A SAT-based Resolution of Lam's Problem

Curtis Bright
University of Windsor

Joint work with Kevin Cheung, Brett Stevens, Ilias Kotsireas, Vijay Ganesh

AAAI-21
January 2021

Overview

We solve **Lam's problem** from geometry by generating verifiable proofs with satisfiability (SAT) solvers and computer algebra.

Computer Science team solves centuries-old math problem

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

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.

The answer: "No."

So far it seems simple. But the reality is far different. The Concordia team had to search through 1,000,000,000,000,000 (that's a thousand trillion) combinations using one of the fastest supercomputers on earth to arrive at a solution.

The problem is so complex that the only way another party could prove the team's answer would be to do the calculations all over again — something which only the most intrepid investigators would even contemplate. The problem has already taken up the lifetimes of many eminent mathematicians.

The particular skill required was in organizing and programming the computer rather than in formulating the equations themselves.

The skills which the team gained while solving the problem have future applications in developing communications networks and cryptography.

Charles Bélanger

Let's see, if we added up the IQs of Concordia's crack Computer Science team, the number, though not in the trillions, would still be pretty high. Members of the team are (left to right) ***Larry Thiel, John McKay, Stanley Swiercz*** *and* ***Clement Lam.***

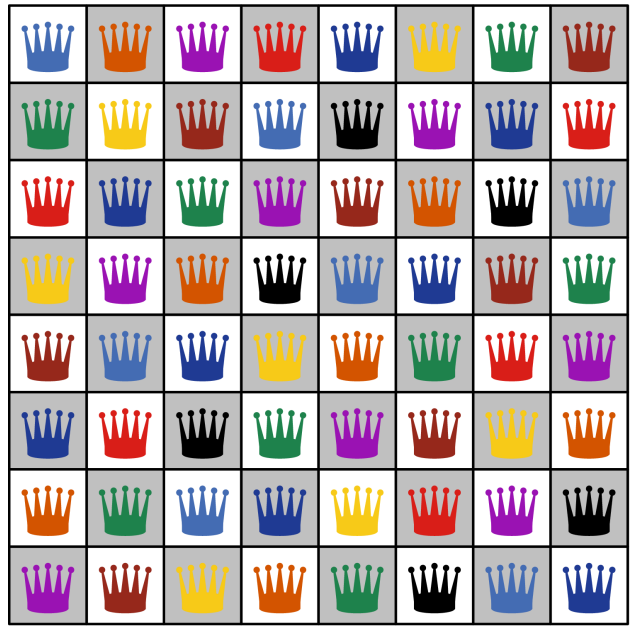
SAT:

Boolean satisfiability problem

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:

- Discrete optimization
- Hardware and software verification
- Proving/disproving conjectures



Additionally, SAT solvers produce unsatisfiability certificates when no solutions exist.

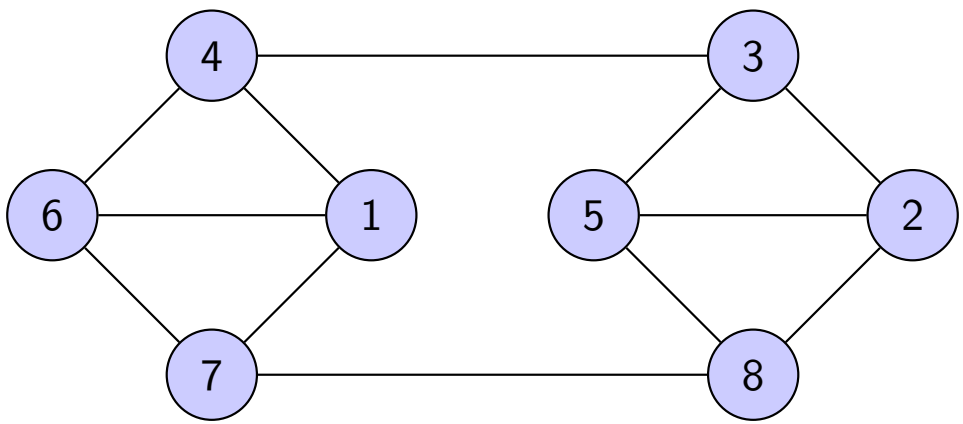
CAS:

Computer algebra system

Effectiveness of CASs

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

- Row reducing a matrix
- Determining if graphs are isomorphic
- Computing the symmetries of combinatorial objects



For example, the symmetry group of this graph has generators $(2\ 5)$, $(3\ 8)(4\ 7)$, and $(1\ 2)(3\ 4)(5\ 6)(7\ 8)$.

SAT + CAS

Search + Math

MathCheck: A SAT+CAS system

We've used MathCheck to construct many kinds of combinatorial objects (see `uwaterloo.ca/mathcheck`).

Hadamard 160 in Cool Tones

Projective Geometry

History

Since 300 BC, mathematicians tried to derive Euclid's "parallel postulate" from his other axioms for geometry.

History

Since 300 BC, mathematicians tried to derive Euclid's "parallel postulate" from his other axioms for geometry.

In the 1800s it was realized this is impossible!

Projective geometries

Projective geometries do not satisfy the parallel postulate but do satisfy Euclid's other axioms.

Instead, projective geometries satisfy a "projective axiom" which says that any pair of lines meet at a unique point.

Projective geometries

Projective geometries do not satisfy the parallel postulate but do satisfy Euclid's other axioms.

Instead, projective geometries satisfy a "projective axiom" which says that any pair of lines meet at a unique point.

All projective geometries with a finite number of points have been classified—except for those with exactly two dimensions (the *projective planes*).

Projective planes: Examples

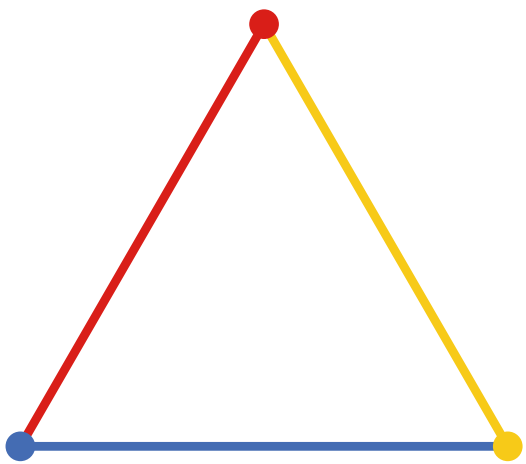
Projective planes satisfy the following axioms:

- Every pair of lines meet at a unique point.
- Every pair of points define a unique line.
- Every line contains $n + 1$ points for some *order* n.

Projective planes: Examples

Projective planes satisfy the following axioms:

- Every pair of lines meet at a unique point.
- Every pair of points define a unique line.
- Every line contains $n + 1$ points for some *order* n.

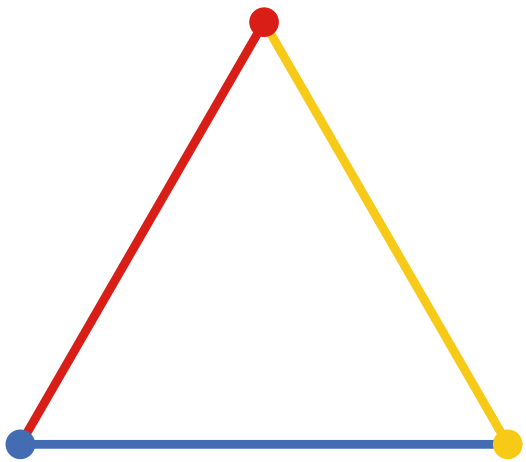


order 1

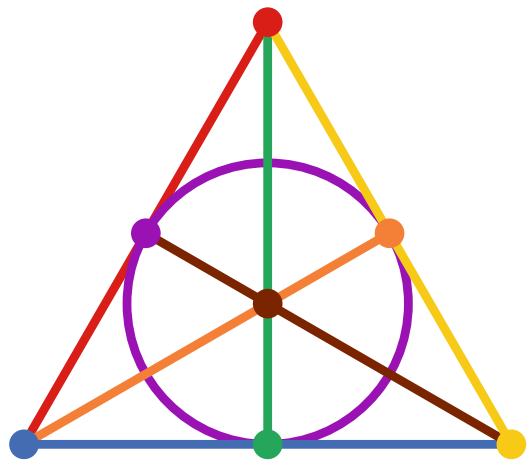
Projective planes: Examples

Projective planes satisfy the following axioms:

- Every pair of lines meet at a unique point.
- Every pair of points define a unique line.
- Every line contains $n + 1$ points for some *order* n.



order 1

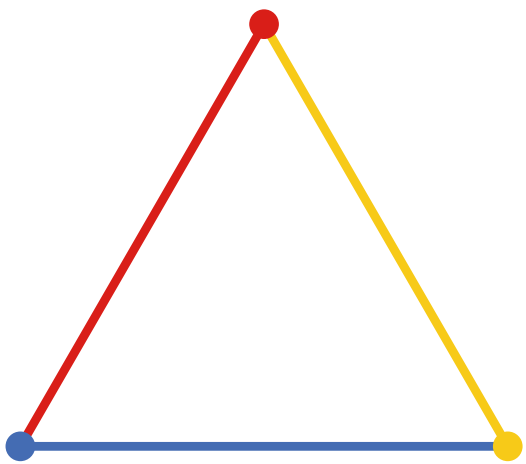


order 2

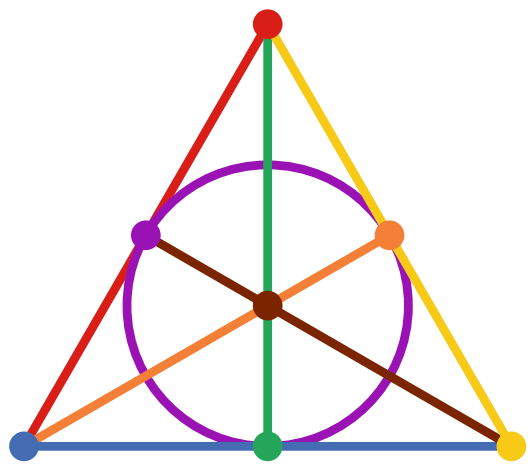
Projective planes: Examples

Projective planes satisfy the following axioms:

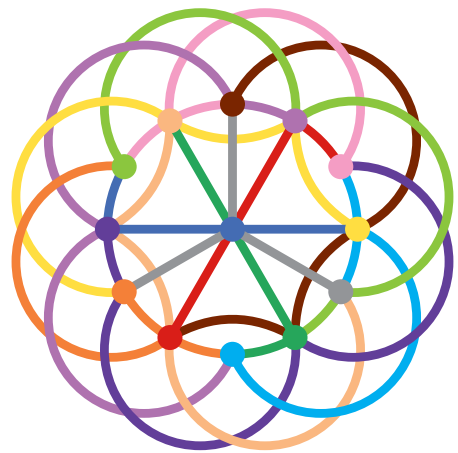
- Every pair of lines meet at a unique point.
- Every pair of points define a unique line.
- Every line contains $n + 1$ points for some *order* n.



order 1



order 2

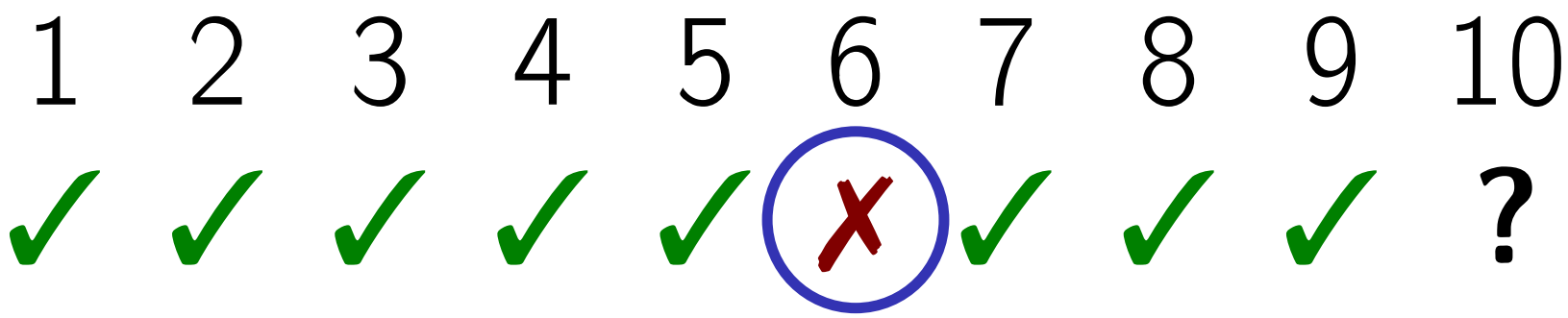


order 3

Projective planes of small orders

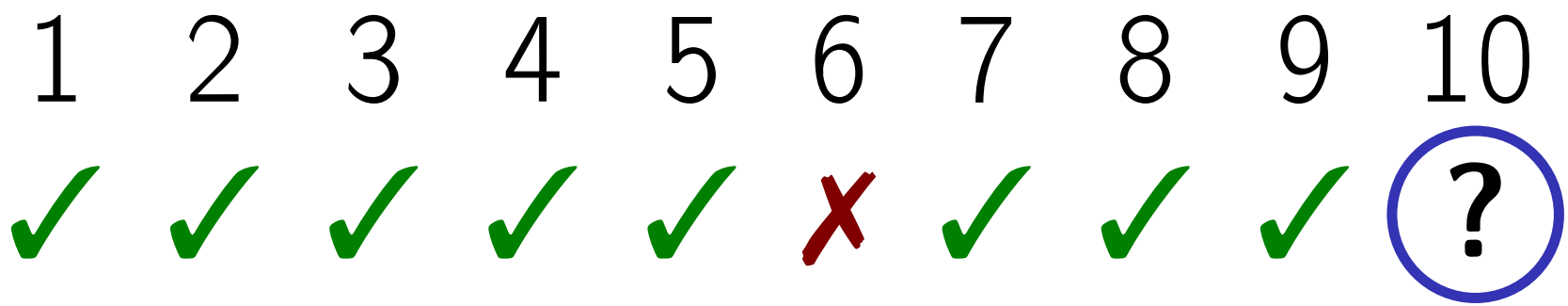
1	2	3	4	5	6	7	8	9	10
✓	✓	✓	✓	✓	✗	✓	✓	✓	**?**

Projective planes of small orders



Theoretical obstruction

Projective planes of small orders



No such plane known

No theoretical obstruction known

Projective planes of small orders

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

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

Clement Lam

Lam's problem

The first critical value of n is $n = 10$. A thorough investigation of this case is currently beyond the facilities of computing machines.

Marshall Hall Jr.
Finite Projective Planes
1955

Lam's problem

The first critical value of n is $n = 10$. A thorough investigation of this case is currently beyond the facilities of computing machines.

Marshall Hall Jr.
Finite Projective Planes
1955

Could computers resolve Lam's problem?

Incidence matrix encoding

The *incidence matrix* of a projective plane is a $\{0, 1\}$ matrix encoding which lines (rows) contain which points (columns):

1	1	0
1	0	1
0	1	1

order 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

order 2

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

order 3

Each row contains exactly $n + 1$ ones (in order n).

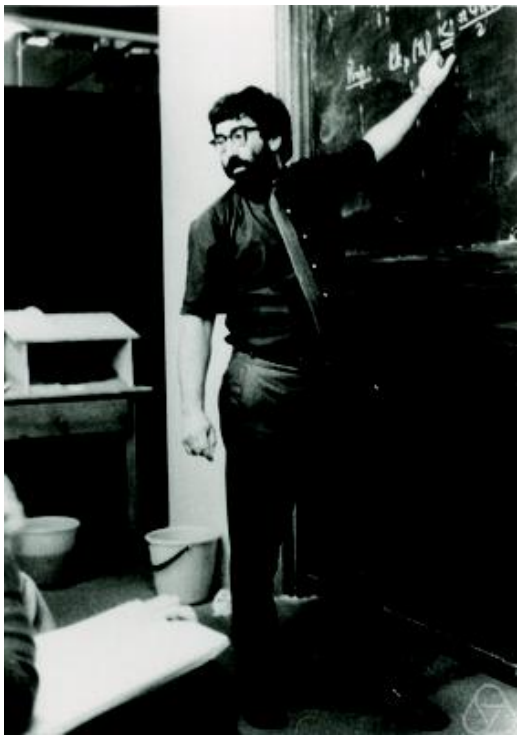
The inner product of any two rows or columns is exactly 1.

Enter coding theory

The *code* generated by a projective plane is the row space of its incidence matrix over $F_2 = \{0, 1\}$.

The *weight* of a $\{0, 1\}$ word is the number of 1s it contains.

A theoretical breakthrough



In 1970, Assmus proved strong properties about how many words of each weight must exist in the code generated by a hypothetical projective plane of order ten.

In particular, the code must contain words of weight 15, 16, or 19.

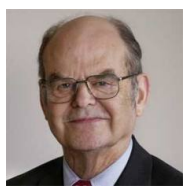
Searches for codewords

A number of searches performed in the twentieth century found no codewords of weight 15, 16, and 19:

Weight	**Year**	**Time**[1]	**Authors**
15	1973	2 h	MacWilliams et al.
16	1974	1,300 h	Carter
	1986	1,900 h	Lam et al.
19	1989	320,000 h	Lam et al.


MacWilliams


Sloane


Thompson


Carter


Lam


Thiel


Swiercz

[1] Estimated time on a VAX machine (one million instructions per second).

Correctness of the result

These searches used custom-written software run once on a single piece of hardware. We must simply trust the searches ran to completion.

This is a lot of trust. The authors were upfront that mistakes were a real possibility.

Correctness of the result

These searches used custom-written software run once on a single piece of hardware. We must simply trust the searches ran to completion.

This is a lot of trust. The authors were upfront that mistakes were a real possibility.

We found discrepancies with the intermediate results of Lam et al. and an independent search of Roy in 2011.

Nonexistence Certificates

Nonexistence certificates

We provide certificates that an independent party can use to verify the nonexistence of a projective plane of order ten.

The certificates rely on an encoding of the existence problem into Boolean logic.

SAT encoding

Each entry in the incidence matrix is represented by a Boolean variable which is true exactly when the entry contains a 1.

How should the projective axiom be encoded in Boolean logic?

SAT encoding

Each entry in the incidence matrix is represented by a Boolean variable which is true exactly when the entry contains a 1.

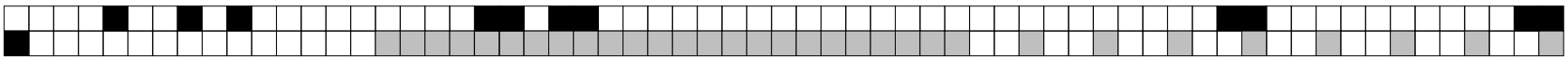
How should the projective axiom be encoded in Boolean logic?

Every pair of lines meet exactly once, therefore:

1. Every pair of lines meet *at most* once.
2. Every pair of lines meet *at least* once.

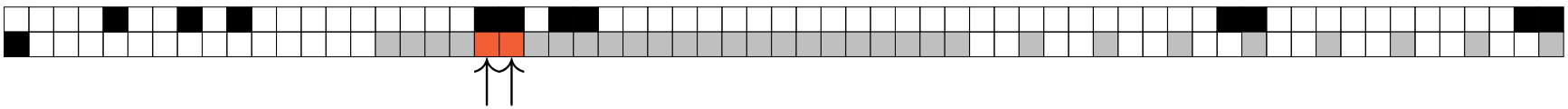
1. Lines meet at most once

Consider the following two lines (grey entries unknown) as an example:



1. Lines meet at most once

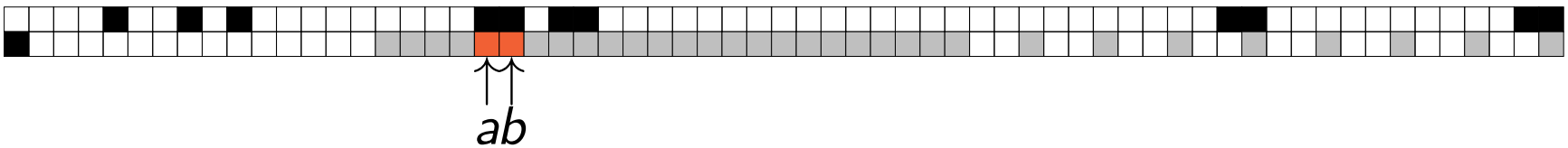
Consider the following two lines (grey entries unknown) as an example:



The highlighted entries cannot both be true or the lines would meet more than once.

1. Lines meet at most once

Consider the following two lines (grey entries unknown) as an example:



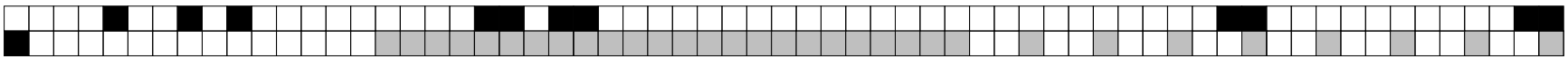
The highlighted entries cannot both be true or the lines would meet more than once.

In Boolean logic:

$$\neg a \vee \neg b$$

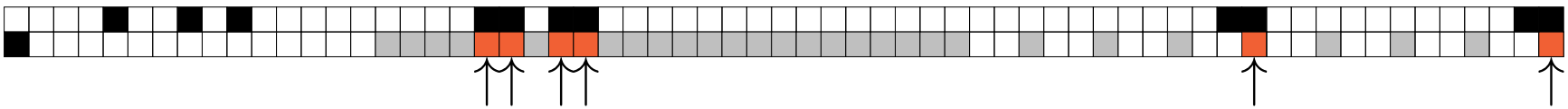
2. Lines meet at least once

Consider the following two lines (grey entries unknown) as an example:



2. Lines meet at least once

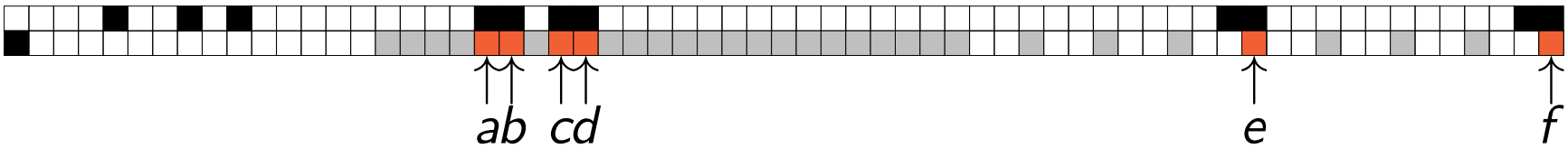
Consider the following two lines (grey entries unknown) as an example:



At least one of the highlighted entries must be true for the lines to meet.

2. Lines meet at least once

Consider the following two lines (grey entries unknown) as an example:



At least one of the highlighted entries must be true for the lines to meet.

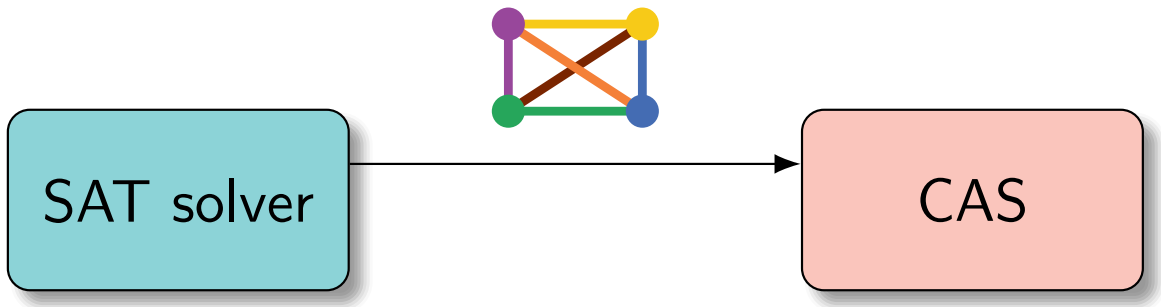
In Boolean logic:

$$a \vee b \vee c \vee d \vee e \vee f$$

Isomorphism Blocking

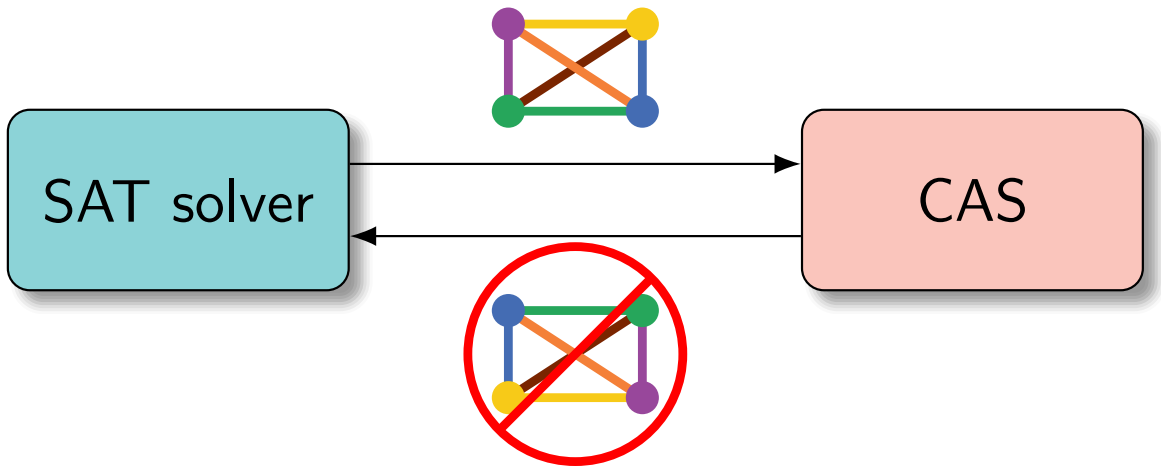
CAS isomorphism blocking

The SAT solver finds partial solutions and sends them to a CAS...



CAS isomorphism blocking

The SAT solver finds partial solutions and sends them to a CAS...



...and the CAS finds a nontrival isomorphism and blocks it.

Results

We generated certificates demonstrating the nonexistence of words of each required weight:

Weight	**Compute time**	**Certificate size**	**Appearing**
15	7 seconds	35 MiB	AAECC 2020
16	30 hours	325 GiB	IJCAI 2020
19	24 months	110 TiB	AAAI 2021

Our searches also completed the fastest—in 2011, a search for weight 16 words required 16,000 hours.

Necessary trust

To believe this nonexistence result you now need to trust:

- The SAT encoding and script to generate the SAT instances.
- The proof verifier.
- The isomorphism blocking clauses—generated with the symbolic computation library nauty.

Conclusion

Many mathematical problems stand to benefit from fast, verifiable, and expressive search tools.

Requires some knowledge of SAT and CAS—but avoids using special-purpose search code that is

- hard to write,
- even harder to make efficient,
- and extremely difficult to verify.

Future work

I'm actively looking for students and collaborators to extend and apply this paradigm to new applications.

Please get in touch if interested (and pass on the word to those who may be)!

Thank you!

`curtisbright.com`