SAT Solving with Computer Algebra

and its Application to Graph Theory and Geometry

Curtis Bright

Carleton University

Computational Geometry Lab Seminar
February 28, 2020

SAT:

Boolean satisfiability problem

SAT:

Boolean satisfiability problem

SAT solvers: Clever brute force

Effectiveness of SAT solvers

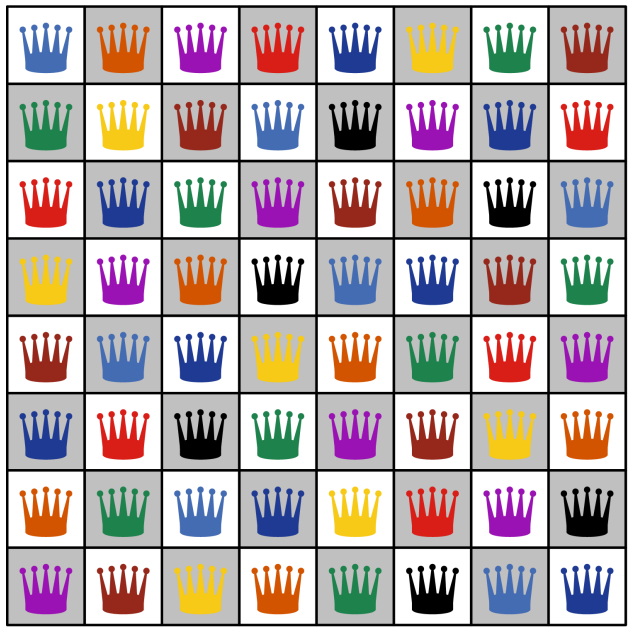
Surprisingly, many problems that have nothing to do with logic can be effectively solved by translating them into Boolean logic and using a SAT solver.

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

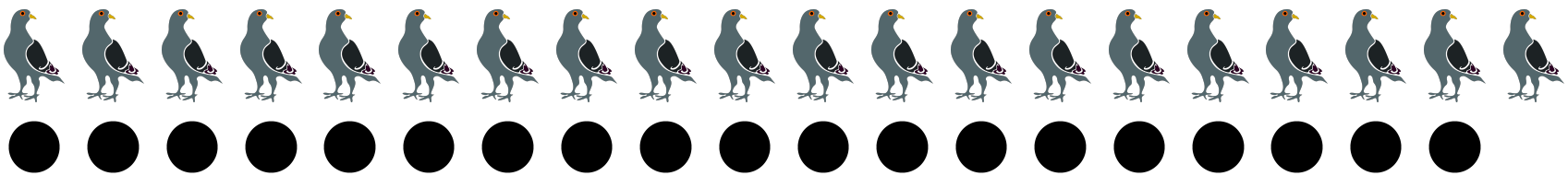
SAT solvers lack mathematical understanding beyond the most basic logical inferences and will fail on some trivial tiny problems.

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 to 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.

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

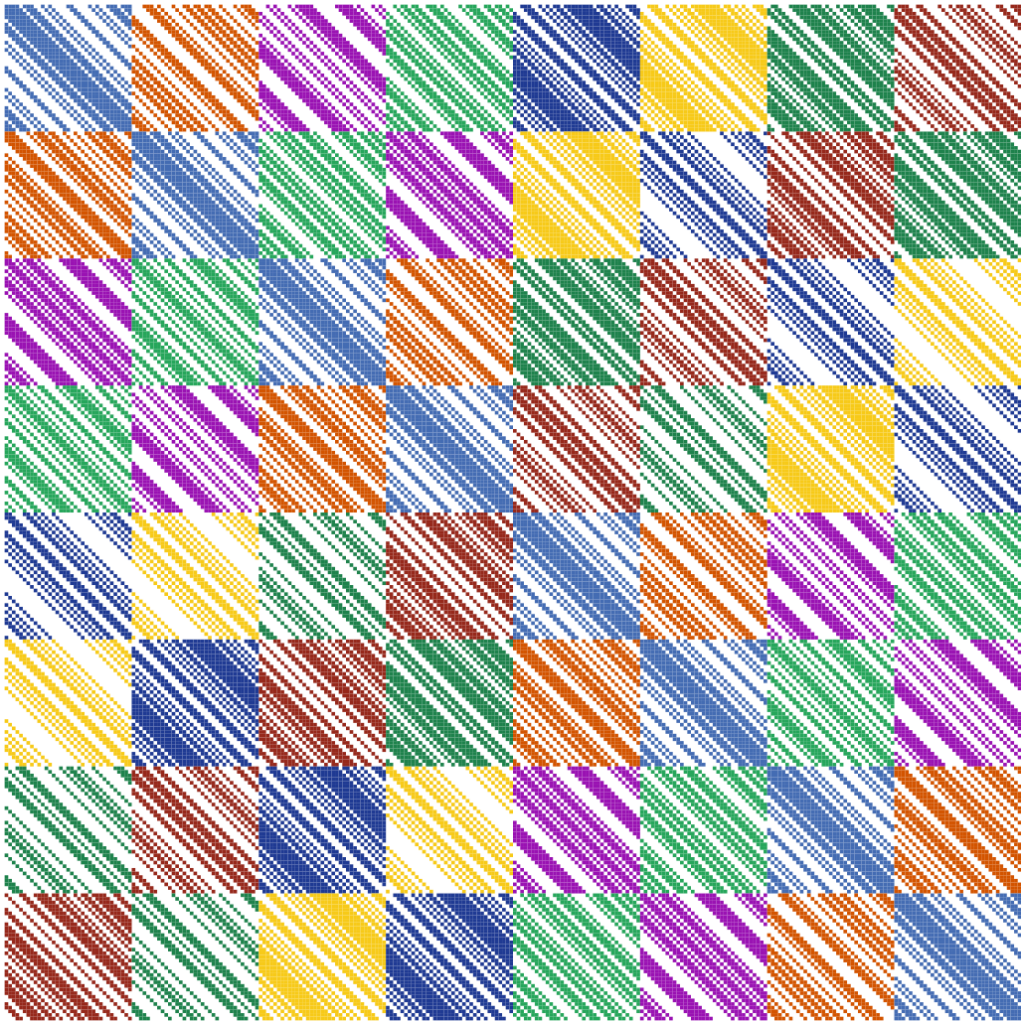
CASs are not optimized to do large searches (in an exponential-sized space).

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)

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

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

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).

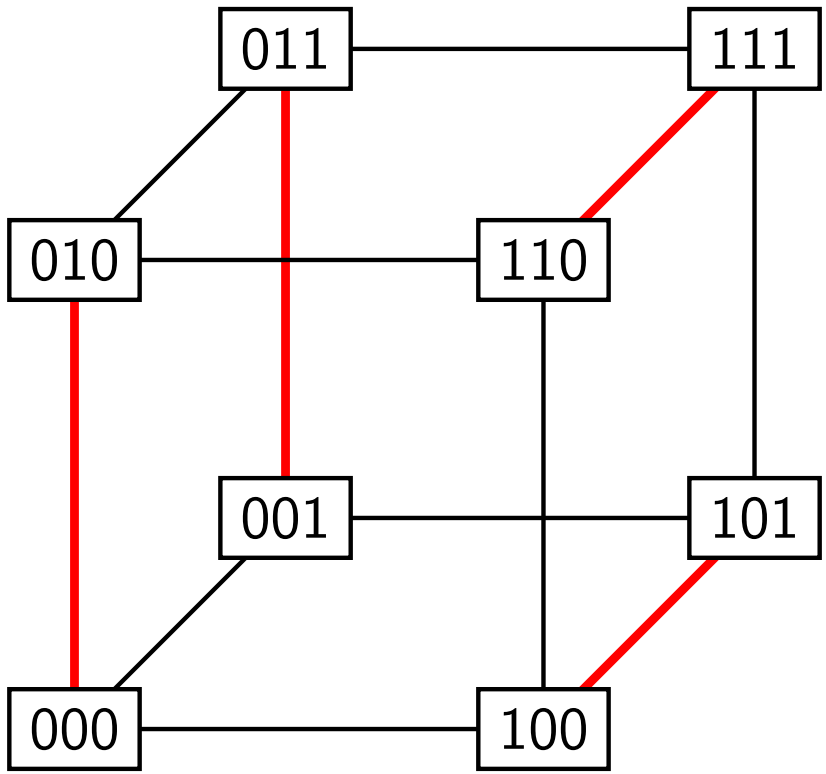
Ruskey–Savage Conjecture

Ruskey–Savage conjecture

Every matching of the hypercube graph of dimension at least two extends to a Hamiltonian cycle of the graph.

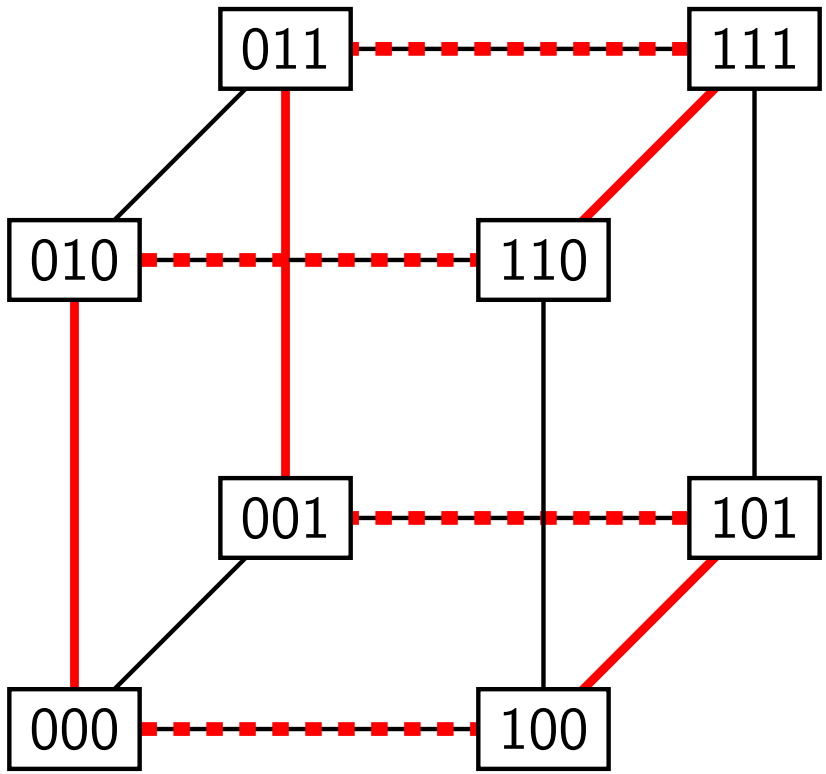
Ruskey–Savage conjecture

Every matching of the hypercube graph of dimension at least two extends to a Hamiltonian cycle of the graph.



Ruskey–Savage conjecture

Every matching of the hypercube graph of dimension at least two extends to a Hamiltonian cycle of the graph.



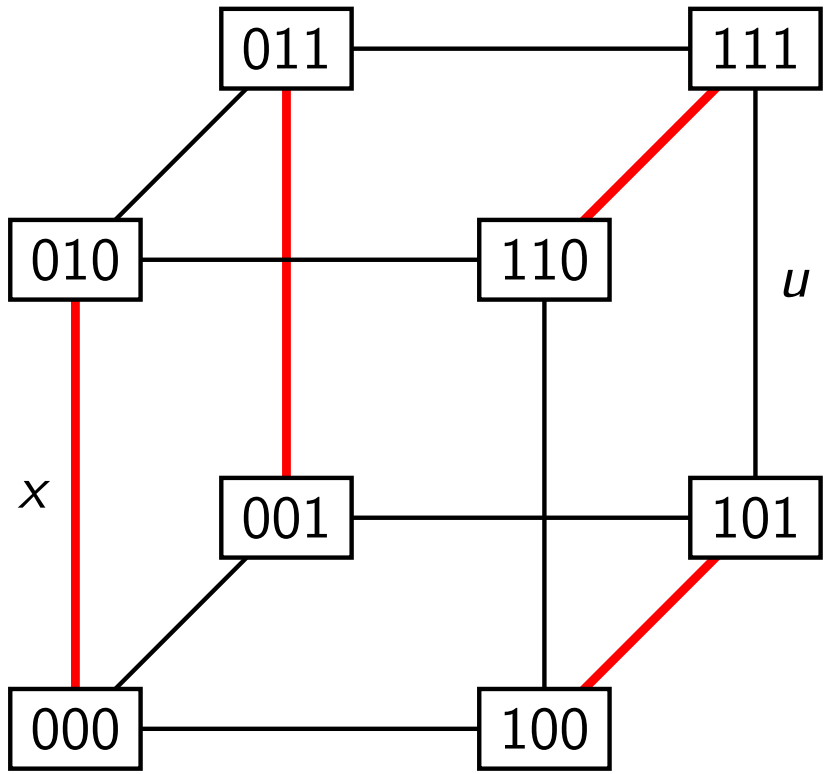
SAT encoding

We use a SAT solver to search for a counterexample.

First, we need a way of encoding the “matching” part.

SAT variables

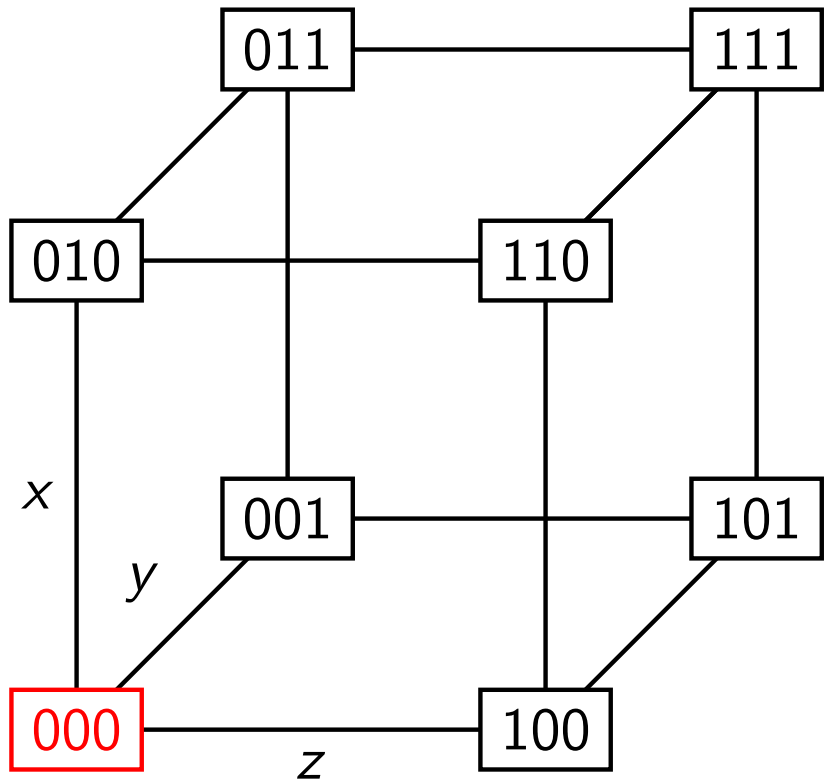
Define a Boolean variable for every edge of the hypercube graph.



In this matching the variable x is true (in the matching) and u is false (outside the matching).

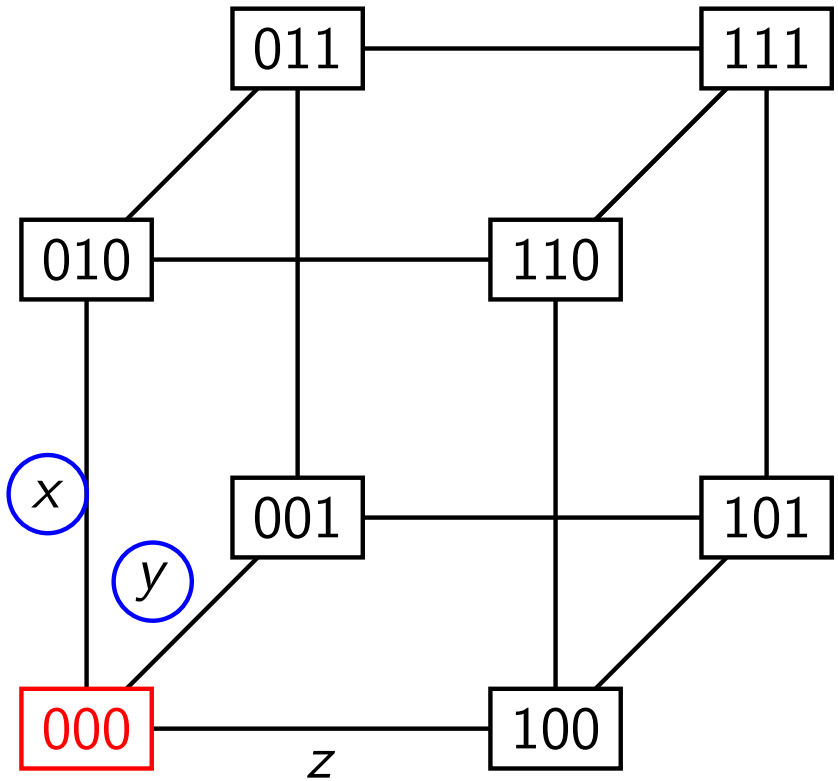
SAT constraints

For every vertex of the graph at most one edge incident to the vertex can be in the matching.



SAT constraints

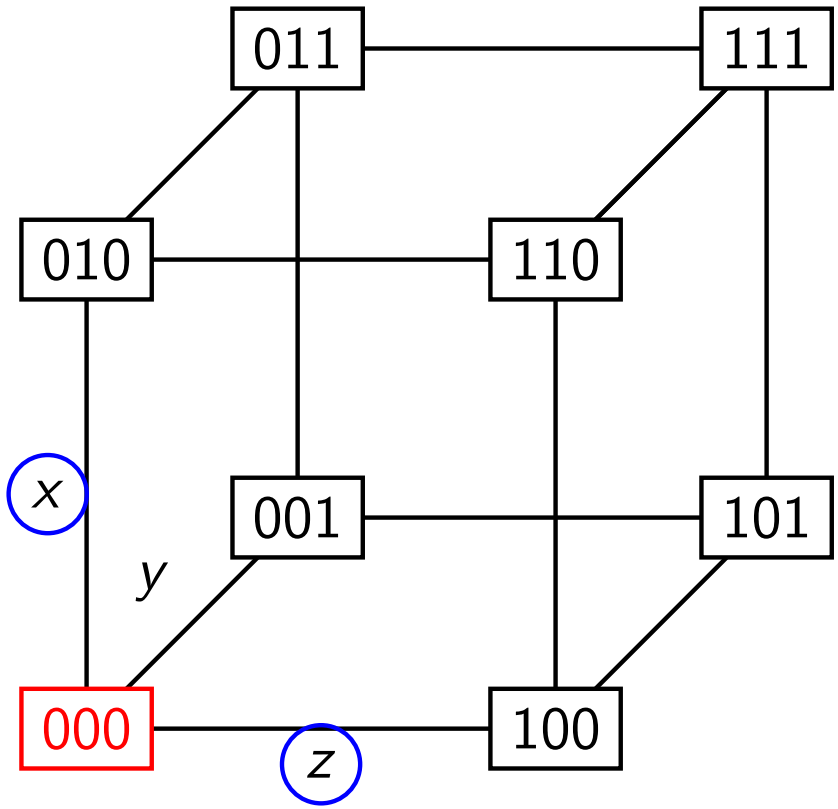
For every vertex of the graph at most one edge incident to the vertex can be in the matching.



In Boolean logic: $\neg x \vee \neg y$

SAT constraints

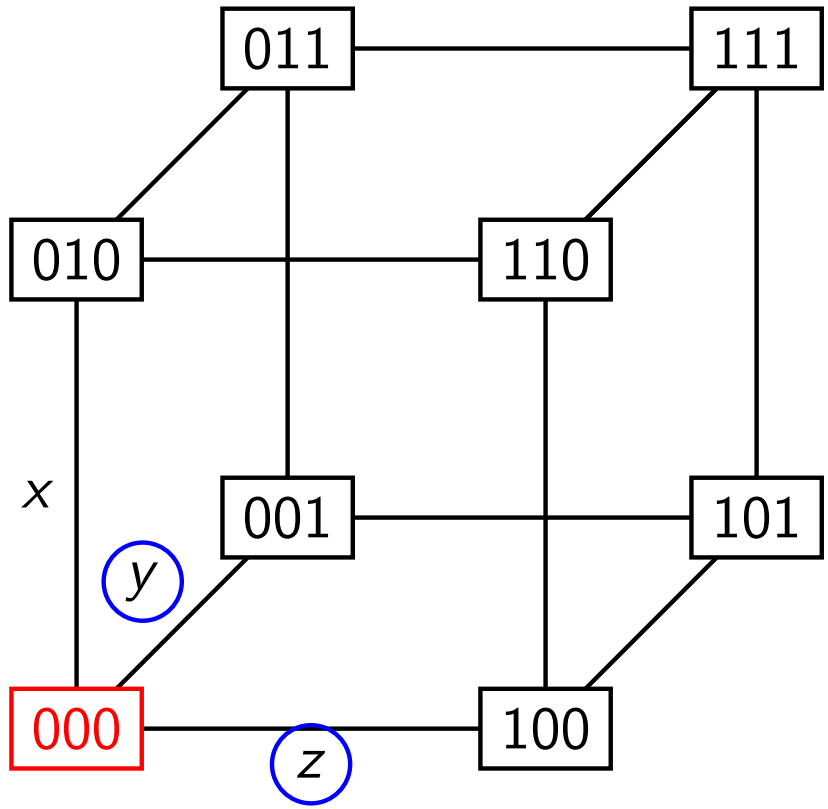
For every vertex of the graph at most one edge incident to the vertex can be in the matching.



In Boolean logic: $\neg x \vee \neg y$, $\neg x \vee \neg z$

SAT constraints

For every vertex of the graph at most one edge incident to the vertex can be in the matching.



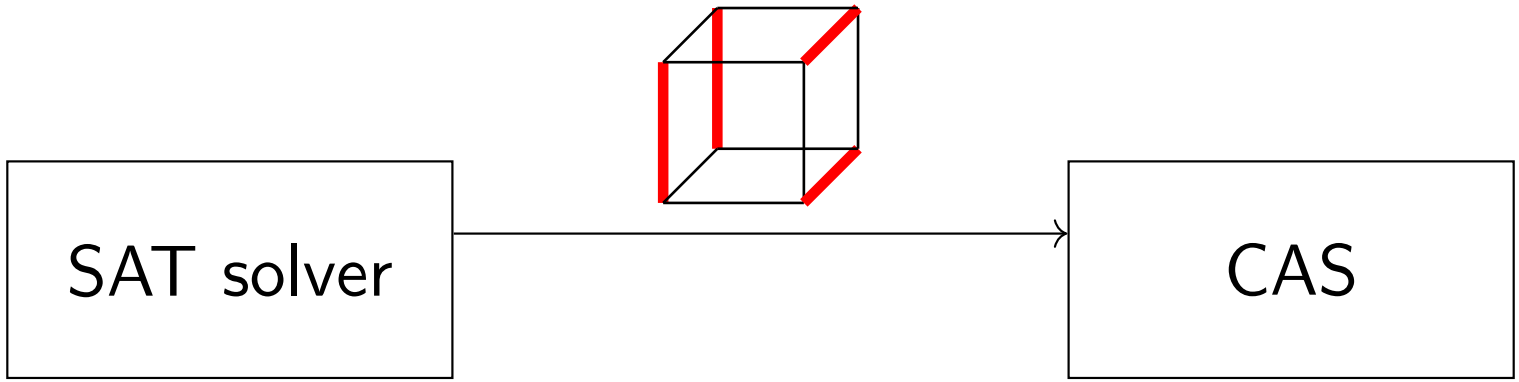
In Boolean logic: $\neg x \vee \neg y$, $\neg x \vee \neg z$, $\neg y \vee \neg z$

Counterexample search

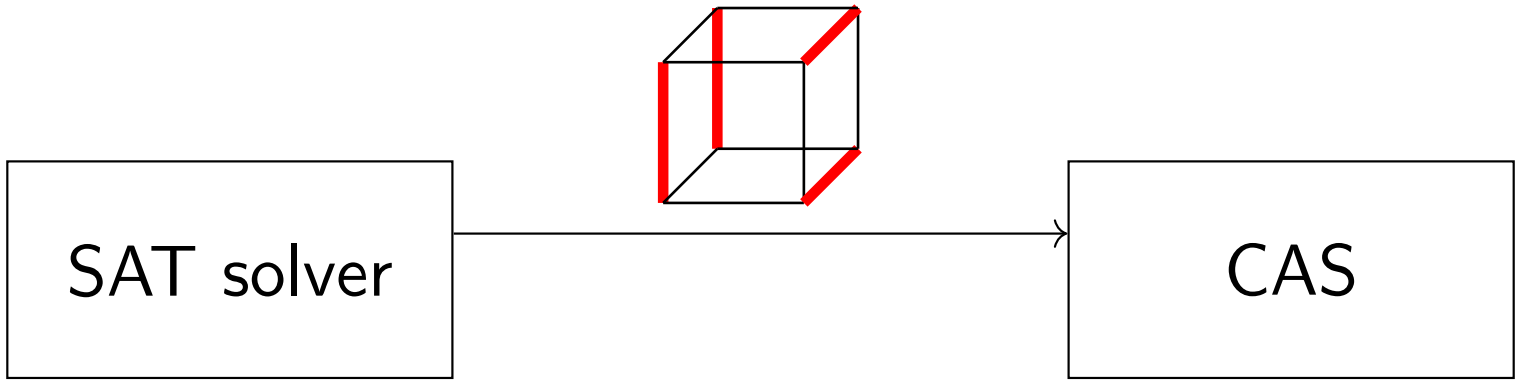
To search for a counterexample we want a matching that does *not* extend to a Hamiltonian cycle.

This is more difficult to encode in SAT, but **given** a matching a CAS can easily check if it extends to a Hamiltonian cycle.

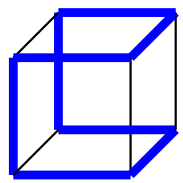
SAT learning



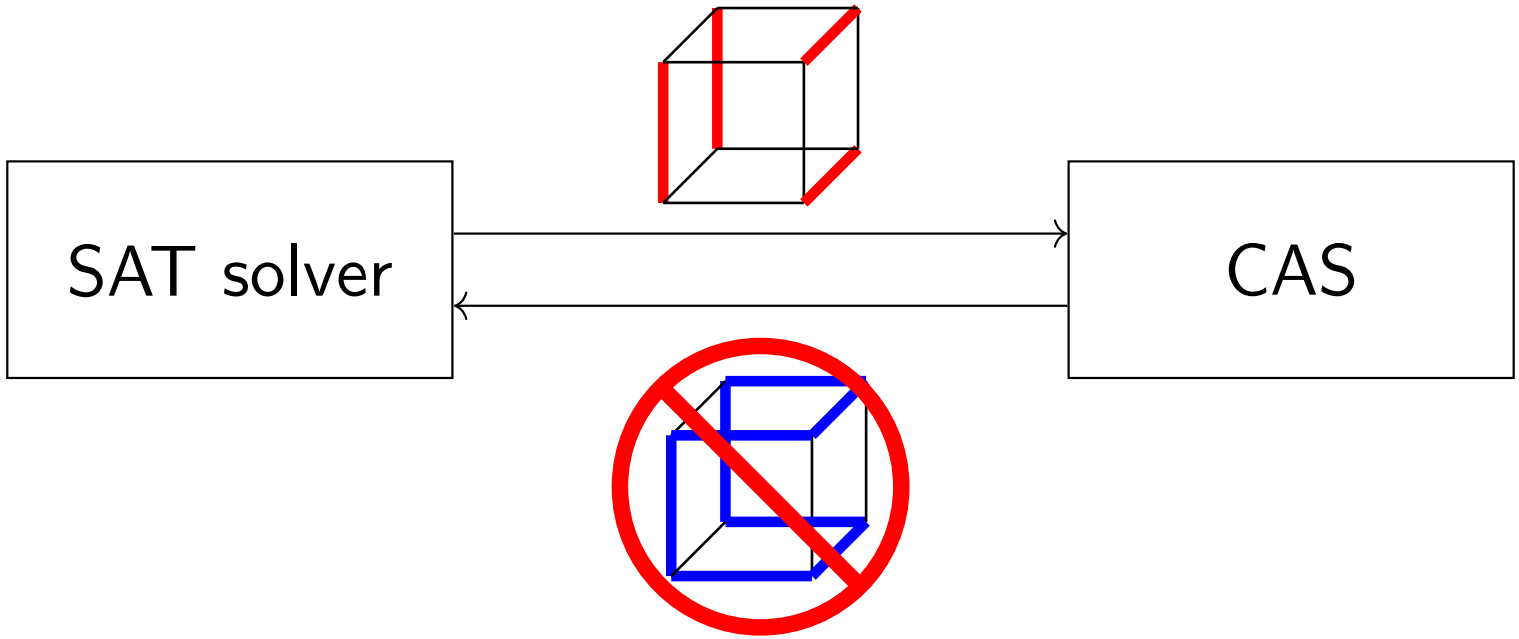
SAT learning



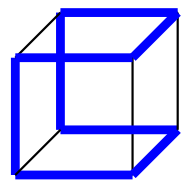
The CAS tries to extend the matching to a Hamiltonian cycle...



SAT learning



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



...if successful, a "conflict clause" is learned.

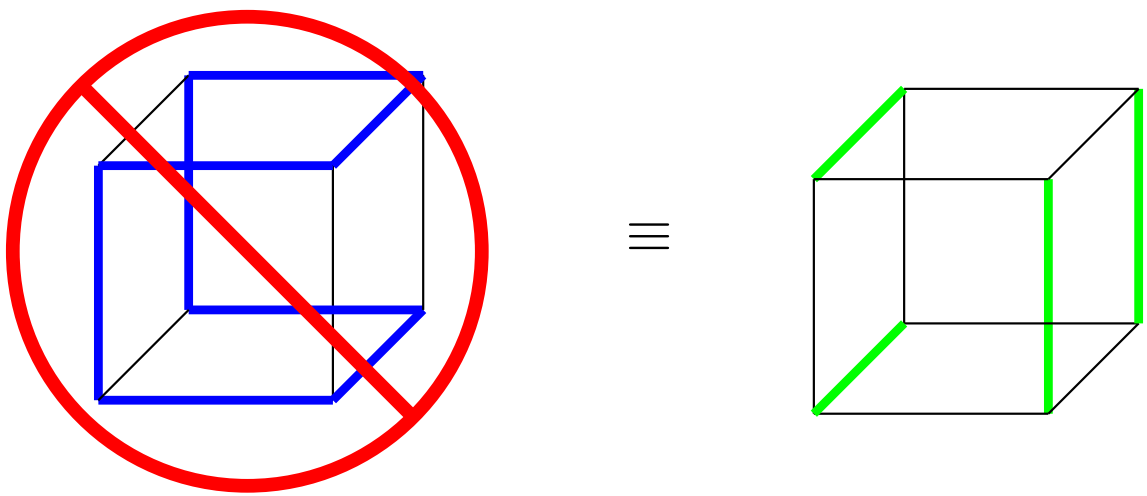
Conflict clause

We can block all subsets of the found Hamiltonian cycle since all trivially extend to a Hamiltonian cycle.

Conflict clause

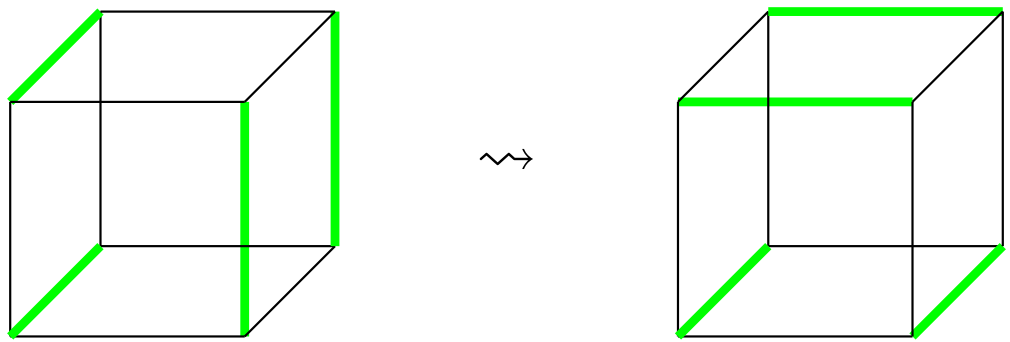
We can block all subsets of the found Hamiltonian cycle since all trivially extend to a Hamiltonian cycle.

We do this by ensuring that one of the edges *not* in the Hamiltonian cycle is in any future matching.



Symmetry learning

Applying any automorphism of the hypercube graph to a conflict clause generates another conflict clause:



Ruskey–Savage conjecture results

Previously it was known that the conjecture holds for the dimensions up to $d = 4$.

Ruskey–Savage conjecture results

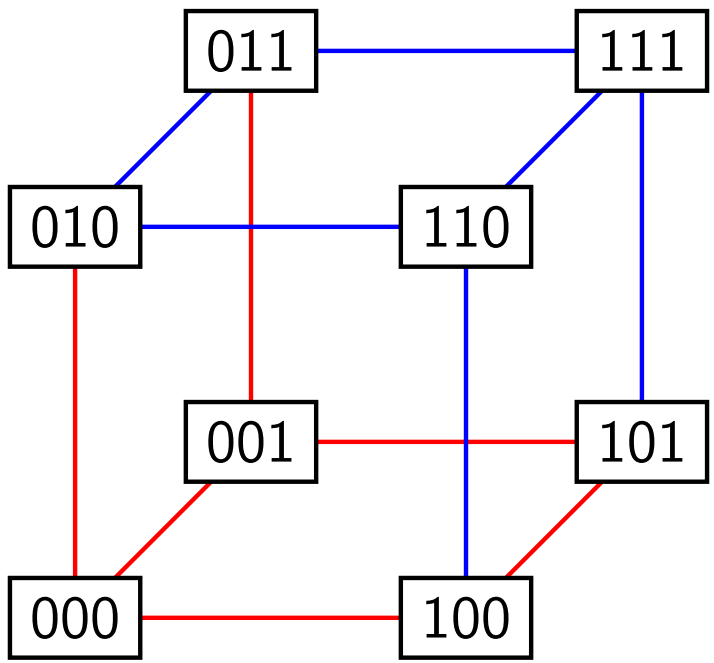
Previously it was known that the conjecture holds for the dimensions up to $d = 4$.

MathCheck solved the $d = 5$ case by checking just 2441 matchings (out of a possible ≈ 13 billion) in 65 minutes.

Norin Conjecture

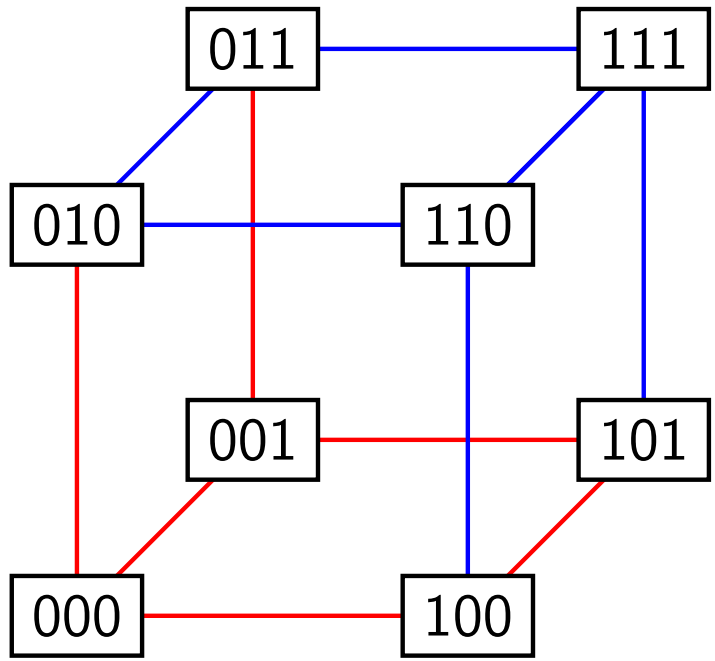
Edge-antipodal colourings

A $\{\text{red}, \text{blue}\}$-colouring of a hypercube graph is *edge-antipodal* if antipodal edges are coloured differently.



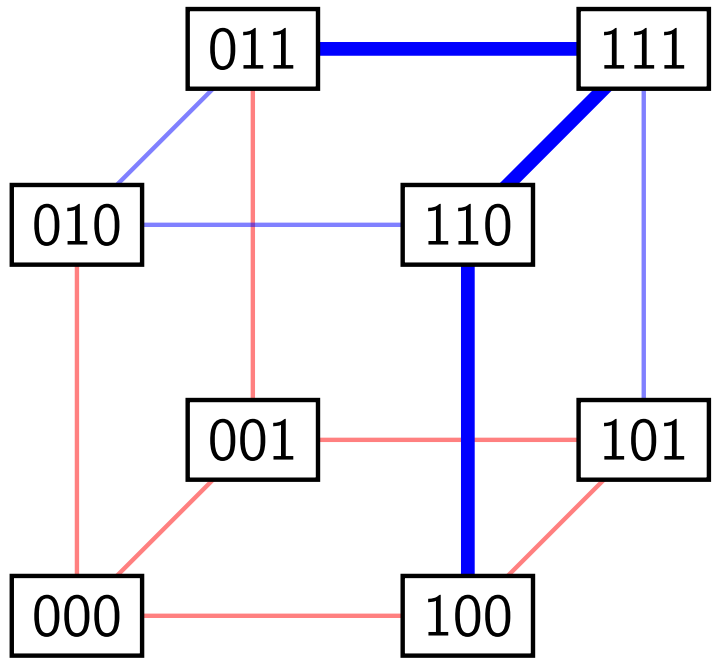
Norin conjecture

In any edge-antipodal colouring of the hypercube graph of dimension at least two there is a blue path joining antipodal vertices.



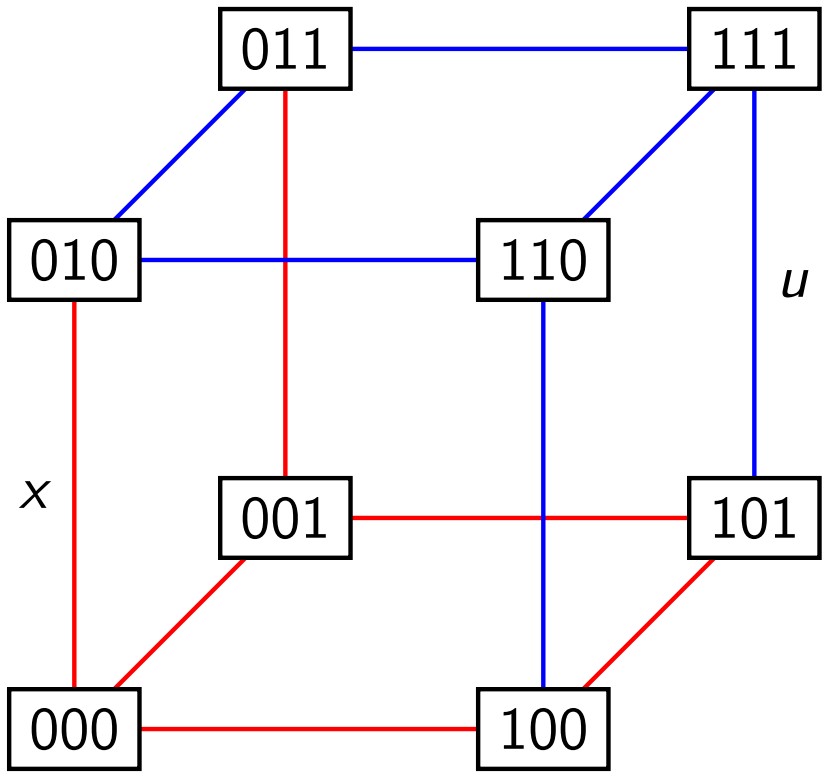
Norin conjecture

In any edge-antipodal colouring of the hypercube graph of dimension at least two there is a blue path joining antipodal vertices.



SAT variables

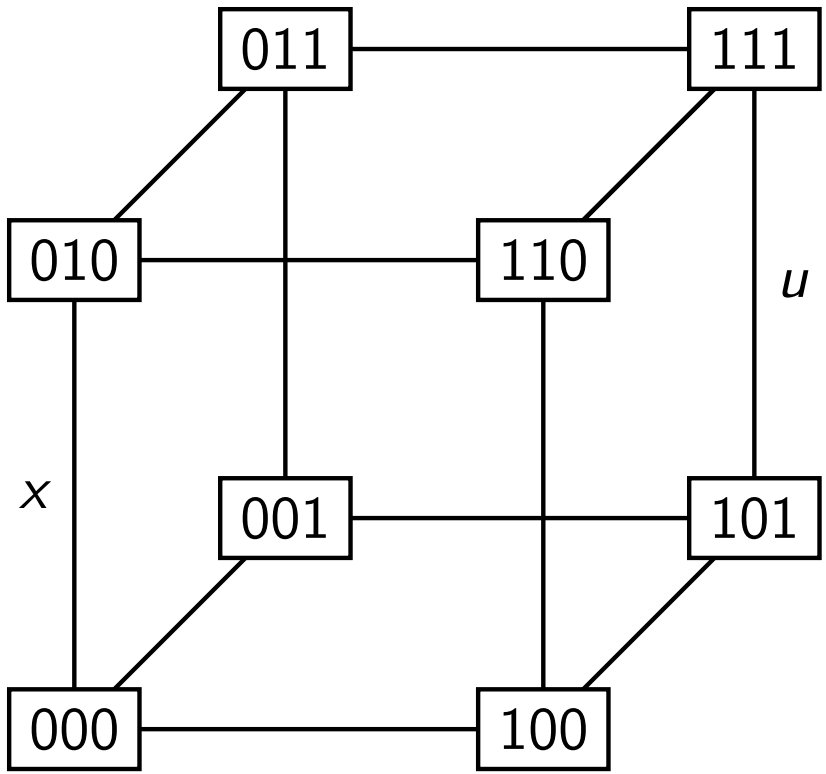
Define a Boolean variable for every edge of the hypercube graph.



In this colouring x is true (red) and u is false (blue).

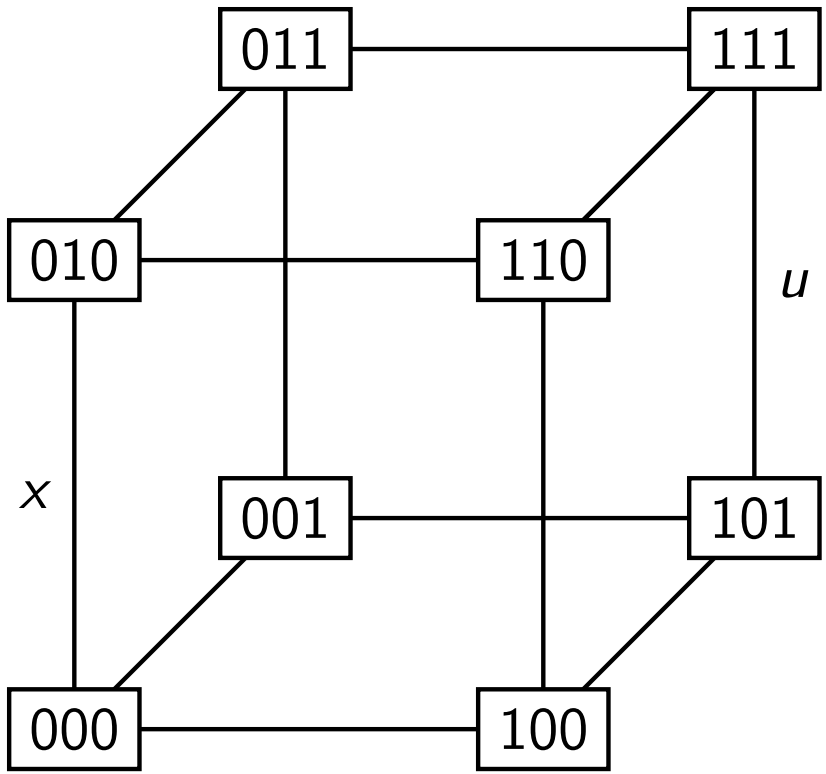
SAT constraints

For antipodal edges of the graph, at least one must be coloured red and at least one must be coloured blue.



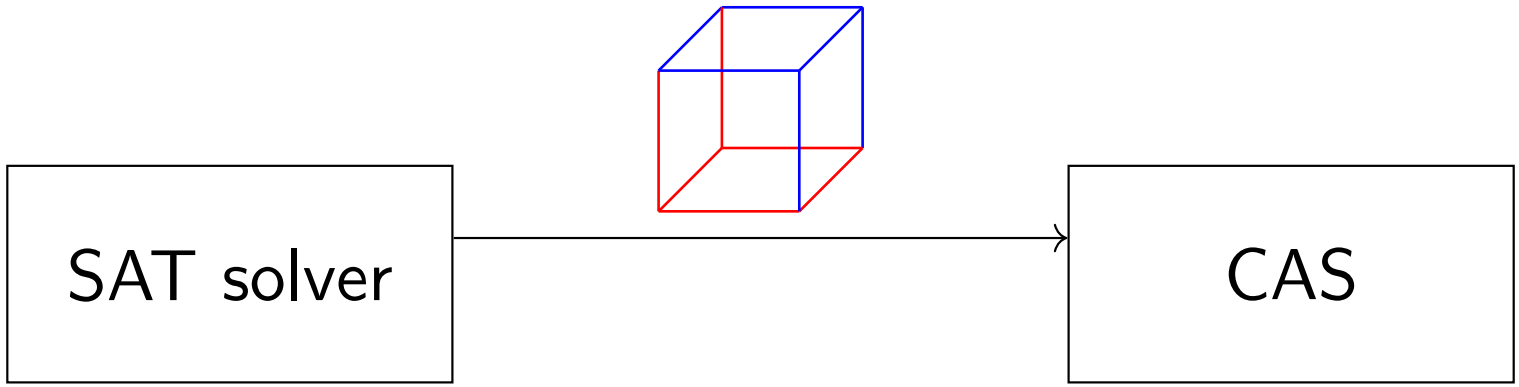
SAT constraints

For antipodal edges of the graph, at least one must be coloured red and at least one must be coloured blue.

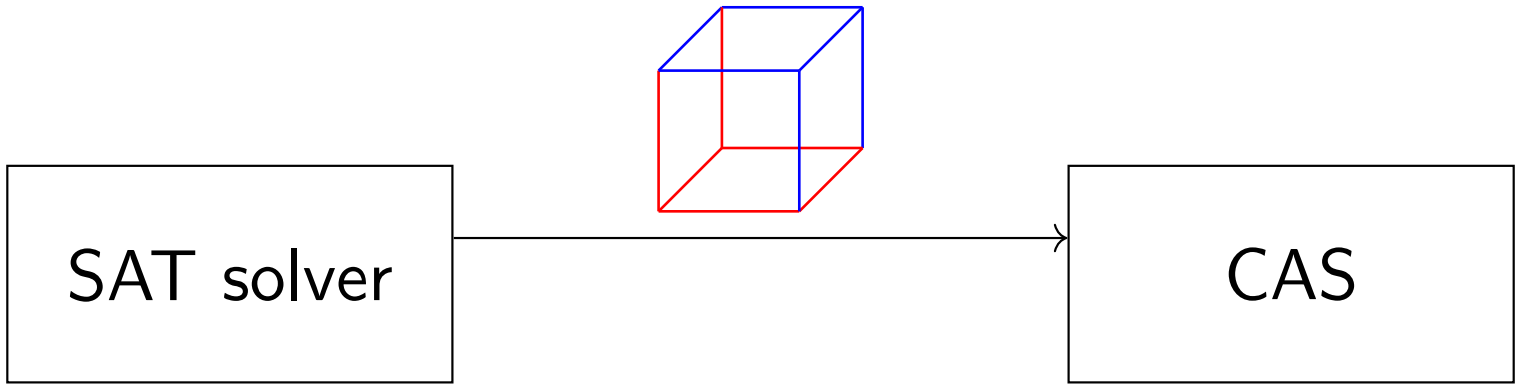


In Boolean logic: $x \vee u$, $\neg x \vee \neg u$

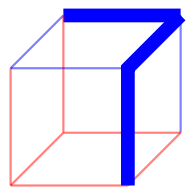
SAT learning



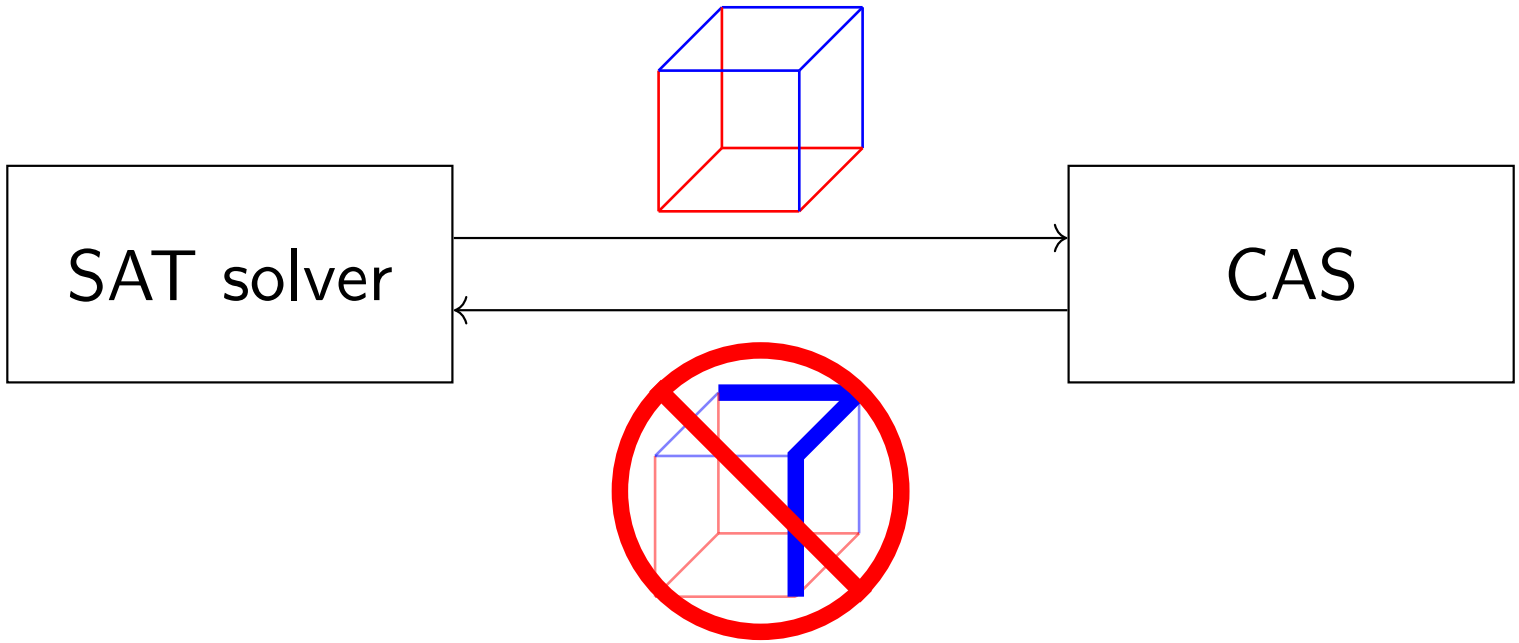
SAT learning



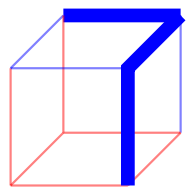
The CAS looks for a blue path between antipodes...



SAT learning



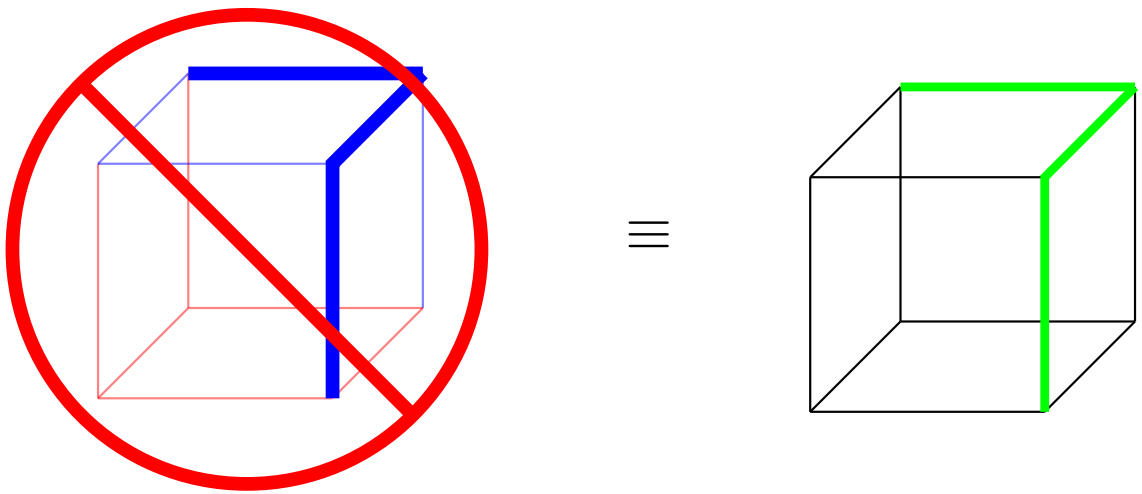
The CAS looks for a blue path between antipodes...



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

Conflict clause

We block all colourings containing the found blue path—at least one of the edges on the blue path must be red.



Norin conjecture results

Previously it was known that the conjecture holds for the dimensions up to five.

MathCheck solved the dimension 6 case by checking just 122 edge-antipodal colourings (out of a possible $\approx 7.9 \cdot 10^{28}$) in 3 minutes.

Lam's Problem

History

For over 2000 years, mathematicians tried to derive Euclid's "parallel postulate" from his first four postulates for geometry.

History

For over 2000 years, mathematicians tried to derive Euclid's "parallel postulate" from his first four postulates for geometry.

In the 1800s, the discovery of geometries like *projective geometry* where the parallel postulate fails showed that this is impossible.

Projective planes

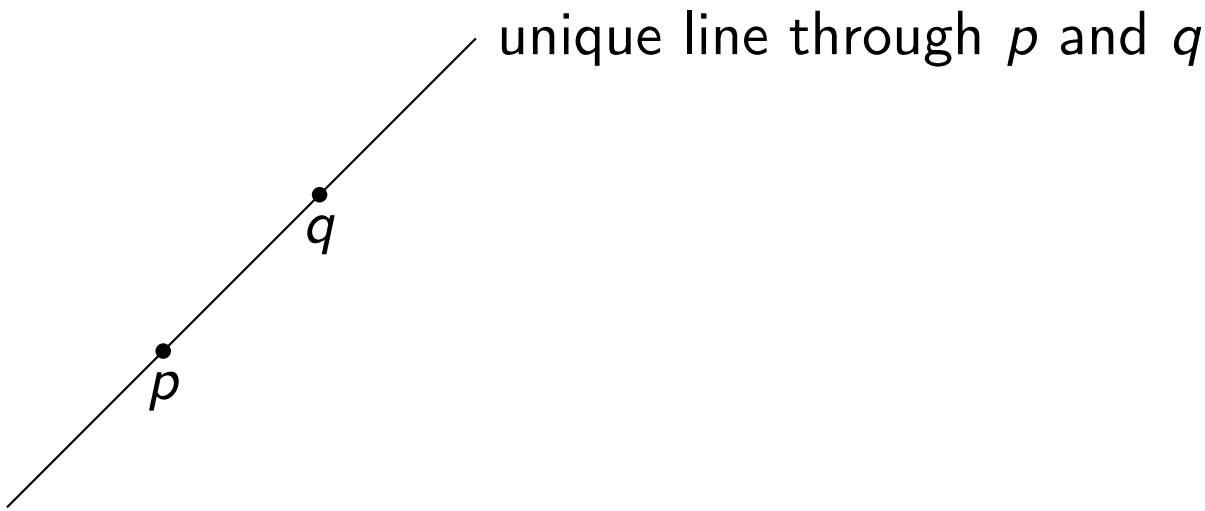
A *projective plane* is a collection of points and lines and a relation between points and lines such that:

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:

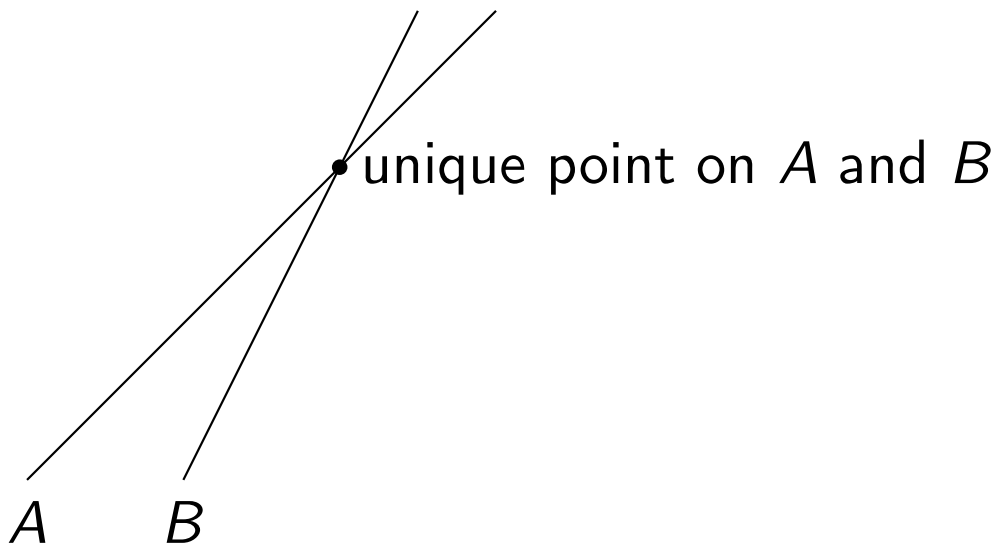
1. There is a unique line through any two points.
2. Any two lines intersect at a unique point.



Projective planes

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

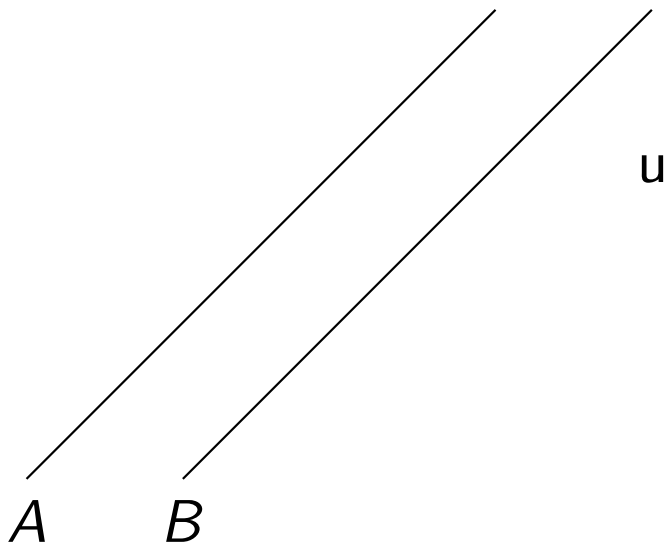
1. There is a unique line through any two points.
2. Any two lines intersect at a unique point.



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.

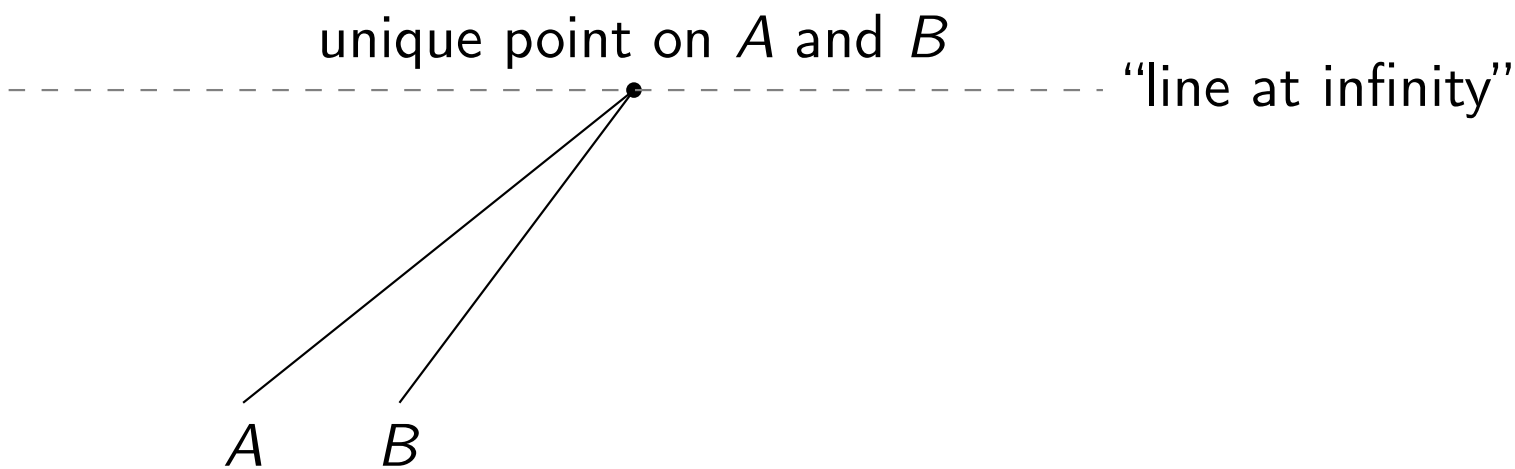


unique point on A and B?

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.



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.

To eliminate trivial cases:

3. No line contains all (or all but one) of the points.

Finite projective planes

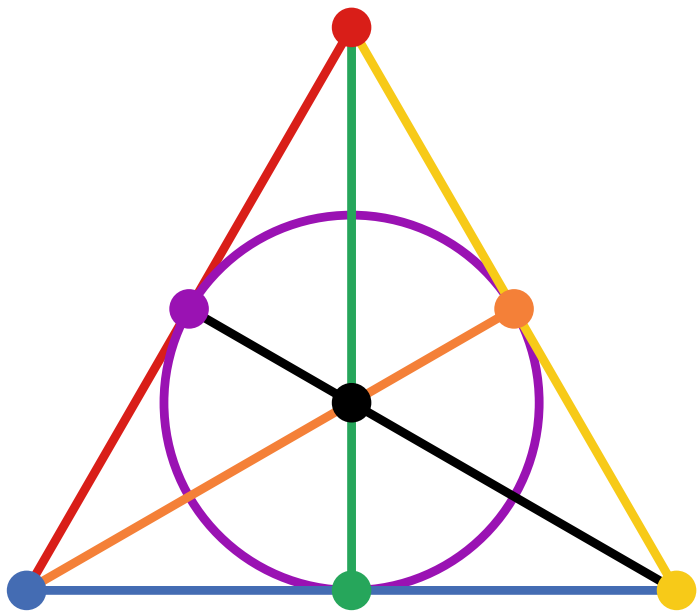
Can a projective plane have a finite number of points?

Finite projective planes

Can a projective plane have a finite number of points?

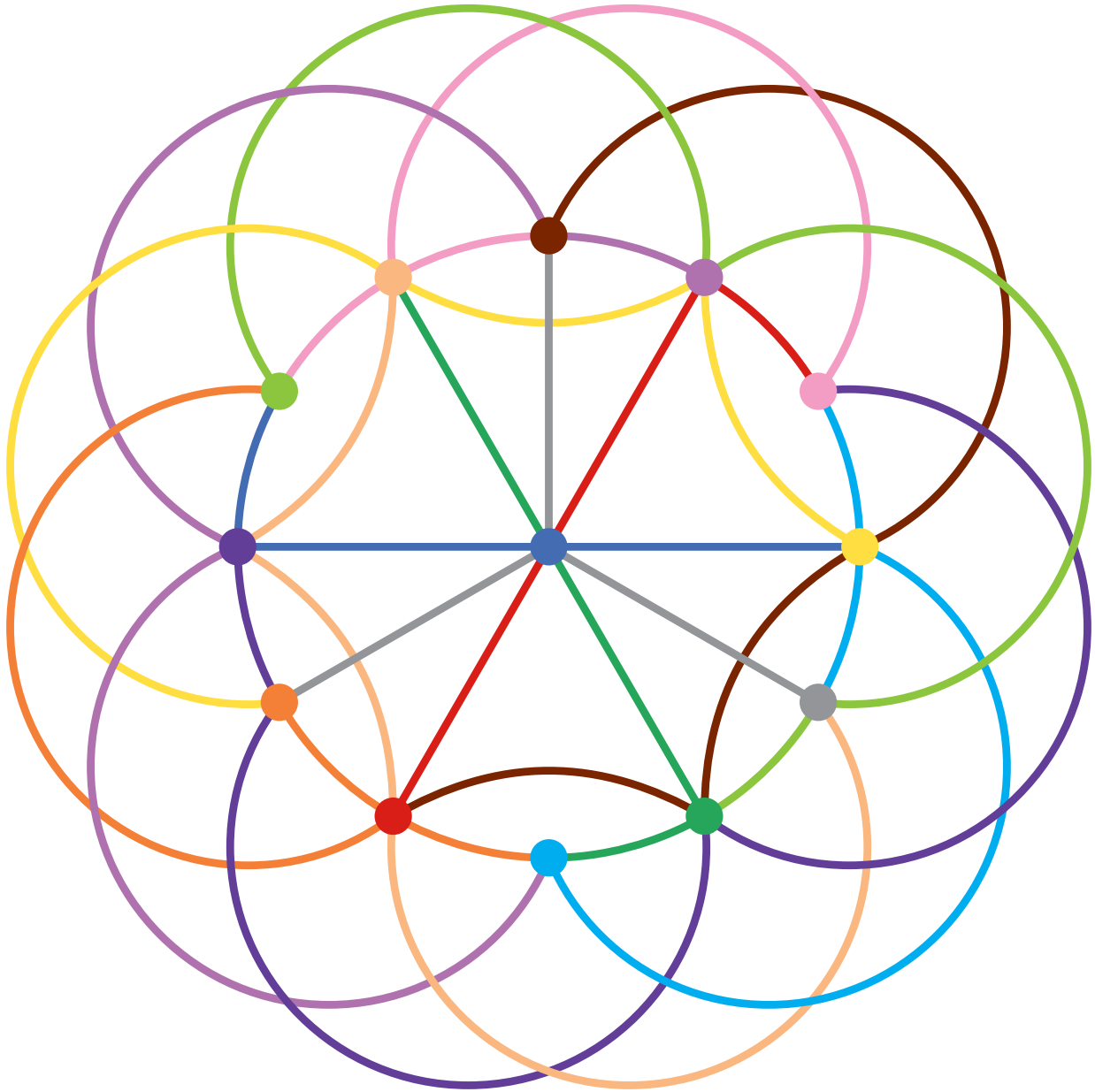
If so, by a counting argument it must have $n^2 + n + 1$ points for some integer n (called the *order* of the plane).

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 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 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 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:

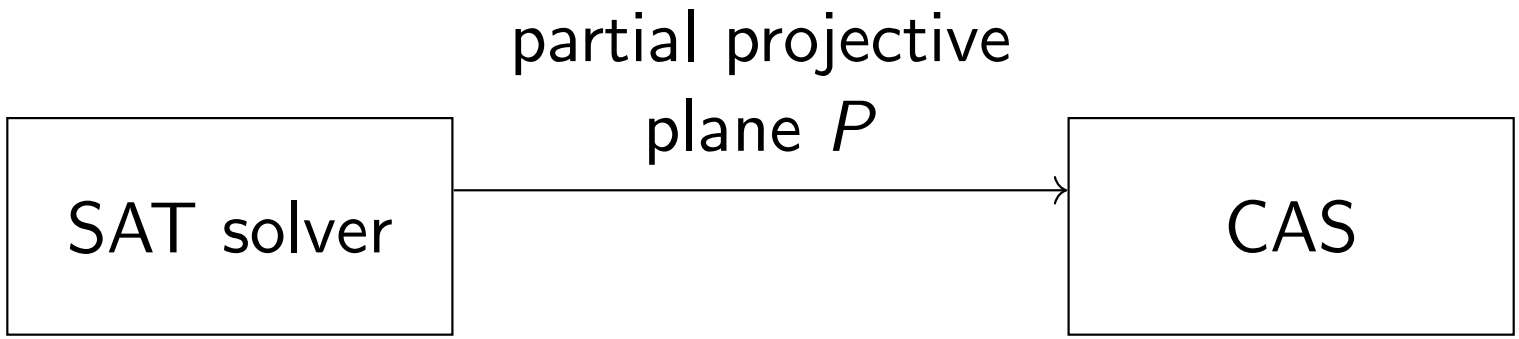
Authors	**Year**	**Language**	**Time**
Roy	2011	C	16,000 hours

Lam's problem: Second case

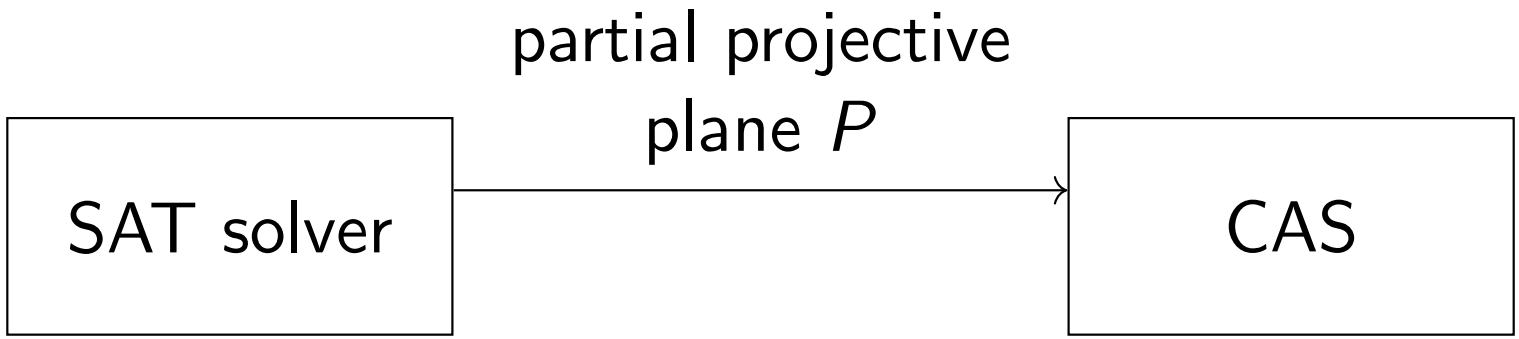
The second case was initiated in 1974 and not entirely searched until 1986. I am only aware of a single verification on a modern desktop prior to our work:

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

Learning method

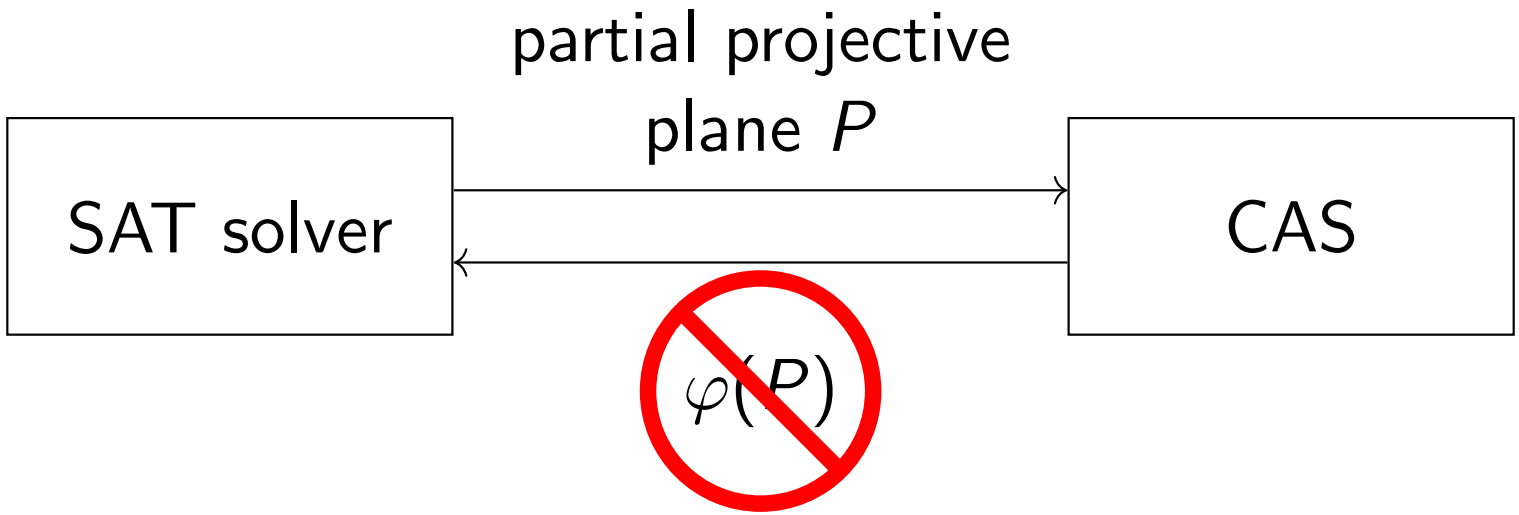


Learning method



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.

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. We are currently working on this case.

Hadwiger–Nelson Problem

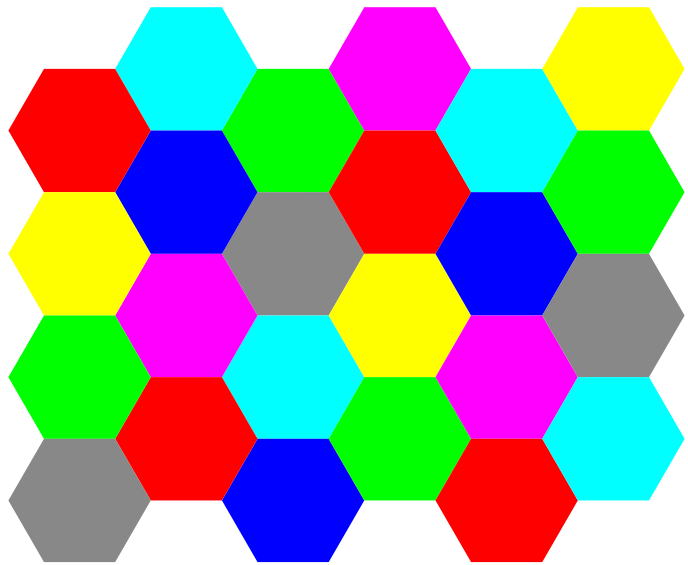
Hadwiger–Nelson problem

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?

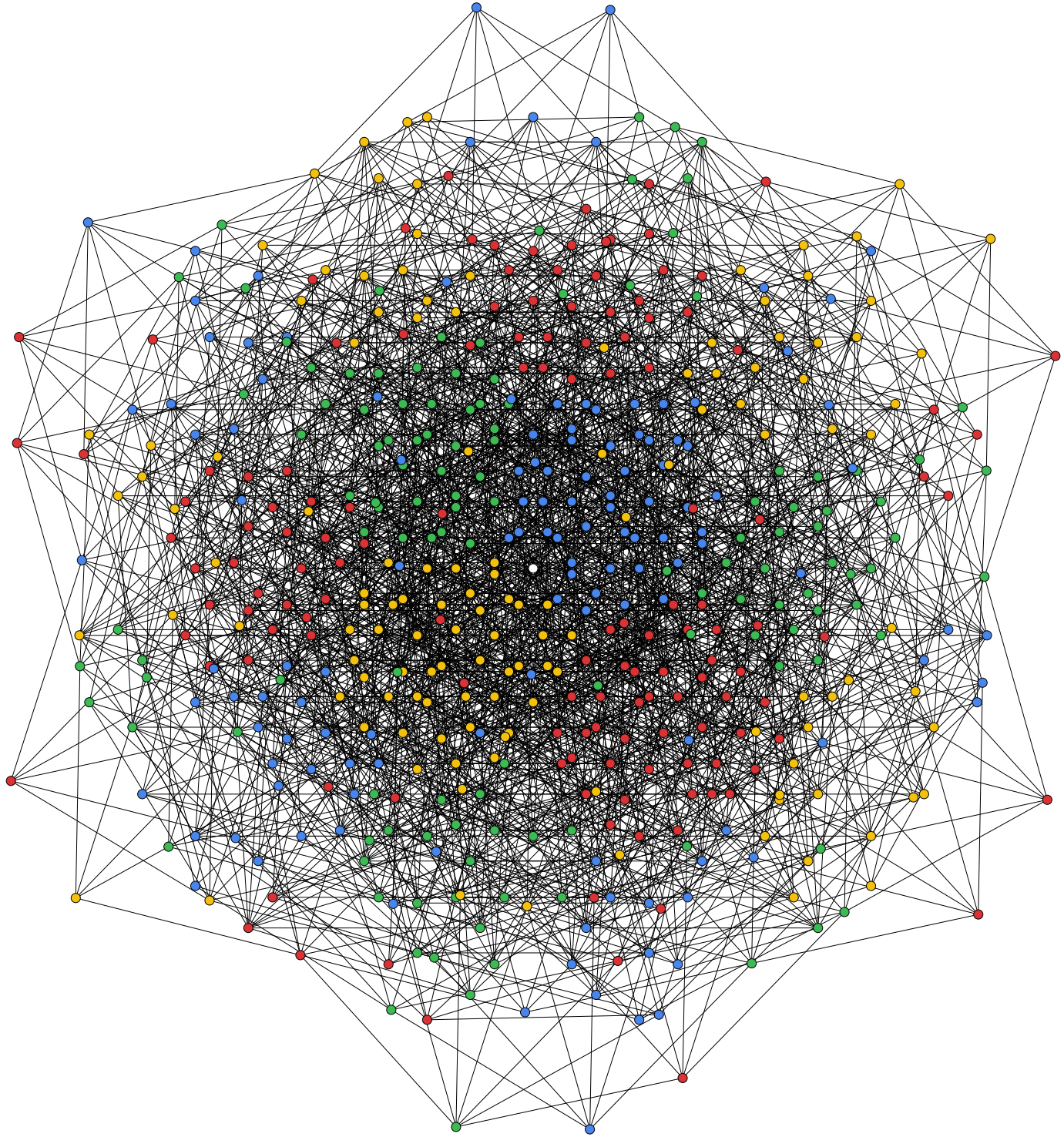
At most seven:



Hadwiger–Nelson problem results

Last year, Heule used a SAT+CAS method to find a unit distance graph with 529 vertices that cannot be coloured with four colours.

Marijn Heule. Trimming graphs using clausal proof optimization. *Principles and Practice of Constraint Programming*, 2019.



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.

Has 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.

Has 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.

Future work

I'm actively looking for a permanent place to continue this research program—and to develop new applications.

For the next two years I'm holding an NSERC fellowship, working with Kevin Cheung and Brett Stevens.

`curtisbright.com`