# Solving Lam's Problem via SAT and Isomorph-Free Exhaustive Generation

Curtis Bright
University of Windsor

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

KR 2022
August 3, 2022

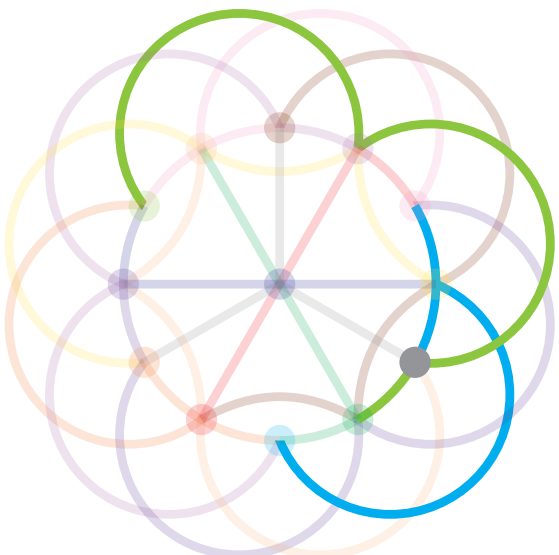
# Lam's Problem

***Lam's problem*** is to show the nonexistence of a type of geometric structure that has intrigued mathematicians for centuries.

This is a projective plane of order three.

# Lam's Problem

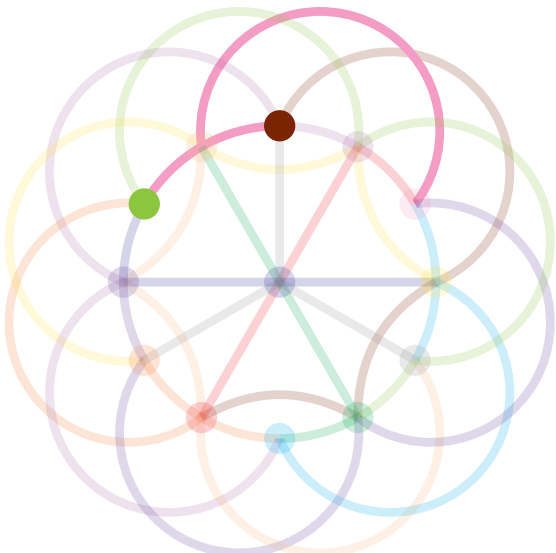
***Lam's problem*** is to show the nonexistence of a type of geometric structure that has intrigued mathematicians for centuries.



Any two lines meet in a unique point.

# Lam's Problem

***Lam's problem*** is to show the nonexistence of a type of geometric structure that has intrigued mathematicians for centuries.



Any two points lie on a unique line.

# Lam's Problem

***Lam's problem*** is to show the nonexistence of a type of geometric structure that has intrigued mathematicians for centuries.

Is it possible to find a structure with these properties but has 111 points and 111 lines?

# Twentieth Century Resolution

Lam's problem was resolved in 1989 using custom-written programs running on a CRAY-1A supercomputer for months.

## 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

Can we ***trust*** a result without formal verification?

# First Verifiable Resolution

We provide the first ***verifiable*** resolution of Lam's problem.

This is accomplished by reducing the problem to an instance of the Boolean satisfiability (SAT) problem. A SAT solver generates a proof certificate which is verified.

To reduce the search space size it was necessary to augment the SAT solver with an isomorph-free exhaustive generation method.

# Reduction to SAT

A Boolean variable is used for every possible point-line incidence (e.g., $a_{ij}$ represents point $i$ is on line $j$).

Constraints specifying the ***projective plane axioms*** are encoded using these variables.

For example, points 1 and 2 cannot lie on both line 1 and line 2 simultaneously. In Boolean logic this is

$$(a_{11} \wedge a_{21}) \rightarrow \neg(a_{12} \wedge a_{22}).$$

# Symmetry Breaking

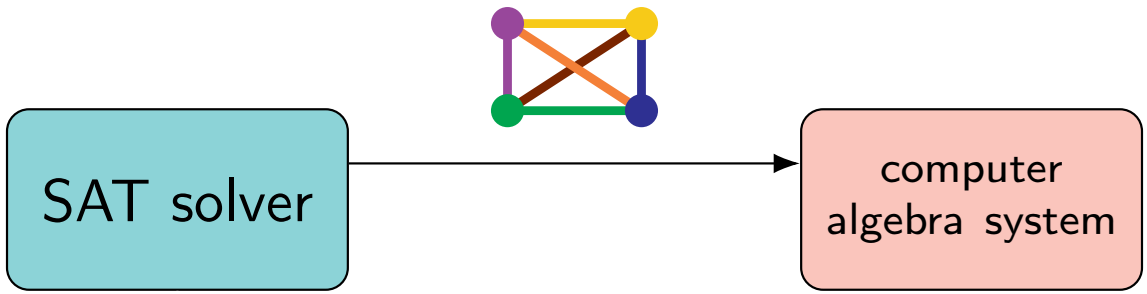
We enforce a lexicographic order on the lines and points of the plane when viewed as binary row and column vectors.

This ***breaks*** row and column symmetries, but many symmetries remain which are not easy to break with static constraints.

We combine the "recorded objects" approach to isomorph-free exhaustive generation with our SAT solver.

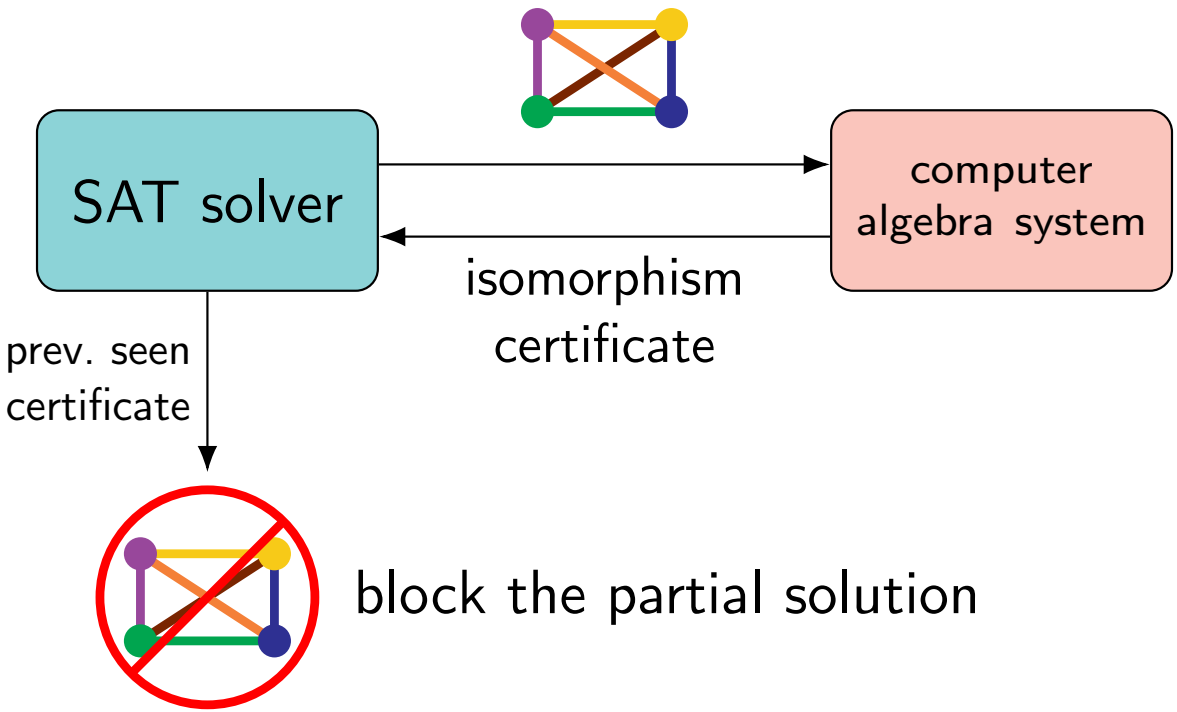
# Isomorph-free Generation in SAT

During the search the SAT solver will find ***partial solutions*** by finding complete definitions for the first few lines of the plane.



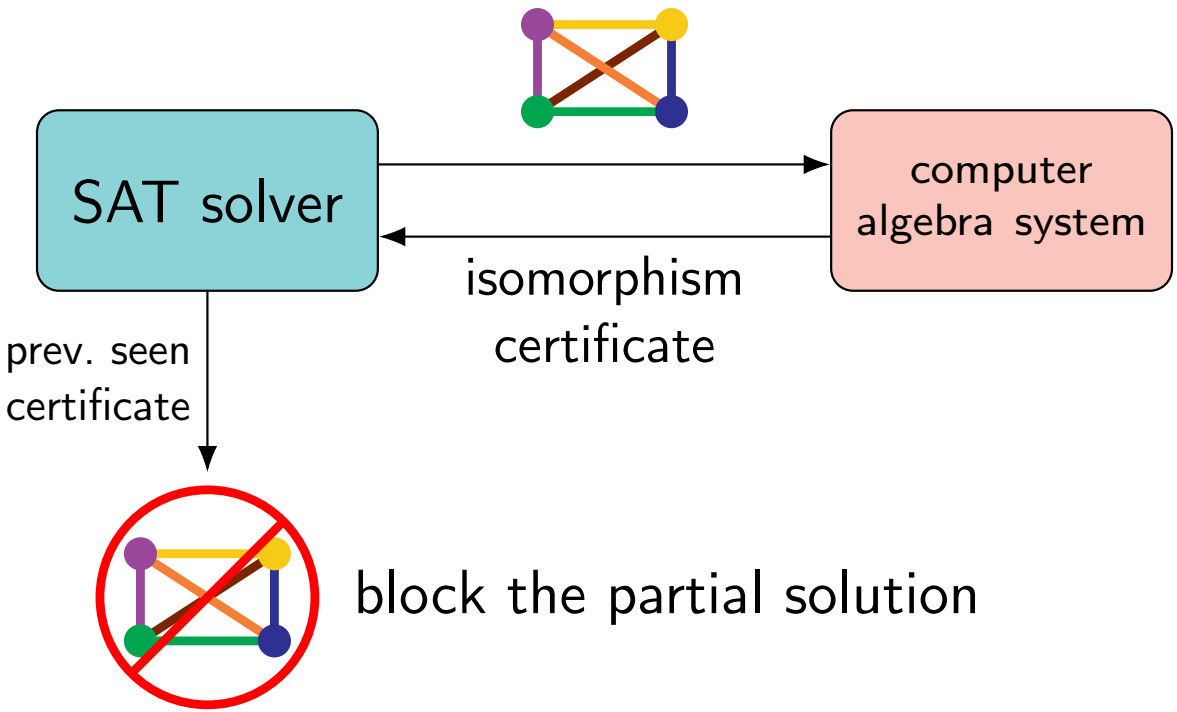
# Isomorph-free Generation in SAT

During the search the SAT solver will find ***partial solutions*** by finding complete definitions for the first few lines of the plane.



# Isomorph-free Generation in SAT

During the search the SAT solver will find ***partial solutions*** by finding complete definitions for the first few lines of the plane.




This method to enumerate all possibilities for the first 19 points is ***150 times*** faster than isomorphism removal at the end.

# Verification of Lam's Resolution

We verified Lam's resolution using the certificates provided by the SAT solver and the computer algebra system.

There has been only a single independent verification of Lam's original search (Roy 2011). Our verification found ***missing cases*** in both the original search and its independent verification.

# Conclusion

Isomorph-free exhaustive generation methods have not typically been exploited by SAT solvers.

In practice, using isomorph-free exhaustive generation inside a SAT solver can speed up searches for many combinatorial objects by an ***exponential factor*** in the object size.

`curtisbright.com`