# A SAT + Computer Algebra System Verification of the Ramsey Problem $R(3,8)$ (Student Abstract)

Conor Duggan, Zhengyu Li, Curtis Bright, and Vijay Ganesh

**Introduction** The Ramsey problem $R(3,8)$ asks for the smallest $n$ such that every red/blue coloring of the complete graph on $n$ vertices must contain either a blue triangle or a red 8-clique.

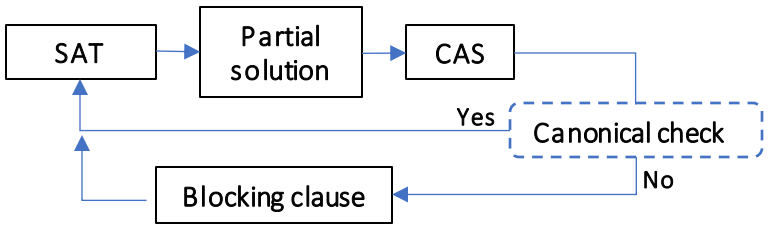
We provide the **first certifiable proof** that $R(3,8) = 28$, automatically generated by a combination of **Boolean satisfiability (SAT) solver** and a **computer algebra system (CAS)**.

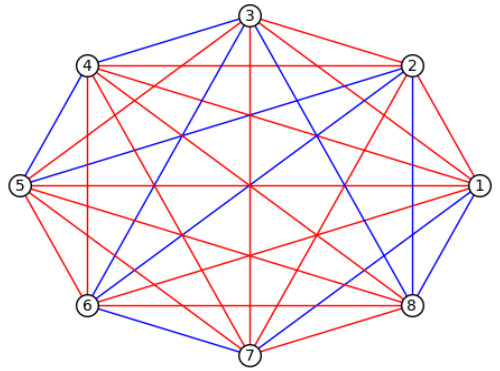
This SAT+CAS combination is significantly faster than a SAT-only approach. While the $R(3,8)$ problem was first computationally solved by McKay and Min in 1992, it was not **formally verified**.

We applied additional SAT encodings:

- Partial static symmetry breaking clauses enforces a lexicographic ordering on the rows of the adjacency matrix.
- The degree of each vertex is in between 5 and 7 inclusive.

**SAT + CAS** We use a CAS to generate blocking clauses through orderly generation, which are returned to the SAT solver. The technique can achieve **perfect symmetry breaking** and **isomorph-free combinatorial generation**.



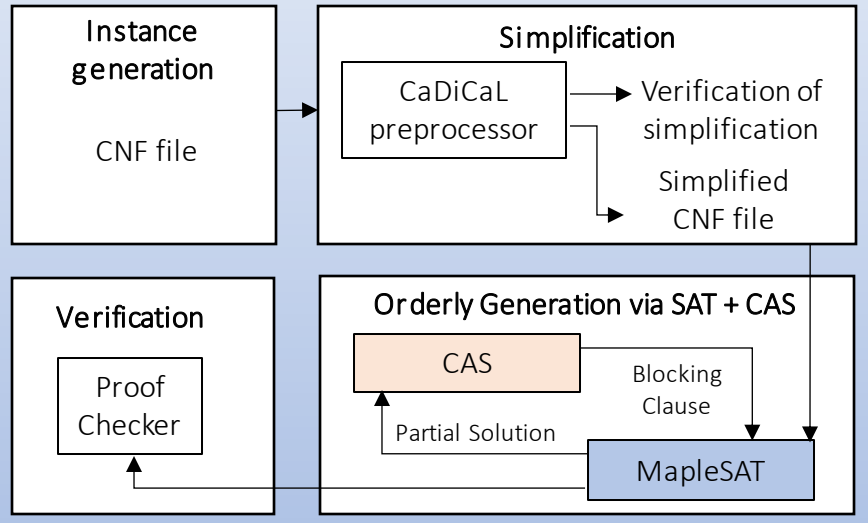

A red/blue edge coloring on 8 vertices without a blue 3-clique or red 4-clique, showing $R(3,4) > 8$.

| $R(3,7)$ ablation study | | |
|---|---|---|
| **Method** | **Encoding** | **Solving Time** |
| SAT | Standard | > 86,400s |
| SAT+CAS | Standard | 162s |
| SAT+CAS | Full | 32s |

Ablation study on $R(3,7)$ on 23 vertices performed using Intel E5-2683 processor running at 2.10 GHz with 4GB of RAM

**Result** $R(3,8)$ on 28 vertices was found to be UNSAT using MapleSAT combined with a CAS, after 96 hours (AMD EPYC 7763 2.45 GHz processor with 16 GiB of RAM).

A 30 GiB DRAT file was generated and verified in 63 hours (Intel Xeon CPU E5-2673 v3 processor running at 2.40 GHz with 28 GiB of RAM).




**Verification**

- This was performed using the *DRAT-trim* proof-checker (Wetzler, Heule, and Hunt Jr 2014) slightly modified to support the addition of trusted clauses.
- The proof checker verifies that each clause can be derived from the previous clauses via resolution. The CAS-derived blocking clauses are verified by evidencing that each clause blocks graphs whose adjacency matrices are not canonical.