

# A Resolution of Lam's Problem via Satisfiability Solvers

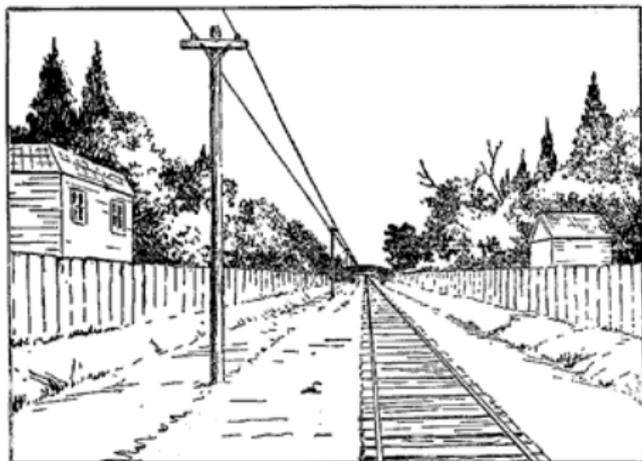
Curtis Bright  
University of Windsor

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

Canadian Mathematical Society Winter Meeting  
December 4, 2020

# Overview

We use a search method based on satisfiability solvers and computer algebra to generate certificates solving Lam's problem from projective geometry.



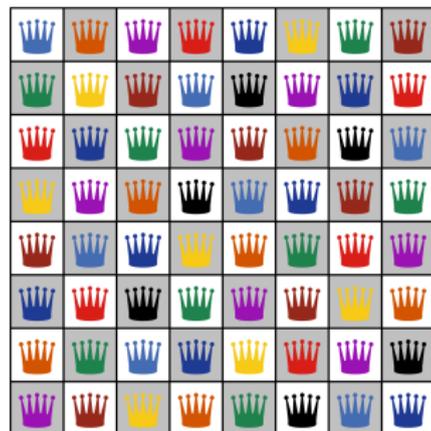
# SAT:

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.

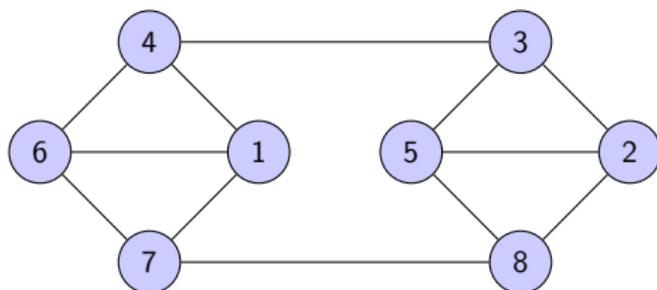
CAS:

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](http://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

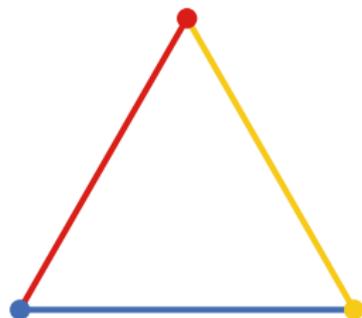
Projective planes satisfy the following axioms:

- ▶ 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*  $n$ .

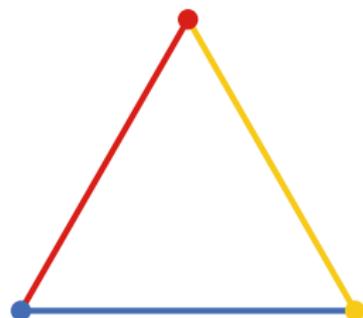


order 1

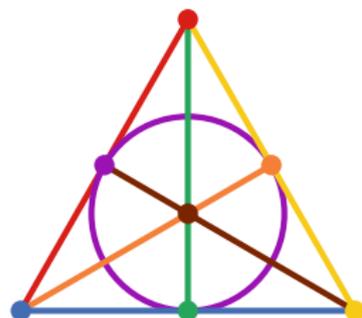
# Projective planes: Examples

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

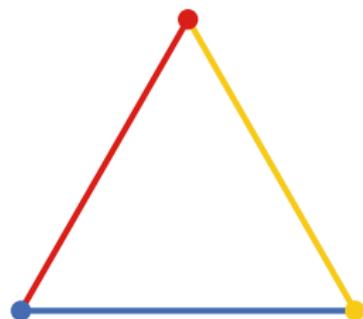


order 2

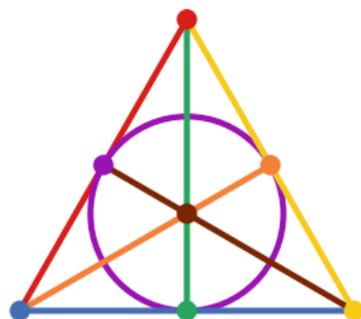
# Projective planes: Examples

Projective planes satisfy the following axioms:

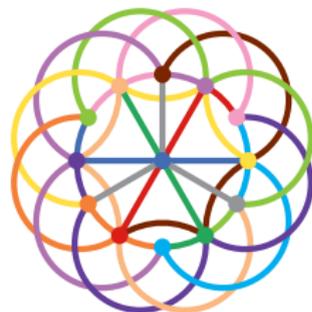
- ▶ Every pair of lines meet at a unique point.
- ▶ Every pair of points define a unique line.
- ▶ Every line contains  $n + 1$  points for some *order*  $n$ .



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

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

Theoretical obstruction

## Projective planes of small orders

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

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 <sup>1</sup>	Authors
15	1973	2 h	MacWilliams et al.
16	1974	1,300 h	Carter
16	1986	1,900 h	Lam et al.
19	1989	320,000 h	Lam et al.



MacWilliams



Sloane



Thompson



Carter



Lam



Thiel



Swiercz

---

<sup>1</sup>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.

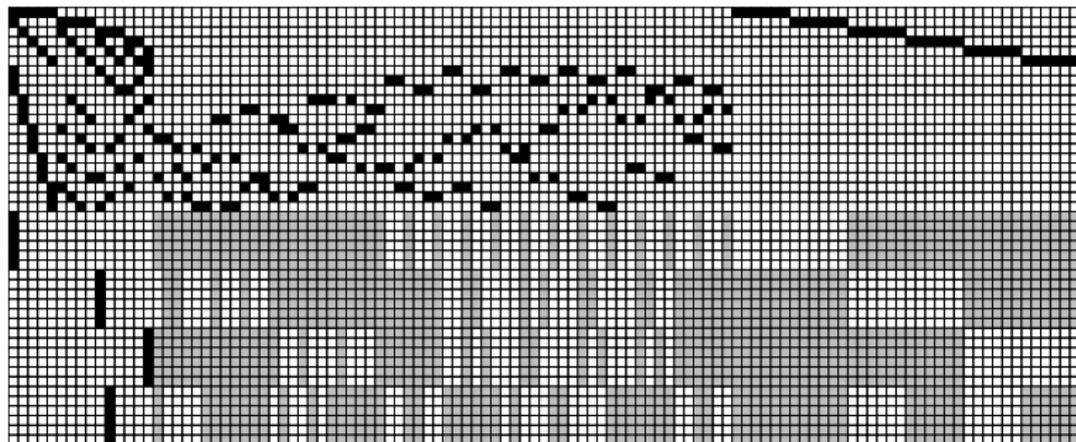
How should the projective axiom be encoded in Boolean logic?

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.

## Incidence matrix starting structure

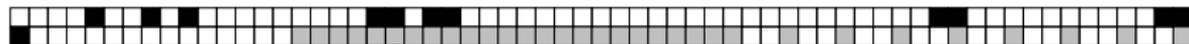
Supposing that a weight 15 codeword exists, without loss of generality one can prove that the first 45 rows of the plane's incidence matrix are:



Here black denotes 1, white denotes 0, and gray denotes unknown.

# 1. Lines meet at most once

Consider the following two lines as an example:



# 1. Lines meet at most once

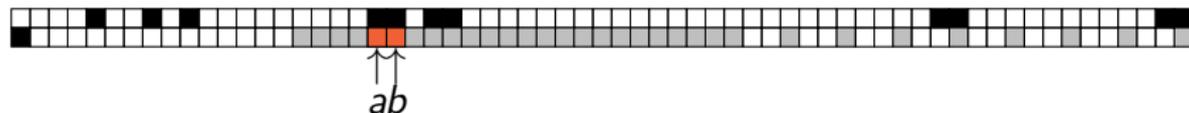
Consider the following two lines as an example:



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 as an example:



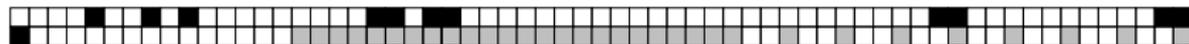
The highlighted entries cannot both be true or the lines would meet more than once.

In Boolean logic:

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

## 2. Lines meet at least once

Consider the following two lines as an example:



## 2. Lines meet at least once

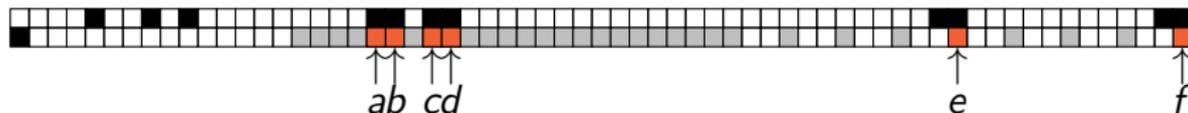
Consider the following two lines as an example:



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 as an example:



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

In Boolean logic:

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

# Solving the SAT instance

Just using the first 27 rows, the SAT instance has approximately:

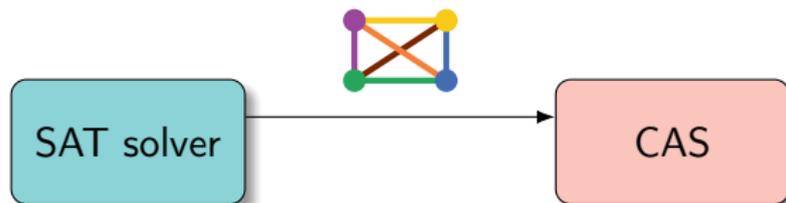
- ▶ 150 unknown variables
- ▶ 1000 clauses
- ▶  $10^{18}$  solutions

However, only 1021 of these solutions are distinct up to permuting their rows or columns (when represented as matrices).

# Isomorphism Blocking

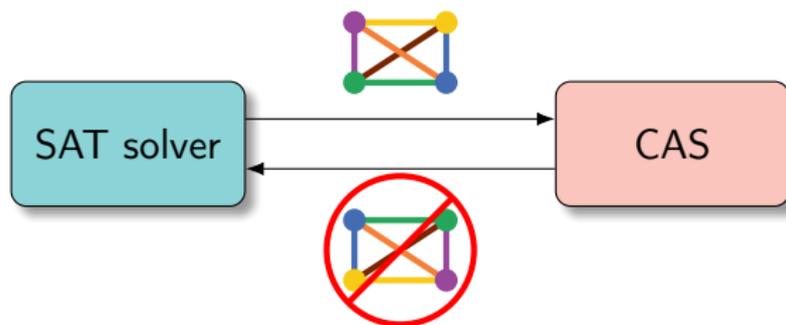
## CAS isomorphism blocking

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 nontrivial isomorphism and blocks it.

## Starting structures (weight 19 case)

In the weight 19 search, the incidence matrix starting structures were generated without and with CAS isomorphism blocking—demonstrating its effectiveness:

<b>Method</b>	<b># structures</b>	<b>Time</b>
SAT	56,157,420	20.0 h
SAT+CAS	650,370	0.4 h

## Starting structures (weight 19 case)

In the weight 19 search, the incidence matrix starting structures were generated without and with CAS isomorphism blocking—demonstrating its effectiveness:

<b>Method</b>	<b># structures</b>	<b>Time</b>
SAT	56,157,420	20.0 h
SAT+CAS	650,370	0.4 h

Double check: Each of the SAT+CAS solutions were verified to be nonisomorphic and generated all of the SAT solutions.

## Solving Lam's problem

SAT instances are generated for each of the starting structures produced by codewords of weights 15, 16, and 19.

The instances have no solutions. The SAT solver produces nonexistence certificates which are verified by a proof verifier.

The certificates total about 110 TiB and were computed in about 690 days of computing time on 2.1 GHz CPU cores. The 2011 verification of Roy used about 1700 days on 2.4 GHz CPU cores.

## 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`