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

Zhengyu Li[1], Curtis Bright[2], Vijay Ganesh[1]
[1]Georgia Institute of Technology, Atlanta, GA, USA
[2]University of Windsor, Windsor, ON, Canada

The Kochen-Specker Problem

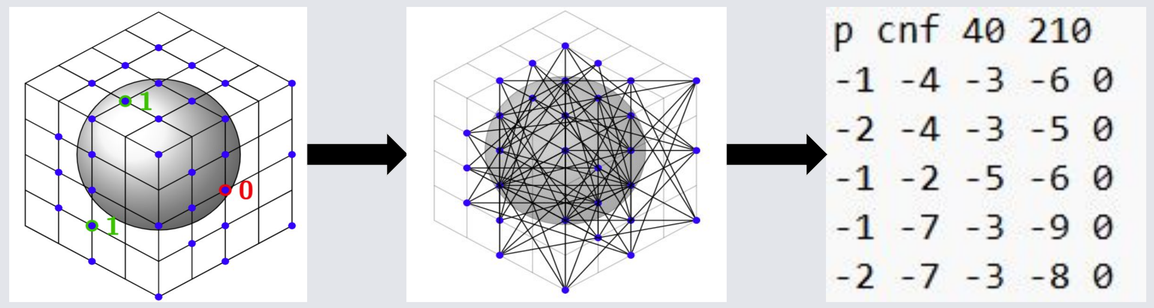
Finding the minimum size of a Kochen-Specker (KS) vector system has been an open problem in quantum foundations for over 50 years.

We obtain a tighter lower bound with four orders of magnitude speedup over previous methods, while providing a formal proof.

This is a challenging problem as we are searching over graphs with up to 23 vertices, which is a total of 2^{253} graphs.

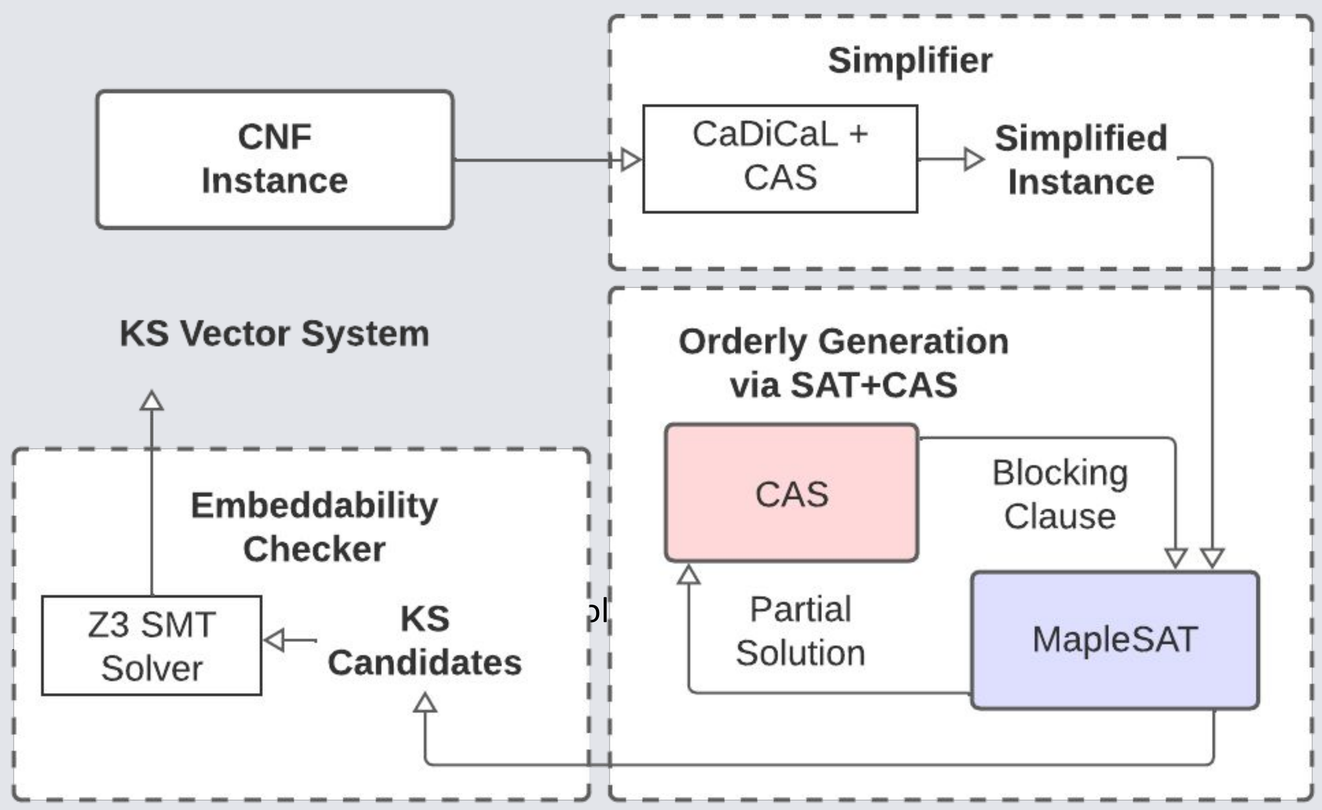
SAT Encoding

We encode properties of the KS graph into conjunctive normal form (CNF), so that any satisfying assignment of the formula correspond to a KS graph.



MathCheck: Isomorph-free Graph Enumeration with Formal Proof

Can SAT Solver + Computer Algebra System solve a long-standing problem in quantum mechanics and provide a formal proof?



Results

n	SAT+CAS	SMS	CAS-only	SAT-only
17	0.3 m	0.2 m	25.2 m	9.0 m
18	1.8 m	1.2 m	455.4 m	266.4 m
19	9.0 m	8.4 m	9506.4 m	11,705.8 m
20	140.5 m	100.8 m	timeout	timeout
21	1945.0 m	1574.4 m	timeout	timeout

Table 2: **SAT+CAS vs. SMS, CAS-only (nauty), and SAT-only:** The total CPU time (in minutes) compared on orders $17 \leq n \leq 21$. All tools are sequential. For higher orders, CAS-only and SAT-only time out at 12,000 minutes.

	Number of Cubes		Total CPU time	
n	SAT+CAS	SMS	SAT+CAS	SMS
22	26,646	18,659	932 h	628 h
23	173,097	313,665	12,116 h	11,922 h

Table 3: **Parallel CnC SAT+CAS vs. parallel SMS:** The number of cubes and total CPU time for the parallel versions of SAT+CAS and the SAT Modulo Symmetries tool on orders $22 \leq n \leq 23$.

Certificate & Formal Proof

Verifying SAT: The SAT solver generates a DRAT proof, which is verified by a third-party proof verifier. This checks that each learned clause can be derived from previous clauses using simple, logically consistent rules.

Verifying CAS: CAS-derived clauses (noncanonical blocking and unembeddable subgraph blocking) are specially tagged in the DRAT proof. These are verified separately using a Python script that applies CAS-derived permutations to confirm the blocked matrices are indeed noncanonical or unembeddable, providing independent certification of the CAS results.

Future Work

The SAT + CAS approach is very general, and we are excited to leverage this approach on more problems in combinatorics and graph theory.

Our Paper