A SAT Solver + Computer Algebra Attack on the Minimum Kochen–Specker Problem

Zhengyu (Brian) Li[1], Curtis Bright[2], Vijay Ganesh[1]

[1]Georgia Institute of Technology, USA
[2]University of Windsor, Canada

Meet the Team

Zhengyu Li

PhD Student in Computer Science
Georgia Institute of Technology

Curtis Bright

Assistant Professor
University of Windsor
School of Computer Science

Vijay Ganesh

Professor
Georgia Institute of Technology
School of Computer Science

The (3D) Kochen–Specker Theorem

The KS theorem states that quantum mechanics is in conflict with classical models: the result of a measurement does not depend on which other compatible measurements are performed simultaneously.

There is a finite set $S \subset \mathbb{R}^3$ such that there is no function $f : S \to \{0, 1\}$ satisfying

$$f(u) + f(v) + f(w) = 1$$

for all triples (u, v, w) of mutually orthogonal vectors in S.

In order to prove their theorem, Kochen and Specker establish the existence of a KS vector system.

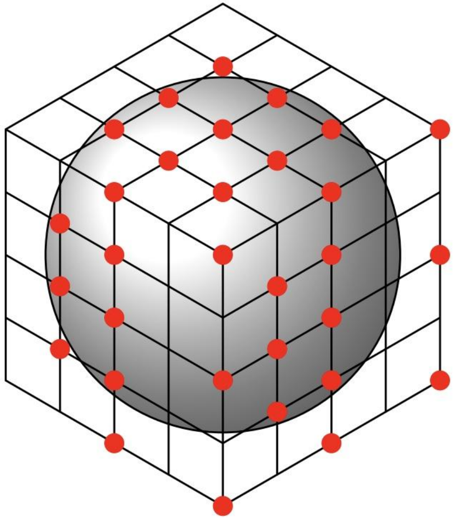
The (3D) Kochen–Specker Set

A set of vectors in 3-dimensional space is a KS set if there is no 01-assignment that satisfies the following two conditions:

1. Two mutually exclusive events (orthogonal vectors) cannot both have the value 1.
2. In a complete context (3 mutually orthogonal vectors) exactly one assignment (vector) has the value 1.

The Minimum KS Problem - Can we do better?

Authors	Year	Bound
Kochen, Specker	1967	≤ 117
Jost	1976	≤ 109
Conway, Kochen	1990	≤ 31
Arends, Ouaknine, Wampler	2009	≥ 18
Uijlen, Westerbaan	2016	≥ 22
Li, Bright, Ganesh	2022	≥ 23
Li, Bright, Ganesh / Kirchweger, Peitl, Szeider	2023	≥ 24



It is unknown if there exists a KS vector system with less than 31 vectors.

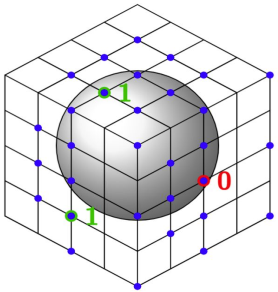
Encoding the KS Problem as a Combinatorial Problem

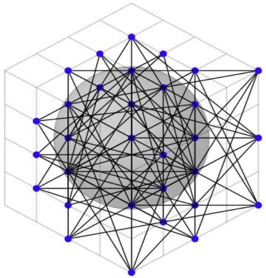
To find a KS set, we want to find graphs *G* such that

- *G* is non-101-colorable: *G* has no possible 101-coloring
- *G* is embeddable: *G* is an orthogonality graph for a 3D vector system

In addition, previous research has proven that *G* for a minimal KS set satisfies

- Squarefree Constraint: *G* must not contain a square subgraph
- Minimum Degree Constraint: every vertex of *G* must have minimum degree 3
- Triangle Constraint: every vertex is part of at least one triangle subgraph



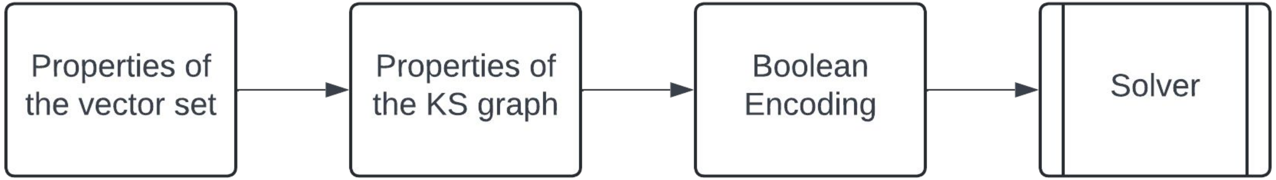


```
p cnf 40 210
-1 -4 -3 -6 0
-2 -4 -3 -5 0
-1 -2 -5 -6 0
-1 -7 -3 -9 0
-2 -7 -3 -8 0
```

Computational Search for the KS sets

We want a computational tool that is:

- Scalable to large combinatorial objects
- Allow custom constraints as input
- Can be formally verified



Computer Algebra Systems (CASs)

Maple™

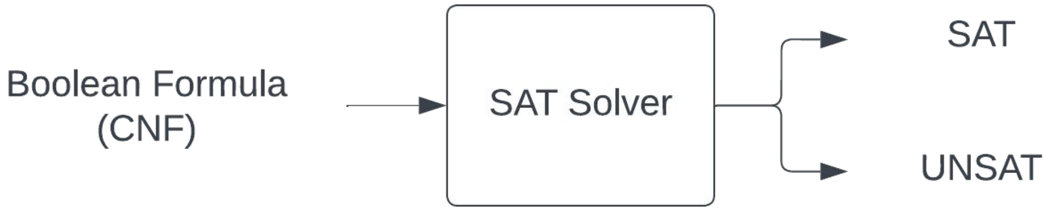
sage

Wolfram Mathematica®

nauty and Traces

Brendan McKay and Adolfo Piperno

Satisfiability (SAT) Solver



A SAT solver is a computer program which solves the Boolean satisfiability problem. It takes a Boolean formula in conjunctive normal form (CNF) as input, and returns

- SAT if it finds a variable assignment that satisfies the input formula
- UNSAT if it can demonstrate that no such assignments exist

Boolean satisfiability is NP-complete, but SAT solvers are effective for many applications.

Motivations of SAT+CAS

- SAT solvers are great at solving search problems specified by simple constraints (clauses).
- Computer algebra systems (CASs) are great at many sophisticated mathematical problems involving little search.
- Problems involving both sophisticated mathematics and search are good candidates for a SAT+CAS approach. (developed in 2015 by Zulkoski, Ganesh, and Czarnecki and independently by Erika Ábrahám)

SAT + CAS = efficient search + mathematical knowledge

Zulkoski, E., Ganesh, V., Czarnecki, K.: MathCheck: A Math Assistant via a Combination of Computer Algebra Systems and SAT Solvers. Springer, Cham (2015).
Ábrahám, E.: Building bridges between symbolic computation and satisfiability checking (2015).

University of Windsor

Georgia Tech

An Emerging Paradigm

There has been a lot of research in recent years involving SAT and computer algebra or related methods.

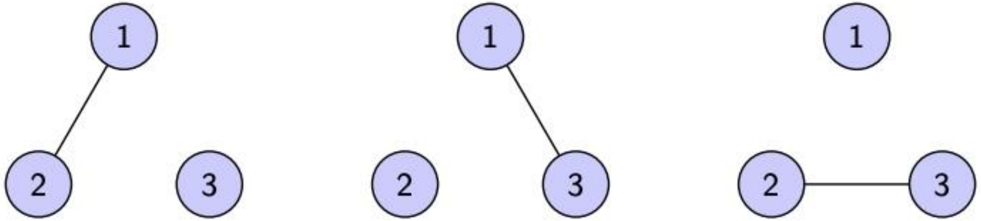
A small and incomplete sample:

- Verification of Ramsey numbers (Duggan, Li, Bright, Ganesh 2024).
- A SAT-based Resolution of Lam's Problem (Bright et al. 2021).
- A Hybrid SAT and Lattice Reduction Approach for Integer Factorization (Ajani, Bright 2023).
- Proving the correctness of multiplier circuits (Kaufmann, Biere 2020).
- Finding new algorithms for 3×3 matrix multiplication (Heule, Kauers, Seidl 2021).
- SAT modulo symmetries for generating combinatorial objects in an isomorph-free way (Kirchweger et al. 2021)
- Making progress on conjectures in geometric group theory (Savela, Oikarinen, Jarvisalo 2020).
- Computing directed Ramsey numbers (Neiman, Mackey, Heule 2020).
- Debugging of digital circuits (Mahzoon, Große, Drechsler 2018).

University of Windsor Georgia Tech

Isomorphism

When generating combinatorial objects we really only care about generating them up to isomorphism. Unfortunately, objects usually have many isomorphic representations.



$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

The Importance of Isomorph-free Generation

For example, a graph with n vertices can have up to $n!$ distinct isomorphic adjacency matrices. This makes the size of the search space for graphs much larger than it needs to be.

To exhaustively generate combinatorial objects it is of utmost importance to detect and remove isomorphic copies of objects as early as possible.

Isomorph-free Orderly Generation

When generating combinatorial objects we only care about generating them up to isomorphism.

The notion of canonicity is defined so that:

- Every isomorphism class has exactly one canonical representative.
- If an adjacency matrix is canonical then its upper-left submatrix of any size is also canonical.

Developed independently by Faradžev and Read in 1978.

University of Windsor

Georgia Tech

Canonicity Examples

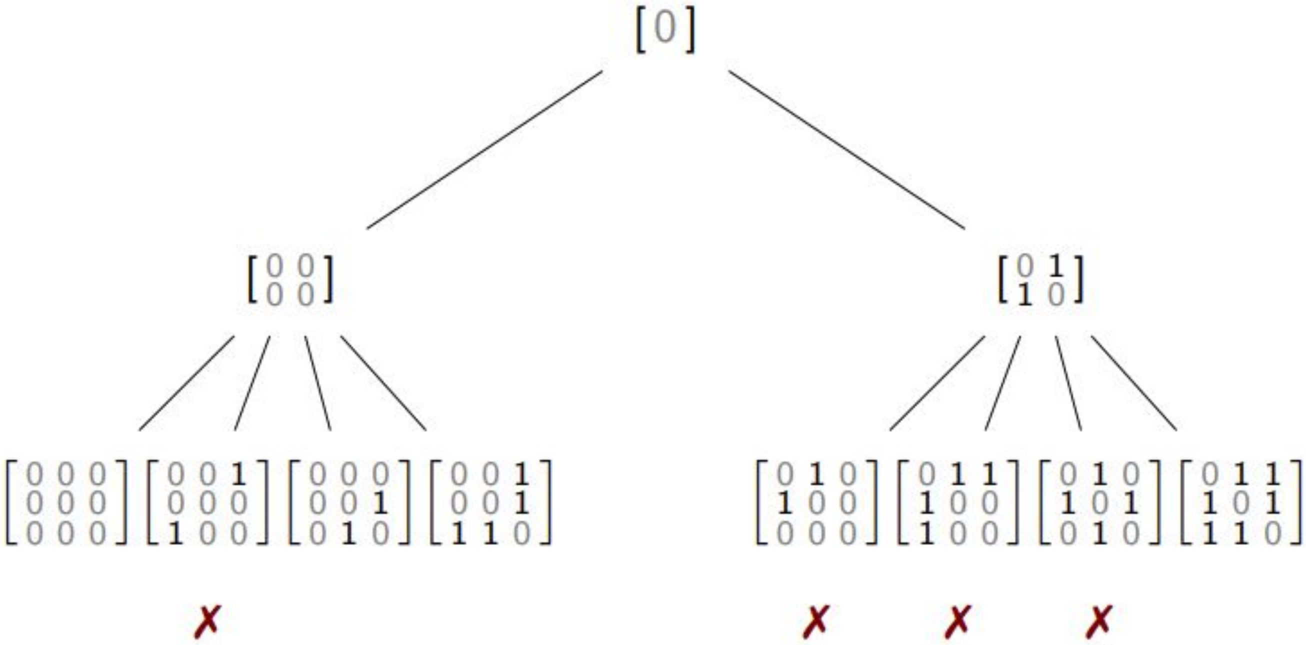
An adjacency matrix is canonical if its “vector representation” is lex-minimal among all matrices in the same isomorphism class.

For example,

Adj. matrix	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$
Vector rep.	$[1\ 0\ 0] >_{\text{lex}}$	$[0\ 1\ 0] >_{\text{lex}}$	$[0\ 0\ 1]$
Canonical?	✗	✗	✓

are isomorphic adjacency matrices but only the last is canonical.

Orderly Generation of Graphs

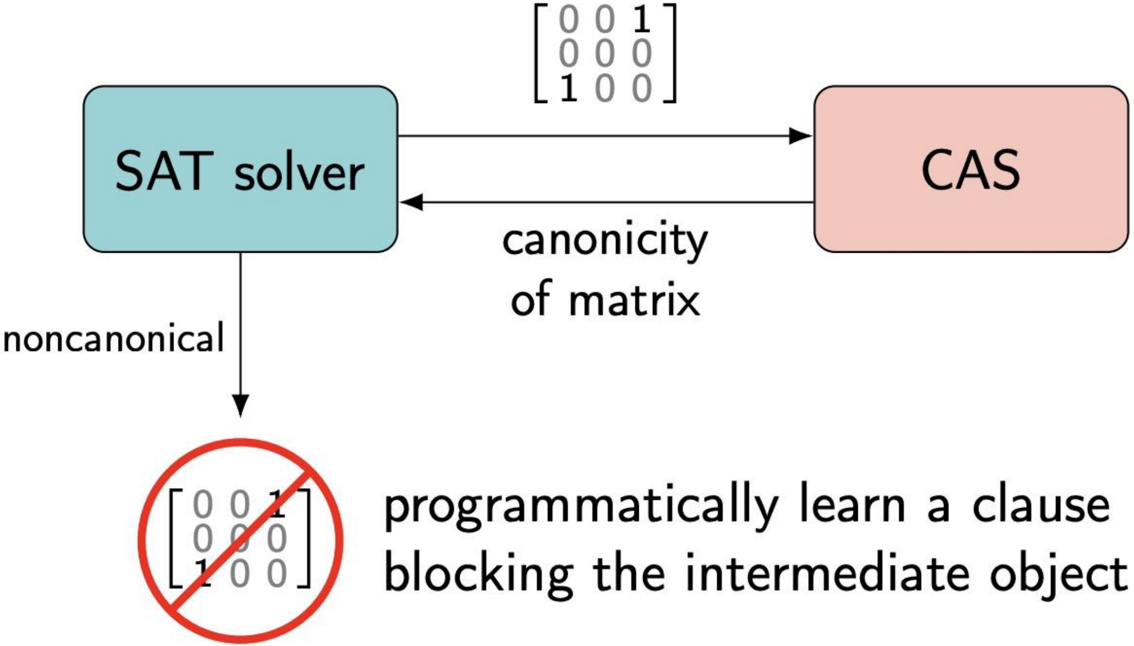


Implementing Orderly Generation

To perform orderly generation we need a canonicity checking method which is a difficult problem.

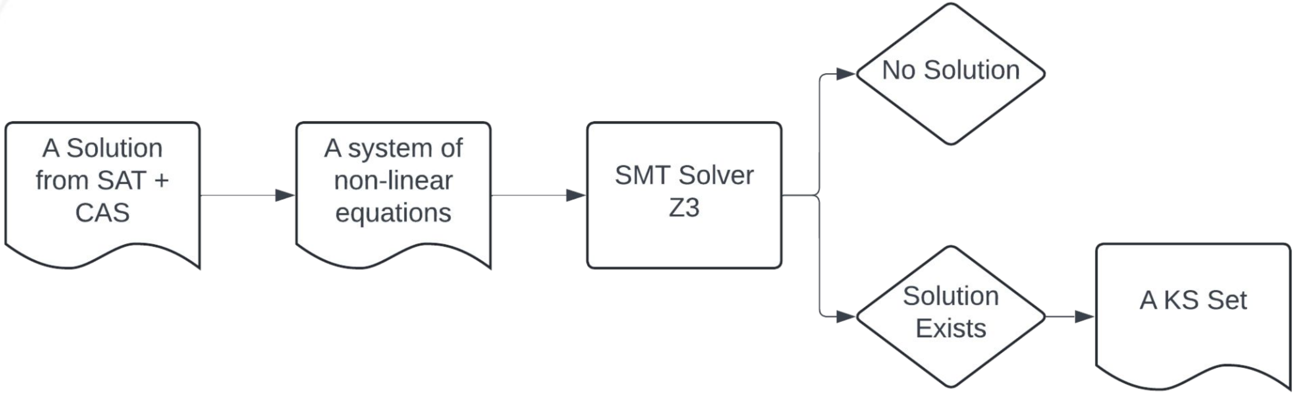
However, verifying that a matrix is noncanonical is often easy—it requires finding a single permutation of the vertices which gives a lexicographically smaller adjacency matrix.

Implementing Orderly Generation

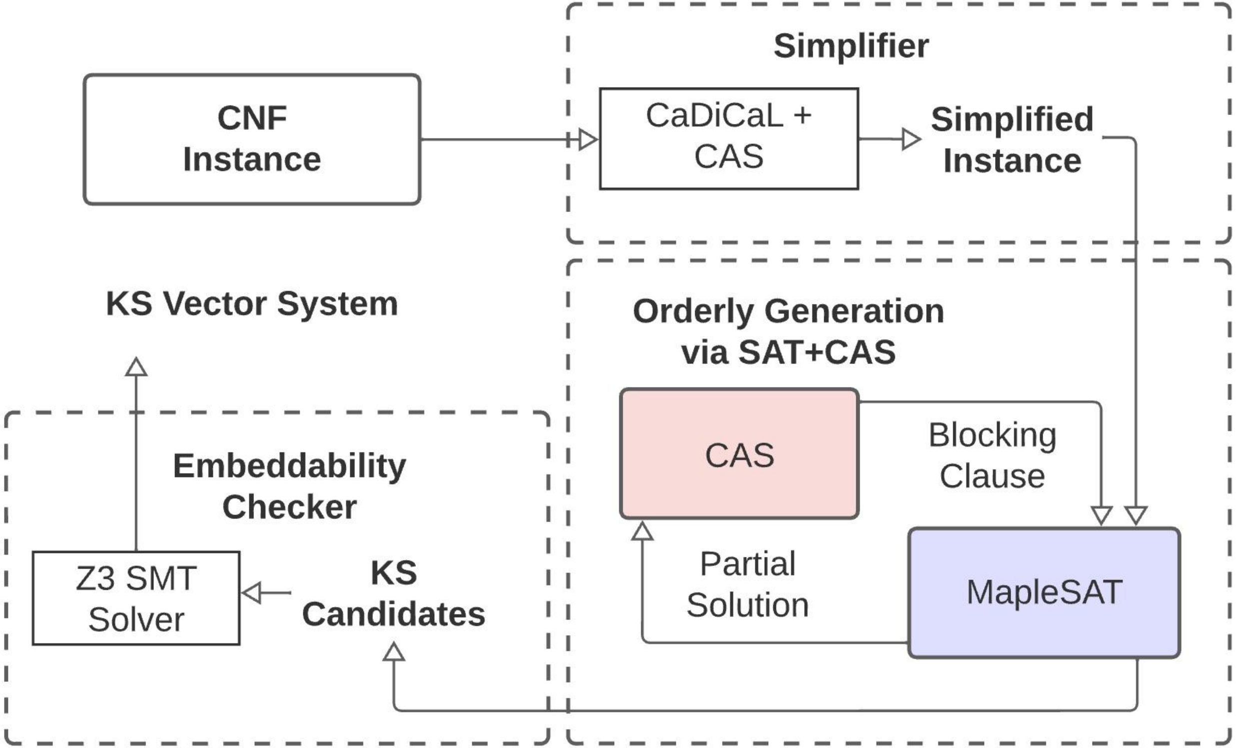


Embeddability Checking

- A solution found by the SAT solver can only be a KS vector system if it is embeddable (there is a vector system that corresponds to this graph).
- We use SMT Solver Z3 to check for embeddability.
- We precompute minimal unembeddable graphs up to order 12, and block solutions that contain such graphs dynamically during solving.



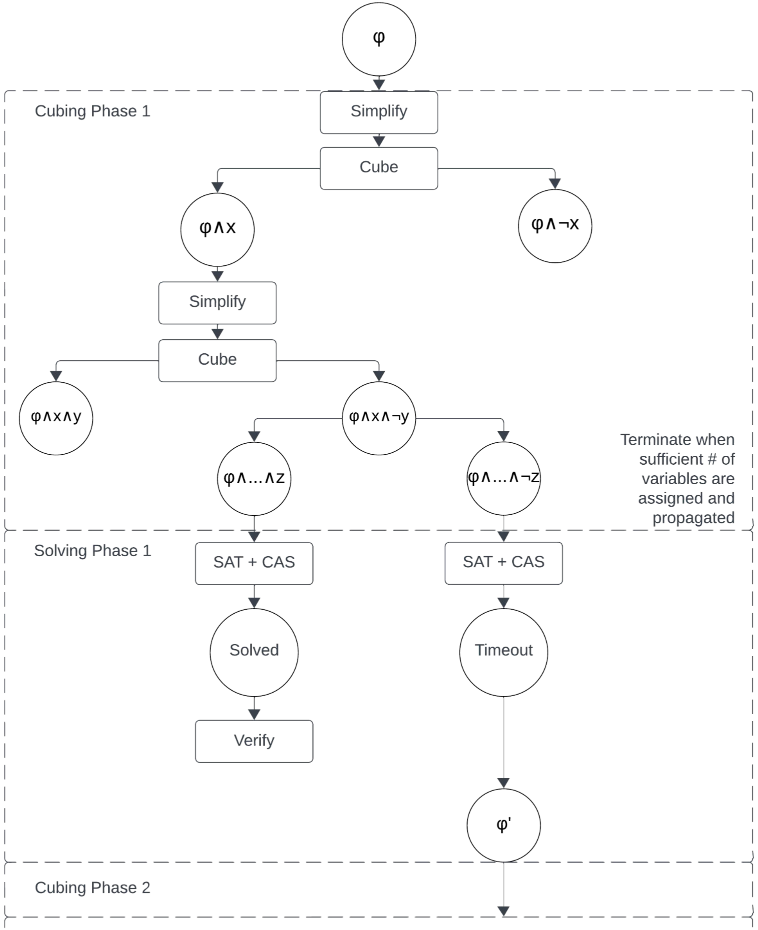
Pipeline Overview



Parallelization

We use a novel Monte Carlo Tree Search (MCTS) based Cube-and-Conquer (CnC) technique to divide the instance into smaller subproblems.

Each subproblem is solved until the proof size exceed 7 GB; then it will be divided into smaller subproblems.



Jha, Piyush, et al. "AlphaMapleSAT: An MCTS-based Cube-and-Conquer SAT Solver for Hard Combinatorial Problems."

University of Windsor

Georgia Tech

Verification

SAT: We have enabled DRAT proof logging in the SAT solver so that certificates are generated.

CAS: A CAS-derived permutation provides a witness that any blocked matrix is noncanonical.

We have certified all results up to order 23 and the uncompressed proofs are over 40 TB in order 23.

Results

Order	SAT+CAS	SAT	CAS	Method
17	0.3 mins	9.0 mins	25.2 mins	Sequential
18	1.8 mins	266.4 mins	455.4 mins	Sequential
19	9.0 mins	11,705.8 mins	9,506.4 mins	Sequential
20	140.5 mins	timeout	timeout	Sequential
21	1,945 mins	timeout	timeout	Sequential
22	932 hours	timeout	timeout	Parallel
23	12,116 hours	timeout	timeout	Parallel

Future Work

- Improve the lower bound of the extended KS set (which requires vectors not explicitly needed to show a 01-valuation, but are needed experimentally.
- Search for KS sets of order 24 and above.

A Promising Future!

- SAT and CAS deserve to be combined, and there should be more work pursuing this idea.
- Many problems in quantum foundations are combinatorial, and we look forward to applying SAT+CAS to more problems in the future.

https://uwaterloo.ca/mathcheck/

Conclusions

- SAT + CAS is a state-of-the-art tool to solve large combinatorial problems.
- AlphaMapleSAT is an efficient tool to perform cube-and-conquer and parallelize the problem.
- Verification is of utmost importance and can be performed using SAT + CAS.