SAT Solving with Computer Algebra

for Fast, Verified Mathematical Search

Curtis Bright
Carleton University

School of Computer Science Seminar
University of Windsor
March 12, 2020

SAT:

Boolean satisfiability problem

SAT:

Boolean satisfiability problem

SAT solvers: Clever brute force

Effectiveness of SAT solvers

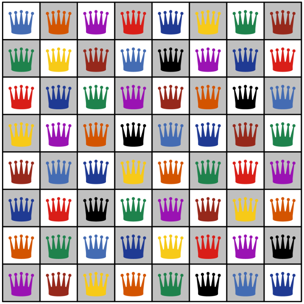
Surprisingly, many problems that have nothing to do with logic can be effectively solved by translating them into Boolean logic and using a SAT solver.

Effectiveness of SAT solvers

Surprisingly, many problems that have nothing to do with logic can be effectively solved by translating them into Boolean logic and using a SAT solver.

Examples

- Discrete optimization
- Hardware and software verification
- Proving/disproving conjectures (my specialty)



Limitations of SAT solvers

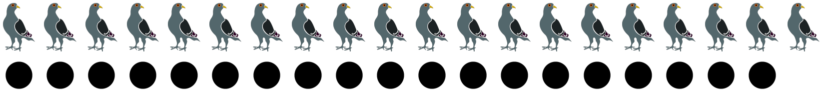
SAT solvers lack mathematical understanding beyond the most basic logical inferences and will fail on some trivial tiny problems.

Limitations of SAT solvers

SAT solvers lack mathematical understanding beyond the most basic logical inferences and will fail on some trivial tiny problems.

Example

Have a SAT solver try to find a way to put 20 pigeons into 19 holes such that no hole contains more than one pigeon...



CAS:

Computer algebra system

CAS:

Computer algebra system

Algorithmic mathematical computing

Effectiveness of CAS

Computer algebra systems can perform calculations and manipulate expressions from many branches of mathematics.

Example

What is the value of

$$\sum_{n=1}^{\infty} \frac{1}{n^2}\,?$$

Effectiveness of CAS

Computer algebra systems can perform calculations and manipulate expressions from many branches of mathematics.

Example

What is the value of

$$\sum_{n=1}^{\infty} \frac{1}{n^2}\,?$$

Maple returns $\pi^2/6$.

Effectiveness of CAS

Computer algebra systems can perform calculations and manipulate expressions from many branches of mathematics.

Example

What is the value of

$$\sum_{n=1}^{\infty} \frac{1}{n^2}\,?$$

Maple returns $\pi^2/6$ ... *not* 1.64493406685.

Limitations

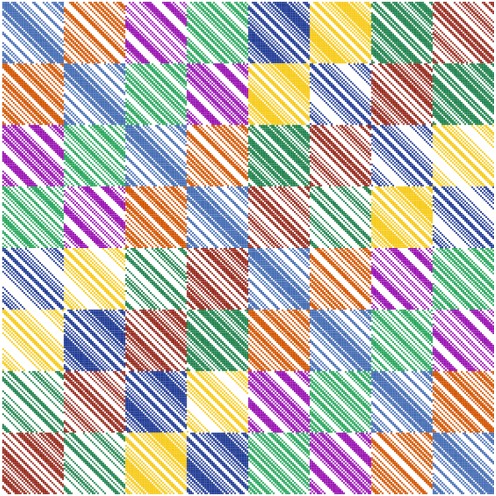
CASs are not optimized to do large searches (in an exponential-sized space).

SAT + CAS

Search + Math

MathCheck: A SAT+CAS system

MathCheck has found over 100,000 combinatorial matrices like this $\{\pm 1\}$-matrix of order 280 with pairwise orthogonal rows:



MathCheck results (see uwaterloo.ca/mathcheck)

Discrete Geometry:
Fastest verification of cases in Lam's problem (1800s).

Graph Theory:
Current best result in the Ruskey–Savage conjecture (1993).
Current best result in the Norin conjecture (2008).

Combinatorics:
Found the smallest counterexample of the Williamson conjecture (1944).
Found three new counterexamples of the good matrix conjecture (1971).
Current best result in the best matrix conjecture (2001).

Number Theory:
Verified a conjecture of Craigen, Holzmann, and Kharaghani (2002).

Lam's Problem

History

Since 300 BC, mathematicians have tried to derive Euclid's "parallel postulate" from his first four postulates for geometry.

History

Since 300 BC, mathematicians have tried to derive Euclid's "parallel postulate" from his first four postulates for geometry.

The existence of projective planes (1800s) shows this is impossible!

Projective planes

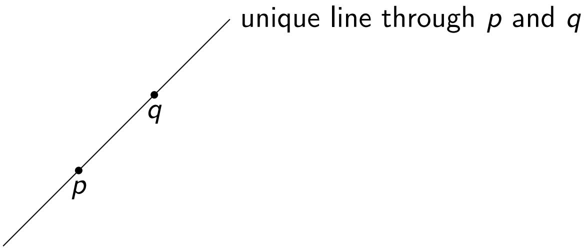
A *projective plane* is a collection of points and lines and a relation between points and lines such that:

1. There is a unique line through any two points.
2. Any two lines intersect at a unique point.

Projective planes

A *projective plane* is a collection of points and lines and a relation between points and lines such that:

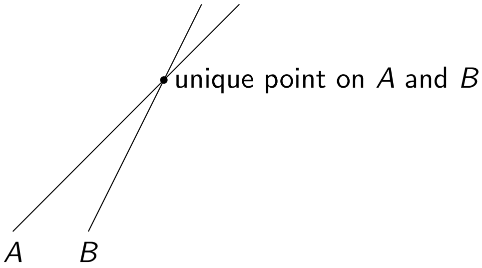
1. There is a unique line through any two points.
2. Any two lines intersect at a unique point.



Projective planes

A *projective plane* is a collection of points and lines and a relation between points and lines such that:

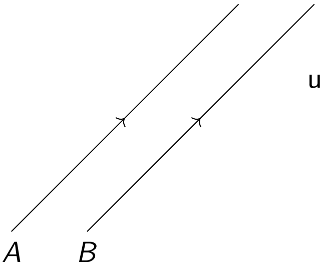
1. There is a unique line through any two points.
2. Any two lines intersect at a unique point.



Projective planes

A *projective plane* is a collection of points and lines and a relation between points and lines such that:

1. There is a unique line through any two points.
2. Any two lines intersect at a unique point.

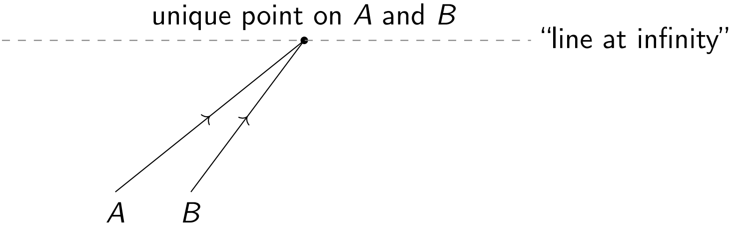


unique point on A and B?

Projective planes

A *projective plane* is a collection of points and lines and a relation between points and lines such that:

1. There is a unique line through any two points.
2. Any two lines intersect at a unique point.



Finite projective planes

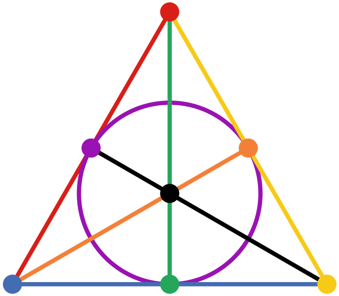
Must a projective plane have an infinite number of points?

Finite projective planes

Must a projective plane have an infinite number of points?

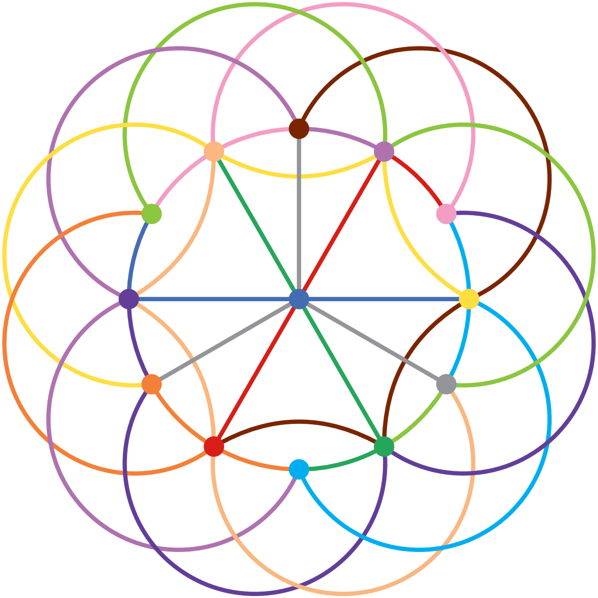
If not, it must have exactly $n^2 + n + 1$ points for some integer n (called the *order* of the plane).

Projective plane of order 2



point 1	line 1
point 2	line 2
point 3	line 3
point 4	line 4
point 5	line 5
point 6	line 6
point 7	line 7

Projective plane of order 3



Projective planes of small orders

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

Projective planes of small orders

2	3	4	5	6	7	8	9	10
✓	✓	✓	✓	✗	✓	✓	✓	?

Lam's Problem

Projective planes of small orders

2	3	4	5	6	7	8	9	10
✓	✓	✓	✓	✗	✓	✓	✓	✗

Supercomputer Search (1973–1989)

Projective plane of order 2: Incidence matrix

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

Boolean matrix of size 7×7 where (i, j)th entry is 1 exactly when the ith line is incident with the jth point.

SAT encoding: false $\equiv 0$, true $\equiv 1$

Lam's problem: First case

The first case of Lam's problem was solved in 1973 and has been verified by at least four independent implementations on modern desktops:

Authors	**Year**	**Language**	**Time**
Roy	2005	C	78 min
Casiello, Indaco, and Nagy	2010	GAP	3.3 min
Clarkson and Whitesides	2014	C	27 sec
Perrott	2016	Mathematica	55 min

Lam's problem: First case

The first case of Lam's problem was solved in 1973 and has been verified by at least four independent implementations on modern desktops:

Authors	Year	Language	Time
Roy	2005	C	78 min
Casiello, Indaco, and Nagy	2010	GAP	3.3 min
Clarkson and Whitesides	2014	C	27 sec
Perrott	2016	Mathematica	55 min
Bright et al.	2019	SAT	6.3 min

Lam's problem: First case

The first case of Lam's problem was solved in 1973 and has been verified by at least four independent implementations on modern desktops:

Authors	**Year**	**Language**	**Time**
Roy	2005	C	78 min
Casiello, Indaco, and Nagy	2010	GAP	3.3 min
Clarkson and Whitesides	2014	C	27 sec
Perrott	2016	Mathematica	55 min
Bright et al.	2019	SAT	6.3 min
Bright et al.	**2019**	**SAT+CAS**	**6.8 sec**

Lam's problem: Second case

The second case was initiated in 1974 and not entirely searched until 1986. I am only aware of a single verification on a modern desktop prior to our work:

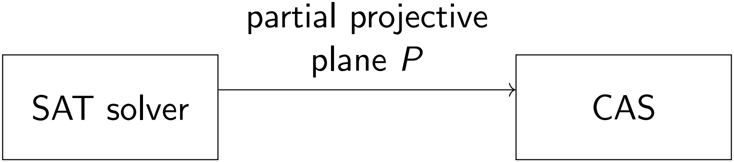
Authors	Year	Language	Time
Roy	2011	C	16,000 hours

Lam's problem: Second case

The second case was initiated in 1974 and not entirely searched until 1986. I am only aware of a single verification on a modern desktop prior to our work:

Authors	**Year**	**Language**	**Time**
Roy	2011	C	16,000 hours
Bright et al.	**2020**	**SAT+CAS**	**30 hours**

Learning method

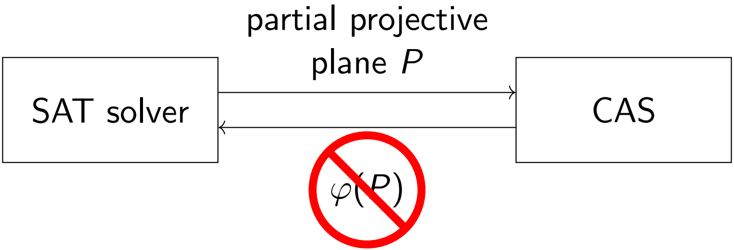


Learning method

$$\boxed{\text{SAT solver}} \xrightarrow{\text{partial projective plane } P} \boxed{\text{CAS}}$$

The CAS computes a nontrivial symmetry φ of the plane...

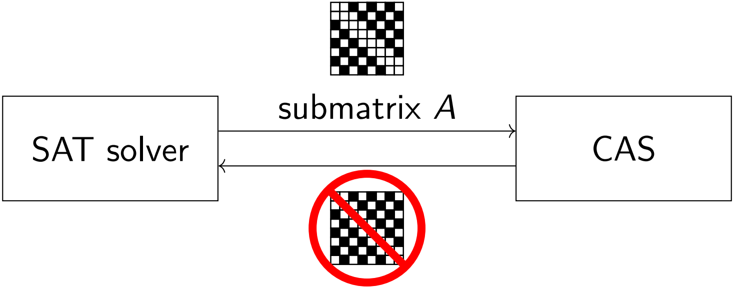
Learning method



The CAS computes a nontrivial symmetry φ of the plane...

...and a symmetry "blocking clause" is learned.

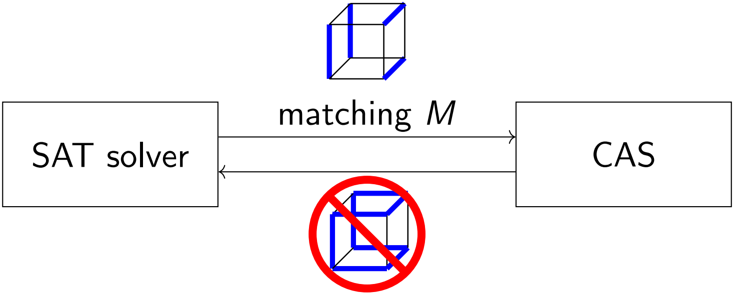
Learning method: Hadamard matrices



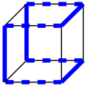
The CAS computes the largest magnitude in the discrete Fourier transform of A. If it is too large...

...a "conflict clause" is learned.

Learning method: Ruskey–Savage conjecture



The CAS tries to extend the matching M to a Hamiltonian cycle...



...and if successful a conflict clause is learned.

Lam's problem: Third case

The final case was solved in 1989 by Lam et al. using 26 months on a supermini computer and 3 months on a supercomputer.

It was verified by Roy in 2011 using 26 months on a desktop.

Lam's problem: Third case

The final case was solved in 1989 by Lam et al. using 26 months on a supermini computer and 3 months on a supercomputer.

It was verified by Roy in 2011 using 26 months on a desktop.

Ultimate goal: Complete a SAT+CAS verification of this case and search for larger projective planes—little is known about projective planes of orders 11 and 12.

Verifiability

All previous searches were unverifiable. They require trusting:

- The hardware, compiler, and operating system used.
- The search code used.
- That the search was successfully run to completion.

Verifiability

All previous searches were unverifiable. They require trusting:

- The hardware, compiler, and operating system used.
- The search code used.
- That the search was successfully run to completion.

This is a lot to trust. *Our searches found bugs in prior searches.*

SAT certification

In contrast, SAT solvers provide unsatisfiability certificates.

Our searches reduce necessary trust to the SAT encoding, the CAS-derived clauses, and a small trusted proof verifier.

SAT certification

In contrast, SAT solvers provide unsatisfiability certificates.

Our searches reduce necessary trust to the SAT encoding, the CAS-derived clauses, and a small trusted proof verifier.

Ultimate goal: Generate a complete nonexistence proof entirely from the projective plane axioms.

More SAT+CAS applications

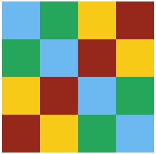
New and emerging areas of application include circuit minimization, verification, cryptography, and program synthesis.

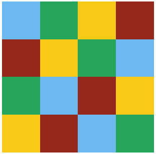
I will outline just two promising applications:

- Mutually orthogonal Latin squares
- The Hadwiger–Nelson problem

Latin squares

An $n \times n$ matrix whose entries contain n symbols is a *Latin square* if each row and column contains exactly one of each symbol.



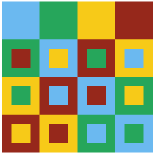


Two Latin squares of order four.

Applications to experimental design, statistics, codes, ...

Orthogonal Latin squares

Two Latin squares are *orthogonal* if all n^2 pairs of entries appear in their superposition.



Pair of orthogonal Latin squares of order four.

Mutually orthogonal Latin squares

A famous open problem (1700s) is to determine how large a set of mutually orthogonal Latin squares can be in order n.

Mutually orthogonal Latin squares

A famous open problem (1700s) is to determine how large a set of mutually orthogonal Latin squares can be in order n.

Euler conjectured if n is of the form $4k+2$ then pairs of orthogonal Latin squares do not exist.

Mutually orthogonal Latin squares

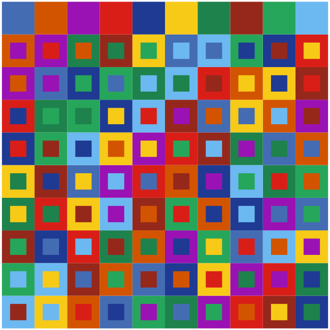
A famous open problem (1700s) is to determine how large a set of mutually orthogonal Latin squares can be in order n.

Euler conjectured if n is of the form $4k + 2$ then pairs of orthogonal Latin squares do not exist.

Mourtos estimated solving the first open case (order ten) would require 273 years using integer and constraint programming.

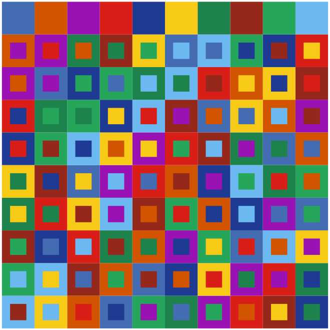
A SAT approach to Latin squares

A SAT approach solves the order six case about 500 times faster than Mourtos' method and is able to find a pair of orthogonal Latin squares of order ten about 20% faster:



A SAT approach to Latin squares

A SAT approach solves the order six case about 500 times faster than Mourtos' method and is able to find a pair of orthogonal Latin squares of order ten about 20% faster:



Ultimate goal: Prove or disprove the conjecture that a triple of mutually orthogonal Latin squares of order ten do not exist.

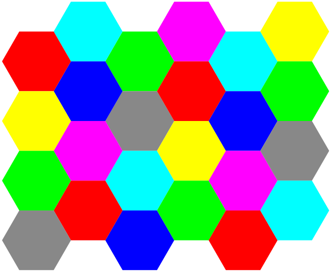
Hadwiger–Nelson problem

How many colours are needed to colour the plane so that no two points separated a distance of 1 are the same colour?

Hadwiger–Nelson problem

How many colours are needed to colour the plane so that no two points separated a distance of 1 are the same colour?

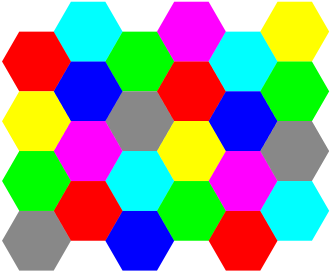
At most 7:



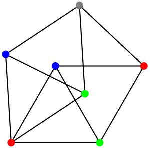
Hadwiger–Nelson problem

How many colours are needed to colour the plane so that no two points separated a distance of 1 are the same colour?

At most 7:



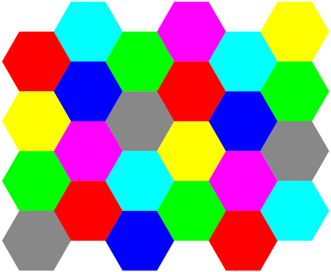
At least 4:



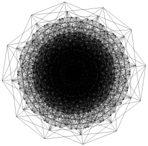
Hadwiger–Nelson problem

How many colours are needed to colour the plane so that no two points separated a distance of 1 are the same colour?

At most 7:



At least 5:



Ultimate goal: Improve these bounds and find the exact answer.

Conclusion

The SAT+CAS paradigm is a new fast way of searching for combinatorial objects—or disproving their existence.

Conclusion

The SAT+CAS paradigm is a new fast way of searching for combinatorial objects—or disproving their existence.

Industry-backed: Maplesoft have already supported SAT+CAS research and are interested in more applications.

Conclusion

The SAT+CAS paradigm is a new fast way of searching for combinatorial objects—or disproving their existence.

Industry-backed: Maplesoft have already supported SAT+CAS research and are interested in more applications.

Wide application: *Many* mathematical problems stand to benefit from faster search tools.

Conclusion

The SAT+CAS paradigm is a new fast way of searching for combinatorial objects—or disproving their existence.

Industry-backed: Maplesoft have already supported SAT+CAS research and are interested in more applications.

Wide application: *Many* mathematical problems stand to benefit from faster search tools.

Bang for your buck: Requires knowledge of SAT and CAS, but generally simpler to write and verify than a special-purpose search.

Thank you and I'm happy to answer any questions.

`curtisbright.com`

Selected References:

Bright, Cheung, Stevens, Roy, Kotsireas, Ganesh. A Nonexistence Certificate for Projective Planes of Order Ten with Weight 15 Codewords. AAECC 2020.

Bright, Cheung, Stevens, Kotsireas, Ganesh. Nonexistence Certificates for Ovals in a Projective Plane of Order Ten. IWOCA 2020.

Bright, Gerhard, Kotsireas, Ganesh. Effective Problem Solving Using SAT Solvers. Maple in Mathematics Education and Research. MC 2019.

Bright, Đoković, Kotsireas, Ganesh. A SAT+CAS Approach to Finding Good Matrices: New Examples and Counterexamples. AAAI 2019.

Bright, Kotsireas, Ganesh. A SAT+CAS Method for Enumerating Williamson Matrices of Even Order. AAAI 2018.

Bright, Kotsireas, Heinle, Ganesh. Enumeration of Complex Golay Pairs via Programmatic SAT. ISSAC 2018.