Unsatisfiability Proofs for Weight 16 Codewords in Lam's Problem

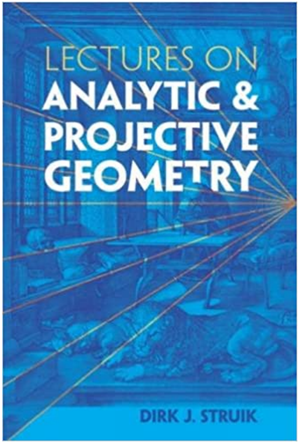
Curtis Bright
University of Windsor

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

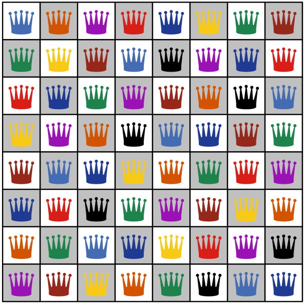
International Joint Conference on Artificial Intelligence
January, 2021

Overview

We generate computer-verifiable proofs solving Lam's problem from projective geometry.



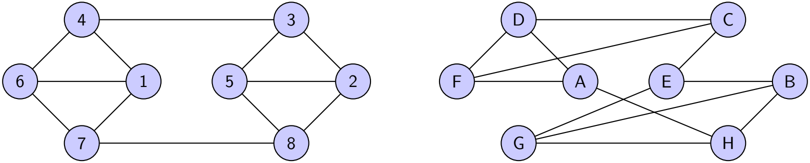
Effectiveness of SAT solvers



Problem: *How many colours are necessary if each set of queens of the same colour must be mutually non-attacking?*

A SAT solver can quickly return a 9-colouring and **provide a certificate** that there are no 8-colourings.

Effectiveness of computer algebra systems (CASs)



Problem: *Are these graphs isomorphic?*

A CAS can quickly return **yes** and provide an explicit isomorphism:

$1 \leftrightarrow$ D	$3 \leftrightarrow$ E	$5 \leftrightarrow$ G	$7 \leftrightarrow$ A
$2 \leftrightarrow$ B	$4 \leftrightarrow$ C	$6 \leftrightarrow$ F	$8 \leftrightarrow$ H

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.

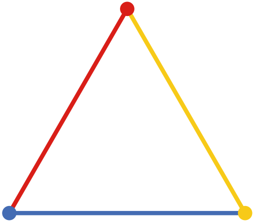
Projective geometry: History

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

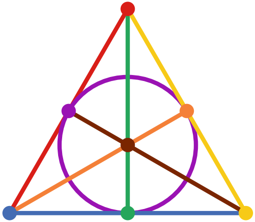
The existence of projective geometries shows this is impossible! These geometries satisfy a "projective axiom" that any two lines meet in a unique point.

Finite projective planes

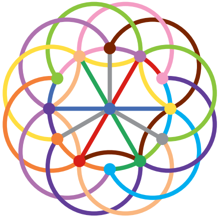
- Every pair of lines meet at a unique point.
- Every pair of points define a unique line.
- Every line contains $n+1$ points for some *order* n.



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

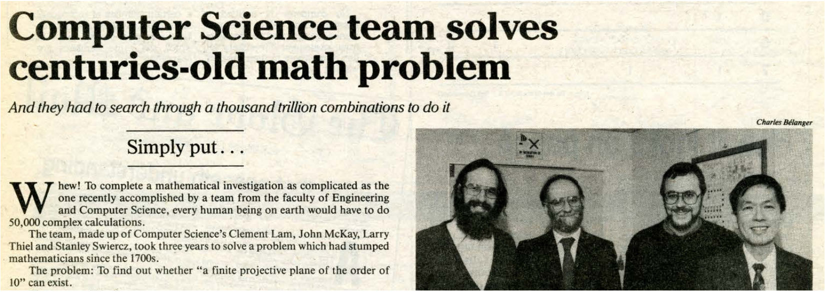
Projective planes of small orders

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

Projective planes of small orders

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



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.

Charles Bélanger

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.

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.

The encoding scripts and a selection of the certificates are available at `uwaterloo.ca/mathcheck`.

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 (representing lines) contains exactly $n + 1$ ones.

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

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.

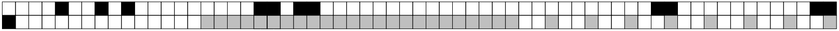
How should the projective axiom be encoded in Boolean logic?

Any two lines meet exactly once, therefore:

1. Any two lines meet *at most* once.
2. Any two lines meet *at least* once.

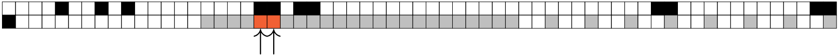
1. Lines meet at most once

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



1. Lines meet at most once

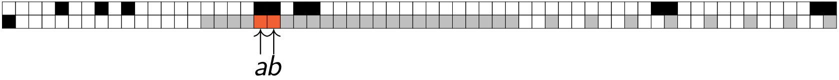
Consider the following two lines (grey entries unknown) as an example:



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:



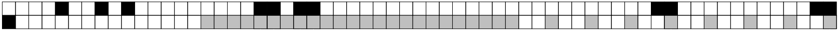
The highlighted entries cannot both be true or the lines would meet more than once.

In Boolean logic:

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

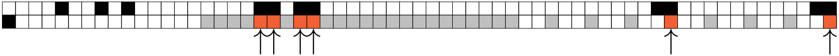
2. Lines meet at least once

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



2. Lines meet at least once

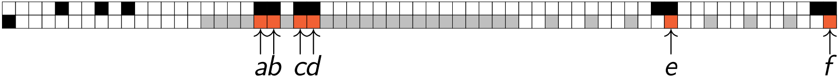
Consider the following two lines (grey entries unknown) as an example:



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.

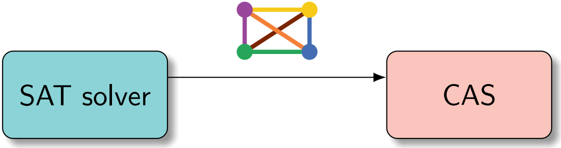
In Boolean logic:

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

Isomorphism Blocking

CAS isomorphism blocking

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

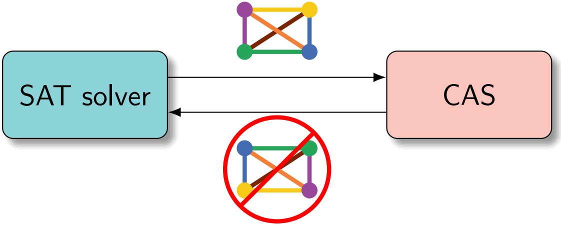


SAT solver

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

The structure of the incidence matrix can essentially be split up into three cases, each resulting in a number of SAT instances.

All of the cases are unsatisfiable and we generated nonexistence certificates for each:

Case	Compute time	Certificate size	Appearing
1	7 seconds	35 MiB	AAECC 2020
2	30 hours	325 GiB	IJCAI 2020
3	24 months	110 TiB	AAAI 2021

In 2011, a verification of case 2 required 16,000 hours.

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`