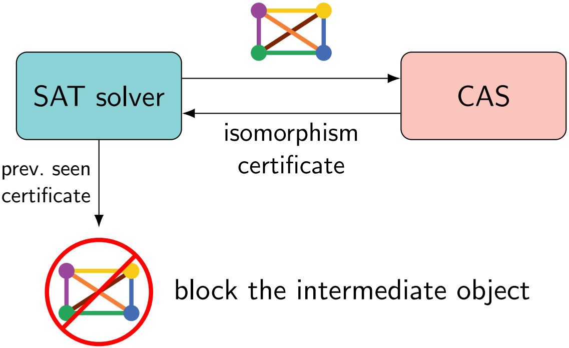
SAT+CAS Method

During the search the SAT solver finds partial solutions by finding complete definitions for the first few lines of the plane. . .



SAT+CAS Method

During the search the SAT solver finds partial solutions by finding complete definitions for the first few lines of the plane. . .



Results

Using theorems from coding theory, Lam's problem can be split into three cases to be solved separately.

Each case involves generating all possibilities for part of the incidence matrix (e.g., all possibilities for its first 19 columns). A SAT+CAS search does this 150 times faster than pure SAT.

Case	**SAT-based**	**CAS-based**	**SAT+CAS**
1	5 minutes	3 minutes	0.1 minutes
2	–	16,000 hours	30 hours
3	–	20,000 hours	16,000 hours

Total running times for solving each case.

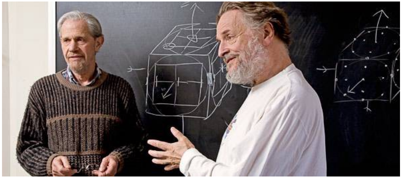
2022

Kochen–Specker Systems

An SC-Square Approach to the Minimum Kochen–Specker Problem
SC-Square Workshop; preprint at arXiv:2306.13319 (Li, Bright, Ganesh)

The Free Will Theorem

Conway and Kochen proved the *Free Will Theorem* in 2006—if humans have have free will then so do quantum particles.[1]

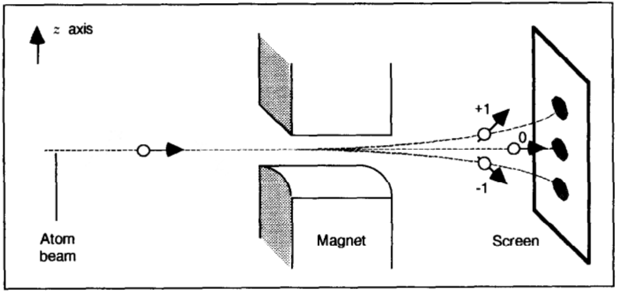


Their proof relies on a finite set of vectors called a Kochen–Specker (KS) system.

[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:

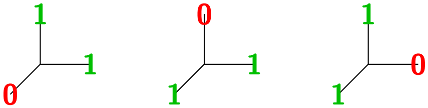


The *spin* of the atom (in this direction) is $+1$, -1, or 0.

Image Source: S. Weinert

The SPIN Axiom

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

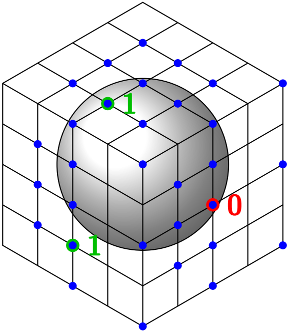


The 101 "conspiracy"

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.



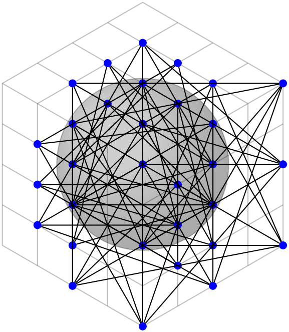
31 vector KS system of Conway and Kochen

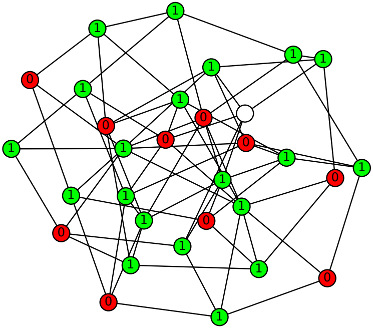
The atom *cannot* have a predetermined spin in every direction!

Image Source: S. Uijlen, B. Westerbaan

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).

Are 31 Vectors Minimal in 3D?

It was known that at least 22 vectors are required.[2] The computation took 75 CPU years using the graph enumeration library nauty.

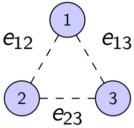
In 2021, I started working with the undergraduate student Zhengyu Li on improving the lower bound.

My intuition was that a SAT+CAS solver would be more effective than nauty alone (which cannot exploit all the known conditions a KS graph must satisfy).

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

SAT Encoding

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:



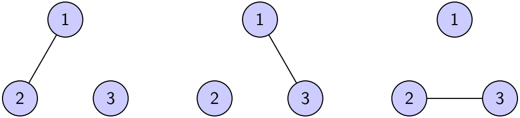
$$\begin{bmatrix} 0 & e_{12} & e_{13} \\ e_{12} & 0 & e_{23} \\ e_{13} & e_{23} & 0 \end{bmatrix}$$

KS graphs must be squarefree, so for each 4-tuple of graph vertices (i, j, k, l), include the constraint

$$\neg e_{ij} \vee \neg e_{jk} \vee \neg e_{kl} \vee \neg e_{li}.$$

Symmetries

Other KS constraints can also be encoded into SAT with some cleverness, **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.[3] However, many distinct isomorphic representations still exist, like

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

Thus, we combine SAT with isomorph-free exhaustive generation.

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

Orderly Generation

Only "canonical" intermediate objects are recorded. The notion of canonicity is defined so that every isomorphism class has exactly one canonical representative.

Developed independently by Faradžev and Read in 1978.[4,5]

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

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

Definition of Canonicity

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

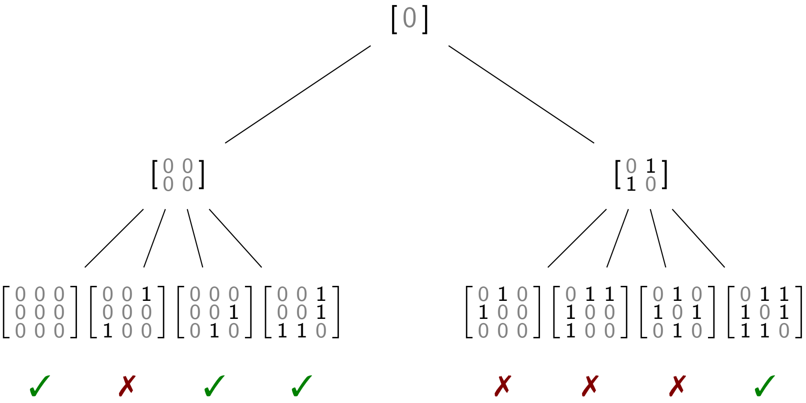
Adj. matrix $\begin{bmatrix}0&1&0\\1&0&0\\0&0&0\end{bmatrix}$ $\begin{bmatrix}0&0&1\\0&0&0\\1&0&0\end{bmatrix}$ $\begin{bmatrix}0&0&0\\0&0&1\\0&1&0\end{bmatrix}$

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? ✗ ✗ ✓

Note that a noncanonical matrix *never becomes canonical* after appending a row and column.

Orderly Generation of Graphs



$$\begin{bmatrix}0\end{bmatrix}$$

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

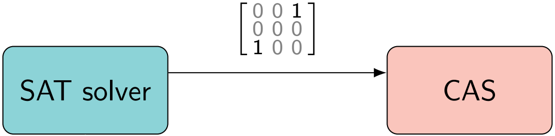
$$\begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix} \begin{bmatrix}0&0&1\\0&0&0\\1&0&0\end{bmatrix} \begin{bmatrix}0&0&0\\0&0&1\\0&1&0\end{bmatrix} \begin{bmatrix}0&0&1\\0&0&1\\1&1&0\end{bmatrix} \qquad \begin{bmatrix}0&1&0\\1&0&0\\0&0&0\end{bmatrix} \begin{bmatrix}0&1&1\\1&0&0\\1&0&0\end{bmatrix} \begin{bmatrix}0&1&0\\1&0&1\\0&1&0\end{bmatrix}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

✓ ✗ ✓ ✓ ✗ ✗ ✗ ✓

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

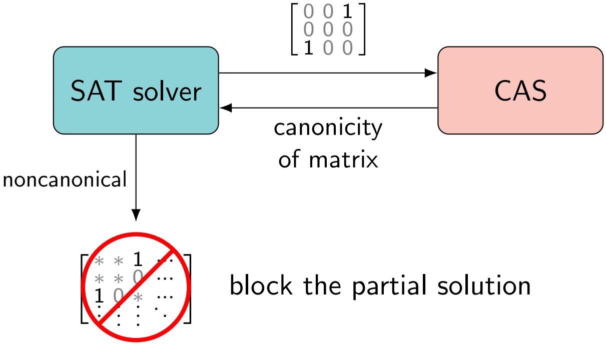
Orderly Generation in SAT

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

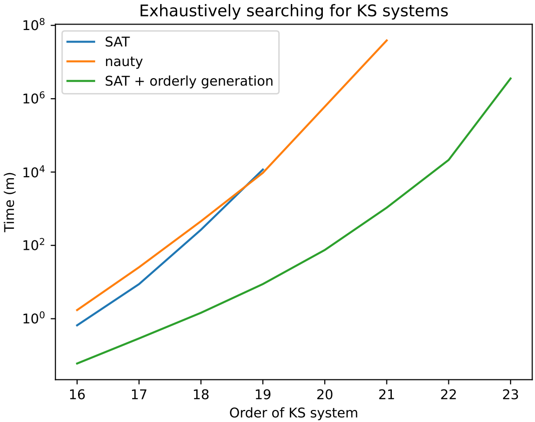

$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$
SAT solver
CAS

Orderly Generation in SAT

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


$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$
SAT solver
CAS
canonicity of matrix
noncanonical
$\begin{bmatrix} * & * & 1 & \cdots \\ * & * & 0 & \cdots \\ 1 & 0 & * & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$
block the partial solution

KS Search Results



No KS system was found, so it **must have at least 24 directions**.
An SMS solver found the same result.[6]

[6] M. Kirchweger, T. Peitl, S. Szeider. Co-Certificate Learning with SAT Modulo Symmetries. *To appear at IJCAI 2023.*

Proof Certificates

The CAS-generated clauses are tagged as "trusted" in the DRAT proof and I modified DRAT-trim to skip the verification of such clauses (originally used while working on Lam's problem[7]).

A separate script verifies the tagged clauses by applying a CAS-derived permutation to the blocked adjacency matrix to verify the blocked matrix is noncanonical.

The uncompressed DRAT proofs up to order 22 were 1.9 TiB.

[7] C. Bright, K. Cheung, B. Stevens, I. Kotsireas, V. Ganesh. Nonexistence Certificates for Ovals in a Projective Plane of Order Ten, *IWOCA 2020*.

2023

Integer Factorization

A Hybrid SAT and Lattice Reduction Approach for Integer Factorization
SC-Square Workshop (Ajani, Bright)

Rivest–Shamir–Adleman Cryptosystem

The commonly-used cryptosystem RSA relies on the difficulty of factoring large integers into primes.

In practice, the RSA algorithm involves a semiprime $N = p \cdot q$ for two randomly chosen primes p and q of the same bitlength.

The best known attack on RSA involves factoring N, but no efficient factoring algorithms are known (unless you use quantum computation).

Reduction to SAT

Multiplication circuits can be converted to SAT straightforwardly by operating directly on the bit-representation of the integers.

> *It's somewhat mind-boggling to realize that numbers can be factored without using any number theory! No greatest common divisors, no applications of Fermat's theorems, etc., are anywhere in sight.*
>
> Donald Knuth, TAOCP 4B

Computer algebra dramatically outperforms SAT on this problem.

Side-channel Attacks

Due to the importance of the factorization problem, cryptographers have studied many ways of attacking it.

One method is via *side-channel attacks* where random bits of the factors of N are leaked.

The SAT approach has the advantage that it is easy to incorporate extra information leaked about the factors of N: just add an appropriate unit clause for each leaked bit.

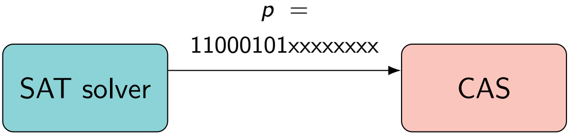
Coppersmith's Method

Most CAS-based methods cannot easily take advantage of leaked bits, but Don Coppersmith showed that if the top or bottom 50% of the bits of p or q are known then N can be efficiently factored via lattice reduction.

However, Coppersmith's method cannot be used if the leaked bits are randomly distributed.

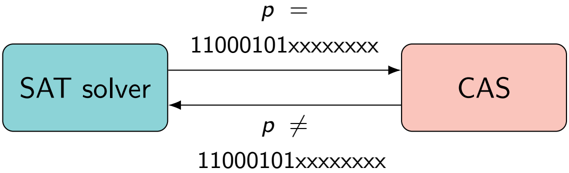
SAT+CAS Method

When the top-half of the bits of p (or q) are assigned, pass them to Coppersmith's method. . .



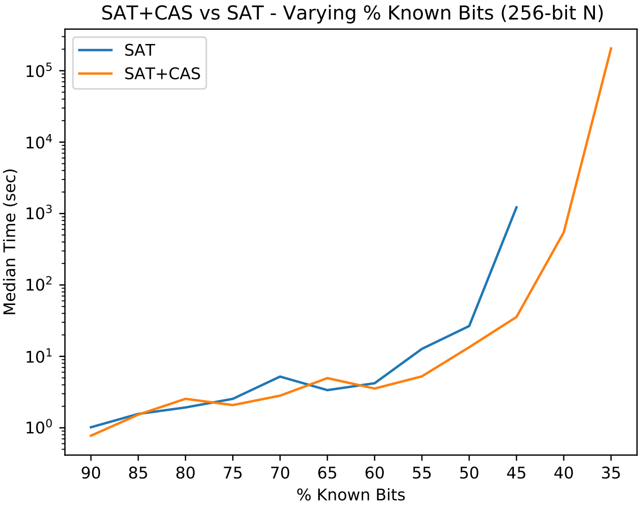
SAT+CAS Method

When the top-half of the bits of p (or q) are assigned, pass them to Coppersmith's method...



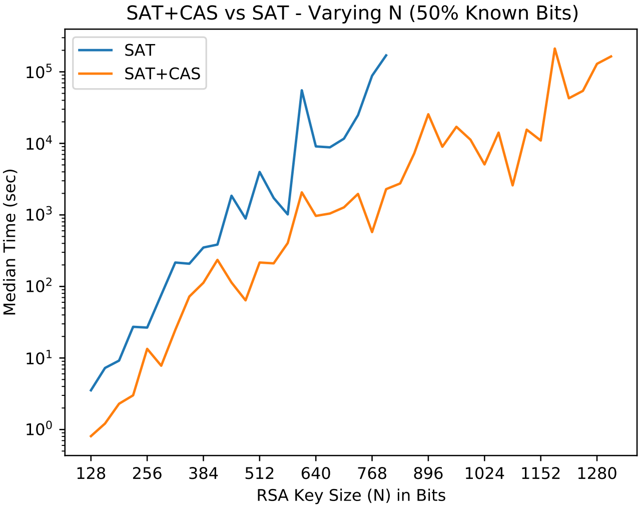
If Coppersmith's method succeeds, then N is factored. If not, learn a clause forcing the high bits of p to change.

Varying the Proportion of Leaked Bits



[The running times originally presented were inaccurate. The plot shown here is a corrected version.]

Varying the Bitlength of N



[The running times originally presented were inaccurate. The plot shown here is a corrected version.]

A Promising Future!

I regularly see SAT+CAS solvers providing exponential speedups over pure SAT and pure computer algebra approaches.

The approach is very flexible and can be applied in many problems requiring search and advanced mathematics. There is a lot of low-hanging fruit and much more remains to be done!

A Promising Future!

I regularly see SAT+CAS solvers providing exponential speedups over pure SAT and pure computer algebra approaches.

The approach is very flexible and can be applied in many problems requiring search and advanced mathematics. There is a lot of low-hanging fruit and much more remains to be done!

Thank You!

`curtisbright.com`