Verified Certificates via SAT & Computer Algebra Systems for the Ramsey R(3,8) and R(3,9) Problems

Zhengyu Li[1], Conor Duggan[2], Curtis Bright[3], Vijay Ganesh[1]

[1]Georgia Institute of Technology, USA

[2]University of Waterloo, Canada

[3]University of Windsor, Canada

The Ramsey Problem

The Ramsey number R(p,q) solves a classic problem: What's the smallest party size needed to guarantee that either p people all know each other, or q people are all strangers to each other?

In graph theory terms, R(p,q) is the smallest number of vertices where every 2-coloring of the edges either contains p vertices all connected by the first color or q vertices all connected by the second color.

SAT Encoding

Edge Variables: Each edge e is a Boolean variable

e = True → edge colored blue

e = False → edge colored red

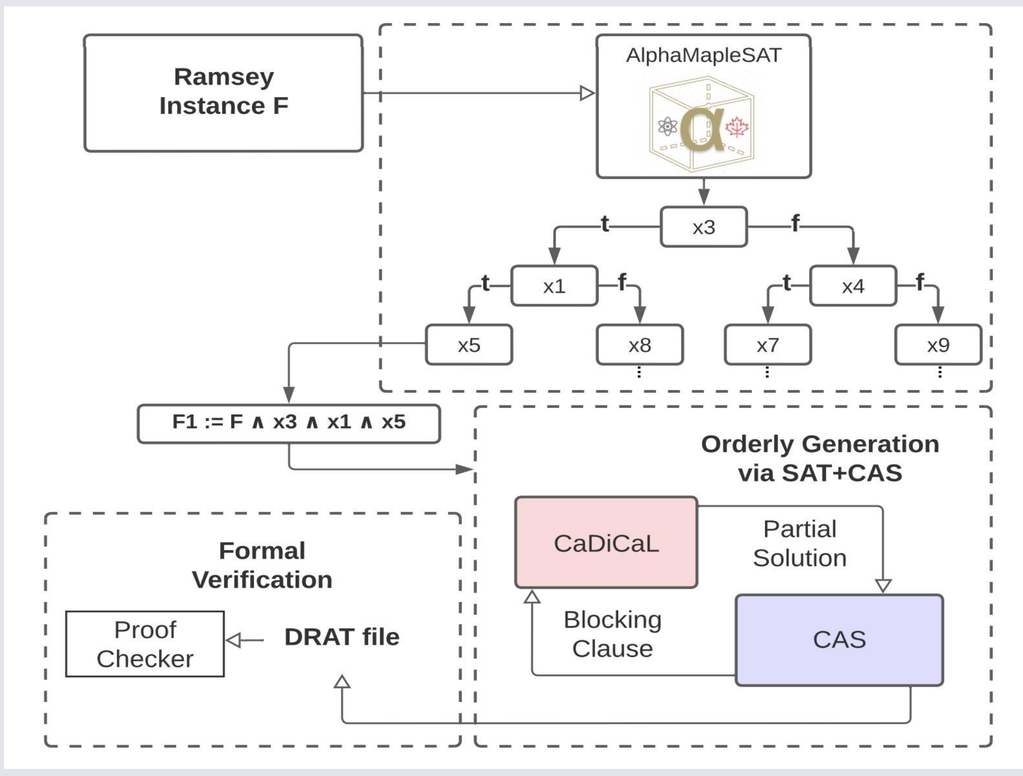
$$\bigwedge_{K_p \subseteq K_n} \bigvee_{e \in K_p} \neg e \quad \text{and} \quad \bigwedge_{K_q \subseteq K_n} \bigvee_{e \in K_q} e$$

Every p-clique must have ≥1 red edge

Every q-clique must have ≥1 blue edge

MathCheck: Isomorph-free Graph Enumeration with Formal Proof

Can SAT Solver + Computer Algebra System verify a long-standing problem in graph theory and provide a formal proof?



Flowchart of the parallelized MathCheck tool. AlphaMapleSAT is used as the cubing solver, and CaDiCaL + CAS is used as the conquering solver.

Results

We successfully verified R(3,8) = 28 and R(3,9) = 36 using SAT+CAS, providing the first independently verifiable certificates for these Ramsey numbers. Our approach achieved orders-of-magnitude speedups over SAT-only methods: solving R(8,3) in 59 hours vs. a 7 day timeout, and R(9,3) in 26 hours using parallelized cube-and-conquer.

	CaDiCaL + CAS		CaDiCaL only	
k	$R(3,k)$	$R(k,3)$	$R(3,k)$	$R(k,3)$
7	14.3 s	8.2 s	564.3 s	220.7 s
8	112.1 h	18.5 h	> 7 days	> 7 days

Table 2: Comparison of sequential runtime for instances $R(3,k)$ and $R(k,3)$ with $k = 7$ and 8. "CaDiCaL + CAS" indicates solutions using CaDiCaL with CAS, while "CaDiCaL only" uses CaDiCaL without CAS. Cardinality constraints are excluded for $k = 7$ to avoid making the instance too easy, but included for $k = 8$. Experiments were conducted on Dual Xeon Gold 6226 processors running at 2.70 GHz.

Instance	Cubing Time	Simplification Time	Solving Time	Verification Time	Wall Clock Time
$R(8,3)$	1,360 s	1,217 s	19,811 s	22,328 s	8 hrs
$R(9,3)$	15,530 s	42,482 s	697,575 s	473,874 s	26 hrs

Table 3: Summary of experimental results for solving $R(8,3)$ and $R(9,3)$ in parallel.

Certificate & Formal Proof

Verifying SAT: DRAT proof logging enabled to generate certificates for all SAT solver decisions. DRAT-trim checker (modified) verifies proof correctness - only need to trust this simple verifier, not the complex SAT solver. CAS-derived clauses marked with 't' prefix are "trusted" clauses in DRAT proof.

Verifying CAS: Witness-based approach where a CAS provides permutation witnesses that prove "non-canonical" partial solutions can be discarded without loss of generality. An independent Python script applies witness permutations to verify each blocked matrix produces lexicographically smaller result. No trust in CAS required - only verify the witness works, not the CAS computation itself.

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