**Satisfiability: Theory, Practice, and Beyond**

April-May 2023

**Simons Institute for the Theory of Computing**

# Isomorph-Free Generation of Combinatorial Objects with SAT Modulo Symmetries

**Stefan Szeider**

**TU Wien, Vienna, Austria**

ALGORITHMS AND COMPLEXITY GROUP

SIMONS INSTITUTE for the Theory of Computing

**joint work with Katalin Fazekas, Markus Kirchweger, Tomas Peitl, and Manfred Scheucher**

FWF Der Wissenschaftsfonds.

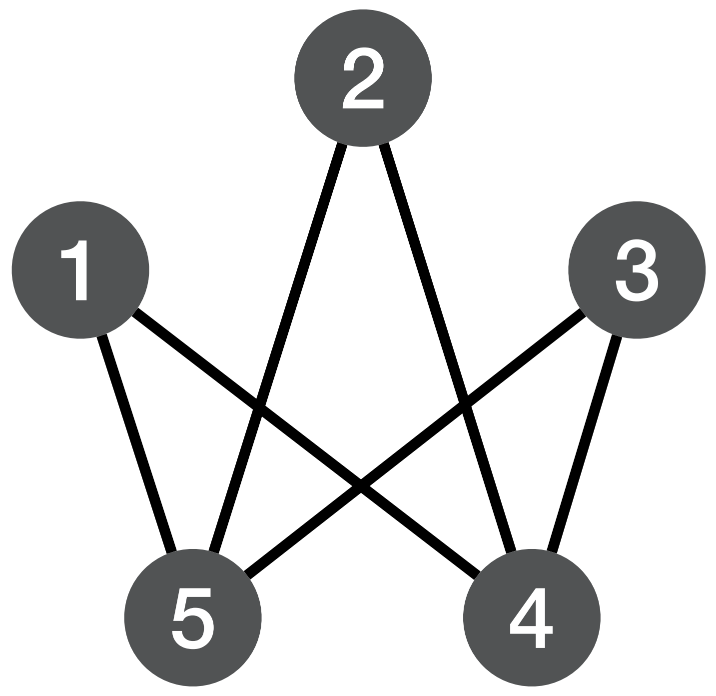
WWTF VIENNA SCIENCE AND TECHNOLOGY FUND

# Generation of Combinatorial Objects

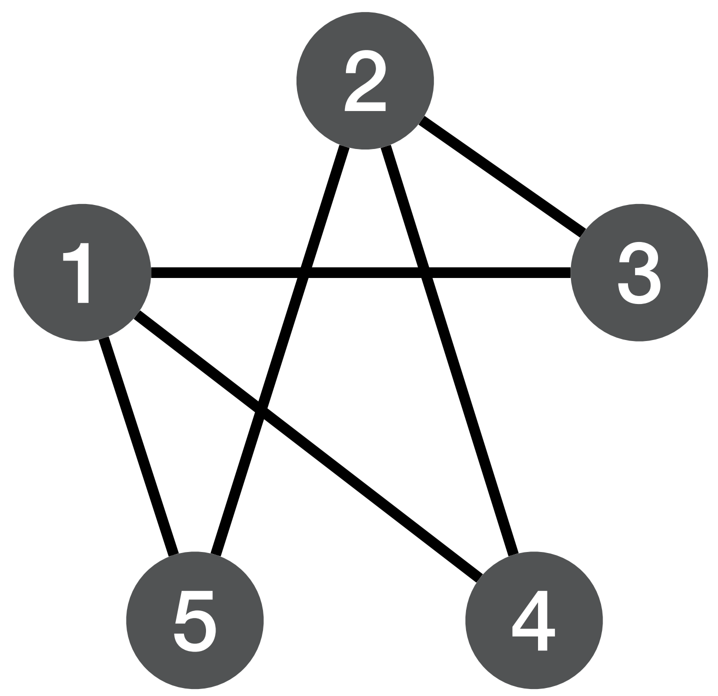
- Many problems in Discrete Mathematics ask for the (non-)existence of combinatorial objects with some property $X$.
- **Combinatorial objects:** graphs, hypergraphs, matroids, etc.
- **Enumeration problems:** Enumerate all objects of size $n$ with property $X$?
- **Extremal problems:** Graphs with smallest/largest number of edges and $n$ vertices with property $X$?
- **Counterexamples to Conjectures:** Show that there is no object with property $X$ of size up to $n$.

# Isomorph-Free Generation

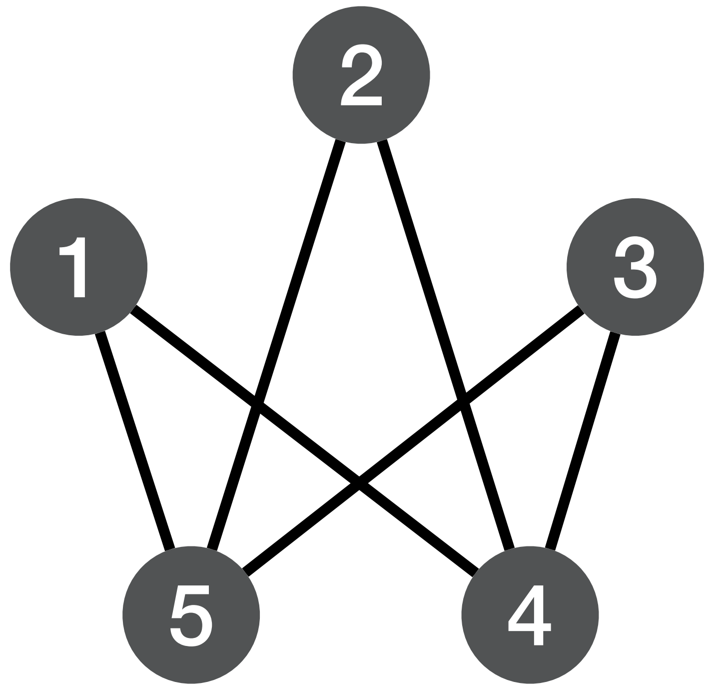
- **Isomorph-free generation:** Number of objects explode quickly, hence we want to avoid generating several isomorphic copies of the same object
- **Canonization:** map each object $G$ to a unique representative $\alpha(G)$ of its isomorphism class
- **Canonical Objects:** Only generate objects $G$ with $\alpha(G) = G$




$1 \mapsto 5$
$2 \mapsto 4$
$3 \mapsto 3$
$4 \mapsto 1$
$5 \mapsto 2$

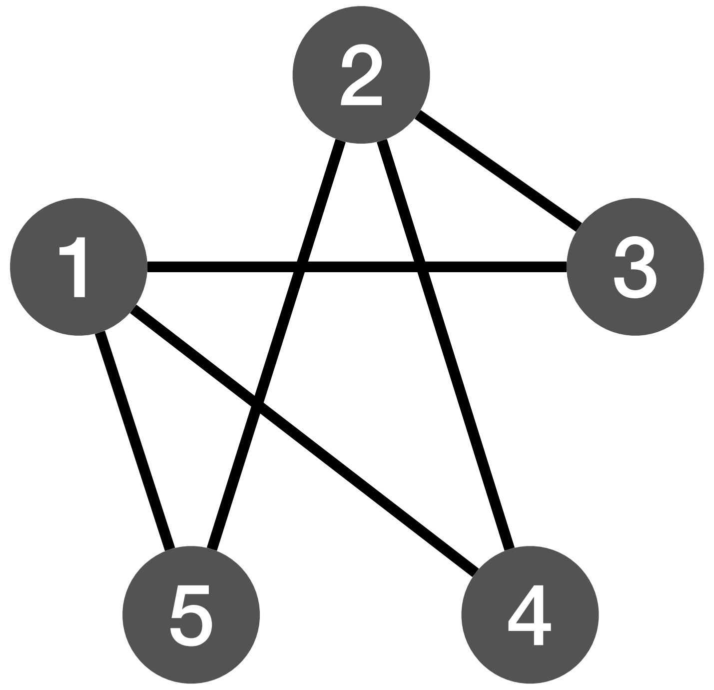


# Canonization by lexicographic ordering




| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 1 |
| 2 | 0 | 0 | 0 | 1 | 1 |
| 3 | 0 | 0 | 0 | 1 | 1 |
| 4 | 1 | 1 | 1 | 0 | 0 |
| 5 | 1 | 1 | 1 | 0 | 0 |

$$\prec$$




| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 | 1 |
| 2 | 0 | 0 | 1 | 1 | 1 |
| 3 | 1 | 1 | 0 | 0 | 0 |
| 4 | 1 | 1 | 0 | 0 | 0 |
| 5 | 1 | 1 | 0 | 0 | 0 |

- consider the adjacency matrix as a long string obtained by concatenating its rows
- order graphs lexicographically by this string
- $\alpha(G) = G$ if $G$ is minimal in its isomorphism class

# Example: connected graphs

| *n* | *connected graphs*<br>A001187 | *canonical connected graphs*<br>A001349 |
|---|---|---|
| 5 | 728 | 21 |
| 6 | 26704 | 112 |
| 7 | 1866256 | 853 |
| 8 | 251548592 | 11117 |
| 9 | 251548592 | 261080 |
| 10 | ≈ 66 billion | 11716571 |
| 11 | ≈ 35 quadrillion ($10^{15}$) | ≈ 1 million |
| 12 | ≈ 73 quintillion ($10^{18}$) | ≈ 16 billion |

OEIS: The On-Line Encyclopedia of Integer Sequences lists > 36000 sequences oeis.org

# Generate & Test

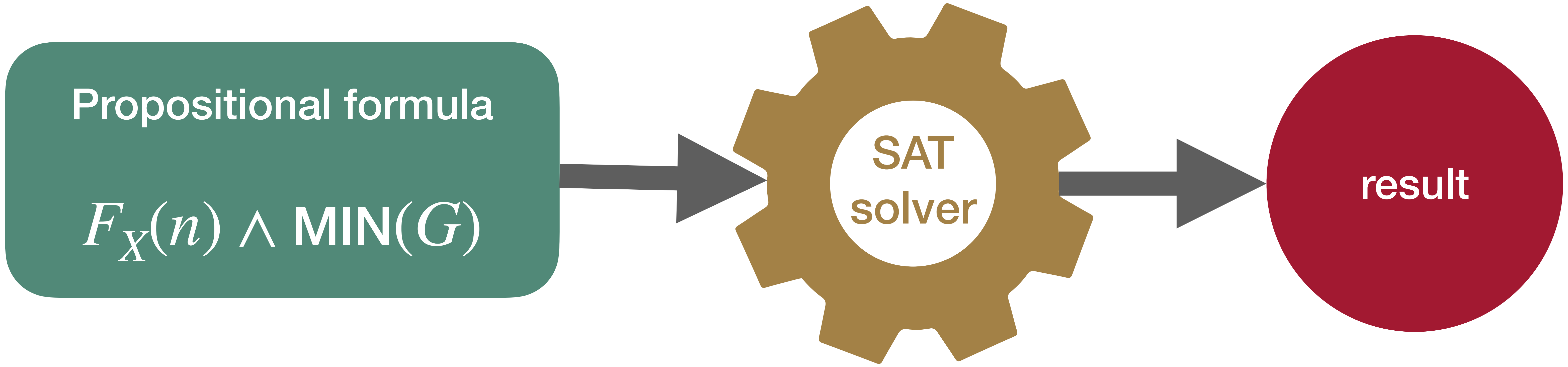
- **Nauty:** popular tool for isomorph-free generation of graphs.
  Based on *canonical construction path method* [McKay 1998]

- **Basic properties**: Good for enumerating graphs with very basic properties like degree restrictions

- **Advanced properties:** handled with generate and test, hence limited to $n \leq 11$ (or slightly larger if degrees are bounded)

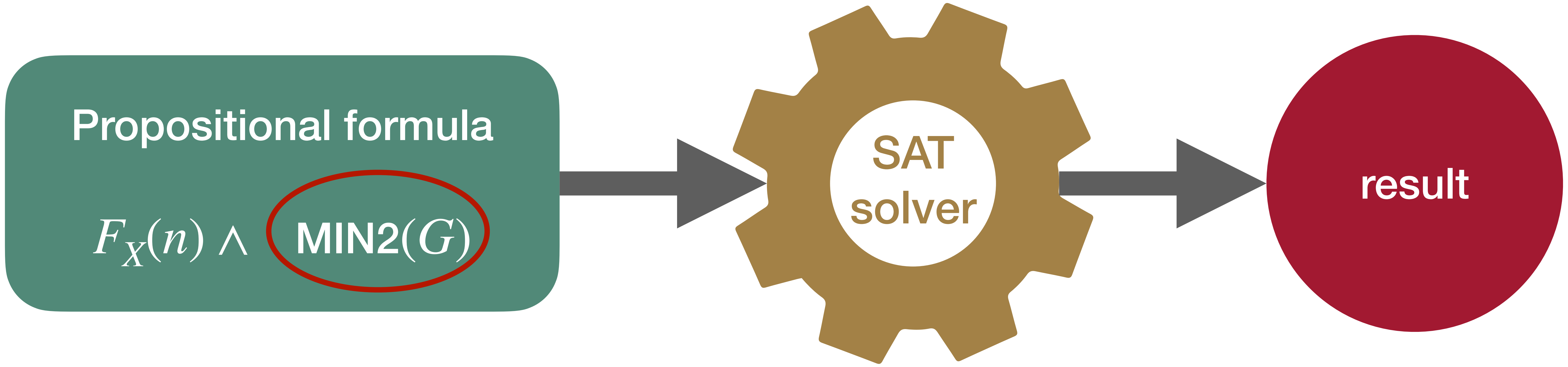
# Static SAT approach

- **Idea:** use SAT to combine generate and test in one process
- **Property:** Express *"G is a graph with $n$ vertices and property $X$"* in a propositional formula $F_X(n)$
- **Object variables:** for each pair $i, j$ of vertices add a variable $e_{i,j}$ which is true iff the edge is present in the graph
- **Auxiliary variables:** used to express the desired property $X$




Problem: MIN$(G)$ no polynomial-size encoding known

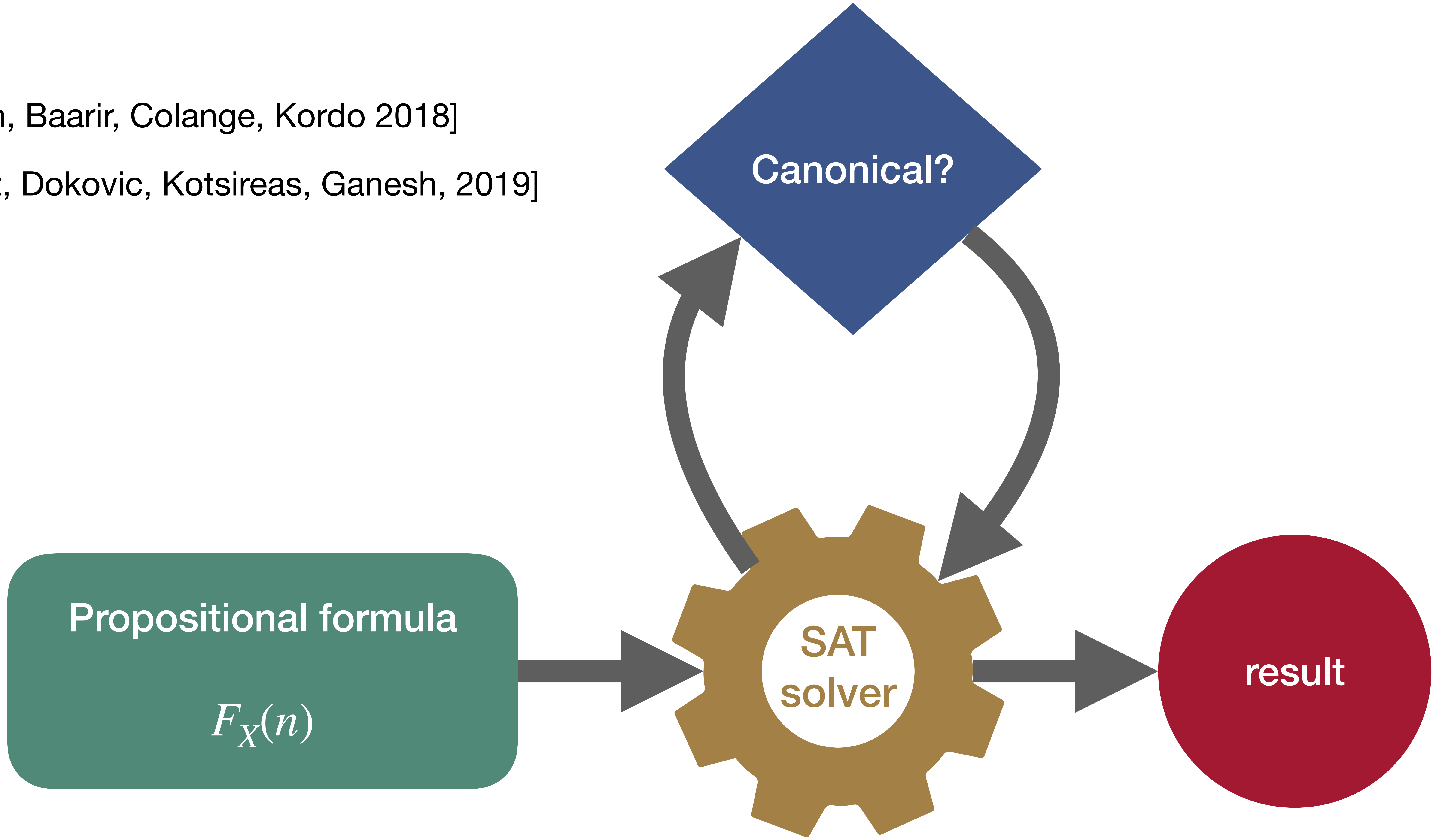
# Static SAT approach




- **Incomplete Static Symmetry Breaking:**
  MIN2$(G)$ = "*if we swap any two vertices the resulting graph isn't lexicographically smaller*"
- [Codish, Miller, Prosser, Stuckey, 2019]
- Good results, although only a small fraction of isomorphic copies is filtered.
- Can we do better?

# Dynamic SAT approach

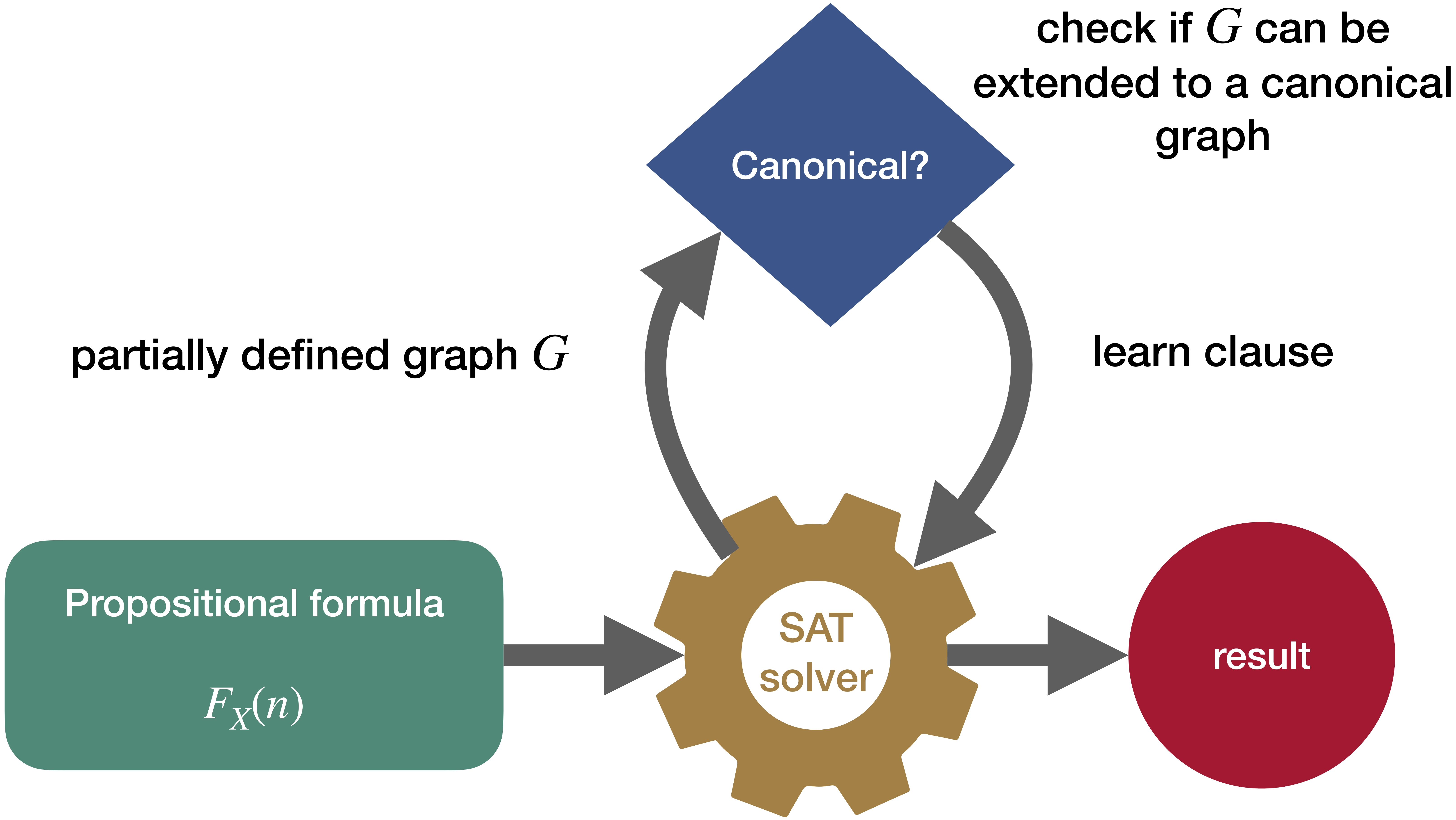
- CDCLSym [Metin, Baarir, Colange, Kordo 2018]
- SAT+CAS [Bright, Dokovic, Kotsireas, Ganesh, 2019]
- and others



# SMS: SAT Modulo Symmetries

Dynamic symmetry breaking by checking the lexicographic minimality of **partially defined graphs** [Kirchweger and Sz. 2021]

# Dynamic Symmetry Breaking with SMS

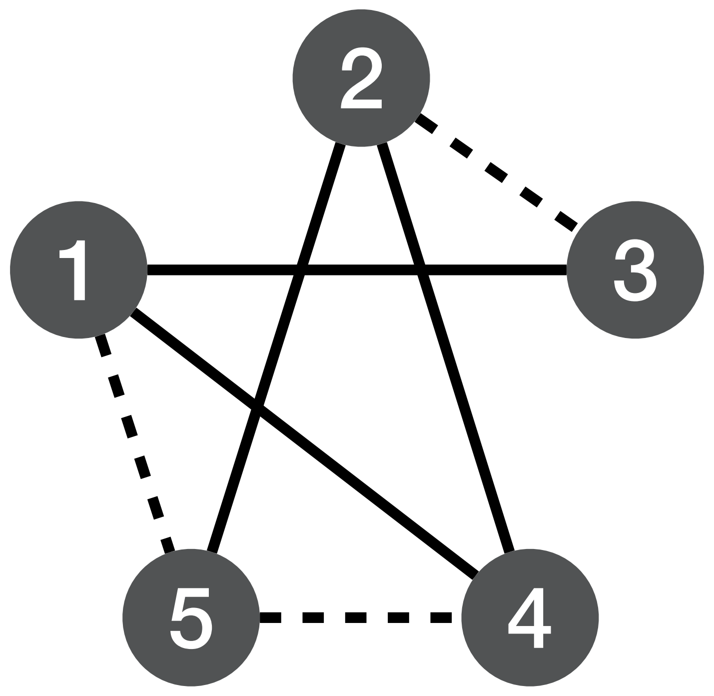


# Partially Defined Graphs

- A partially defined graph is a graph where for some of its edges it is undecided whether they are present or not

defined edge

undefined edge




| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 | * |
| 2 | 0 | 0 | * | 1 | 1 |
| 3 | 1 | * | 0 | 0 | 0 |
| 4 | 1 | 1 | 0 | 0 | * |
| 5 | * | 1 | 0 | * | 0 |

- $G$ is specified by a partition of $E(G)$ into $D(G)$ and $U(G)$.

  (the **d**efined edges and the **u**ndefined edges)

# Natural partial order

- The partially defined graphs over vertex set $\{1,\ldots,n\}$ are partially ordered by

  $G_1 \sqsubseteq G_2$ if $D(G_1) \subseteq D(G_2)$ and $U(G_2) \subseteq U(G_1)$.

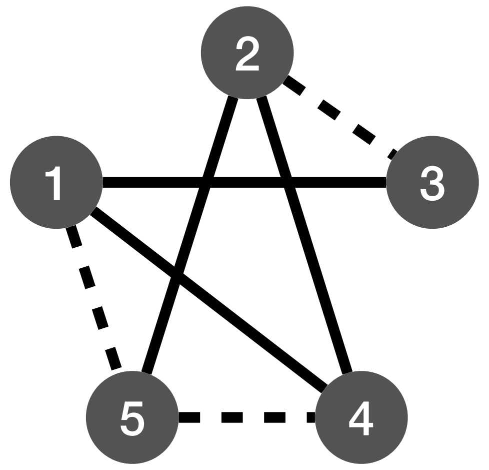
- The minimal element is the graph with all edges undefined.

- The maximal elements are all fully defined graphs over $\{1,\ldots,n\}$.

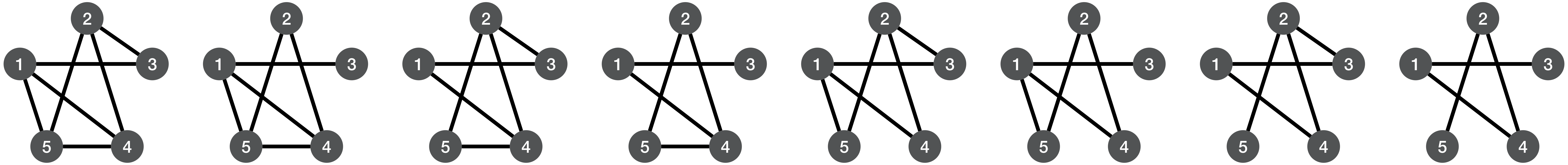
- $\mathscr{X}(G) =$ all fully defined graphs $H$ with $G \sqsubseteq H$

# Extensions to fully defined graphs

partially defined graph $G$




$\mathscr{X}(G)$: set of all fully defined graphs $G$ can be extended to



# Canonicity of partially defined graphs

- **Ideal solution:** reject the current branch if $G$ is **non-canonical** in the sense that none of $H \in \mathscr{X}(G)$ is canonical.
- I.e., if for all $H \in \mathscr{X}(G)$ there is a permutation $\pi$ such that $\pi(H) < H$.
- Extremely difficult to check: need to consider an exponential number of graphs in $\mathscr{X}(G)$, each of them requiring exponential time in the worst case to find the permutation.
- Even if we have determined that $G$ is not canonical, how can we verify this succinctly within a proof.
- Solution: weaker form of canonizity for partially defined graphs

# Certified non-canonicity

- SMS uses the following weaker form of canonizity:
- We reject the current branch if $G$ is **certified non-canonical,**
- i.e., if there is a permutation $\pi$ such that $\pi(H) < H$ for all $H \in X(G)$.
- We can use the permutation $\pi$ as a certificate that can be later verified and checked by an independent method.
- If $G$ is fully defined, it is non-canonical iff it is certified non-canonical.
- Thus we have a **full symmetry breaking** since sooner or later all symmetries will be detected.

# Minimizing learned clauses

- If we have determined that $G$ is certified non-minimal, we can learn a clause

  $$C(G) = \bigvee_{ij \text{ is defined edge}} \neg e_{i,j} \vee \bigvee_{ij \text{ is non-edge}} e_{i,j}$$

  which forbids $G$ and all $G' \sqsupseteq G$

- We can do even better: Compute a $\sqsubseteq$-smallest graph $H \sqsubseteq G$ such that $\pi$ is a certificate for its non-canonizity. Then learn the clause $C(H)$.

- We call $H$ a $(G, \pi)$-**obstruction**.

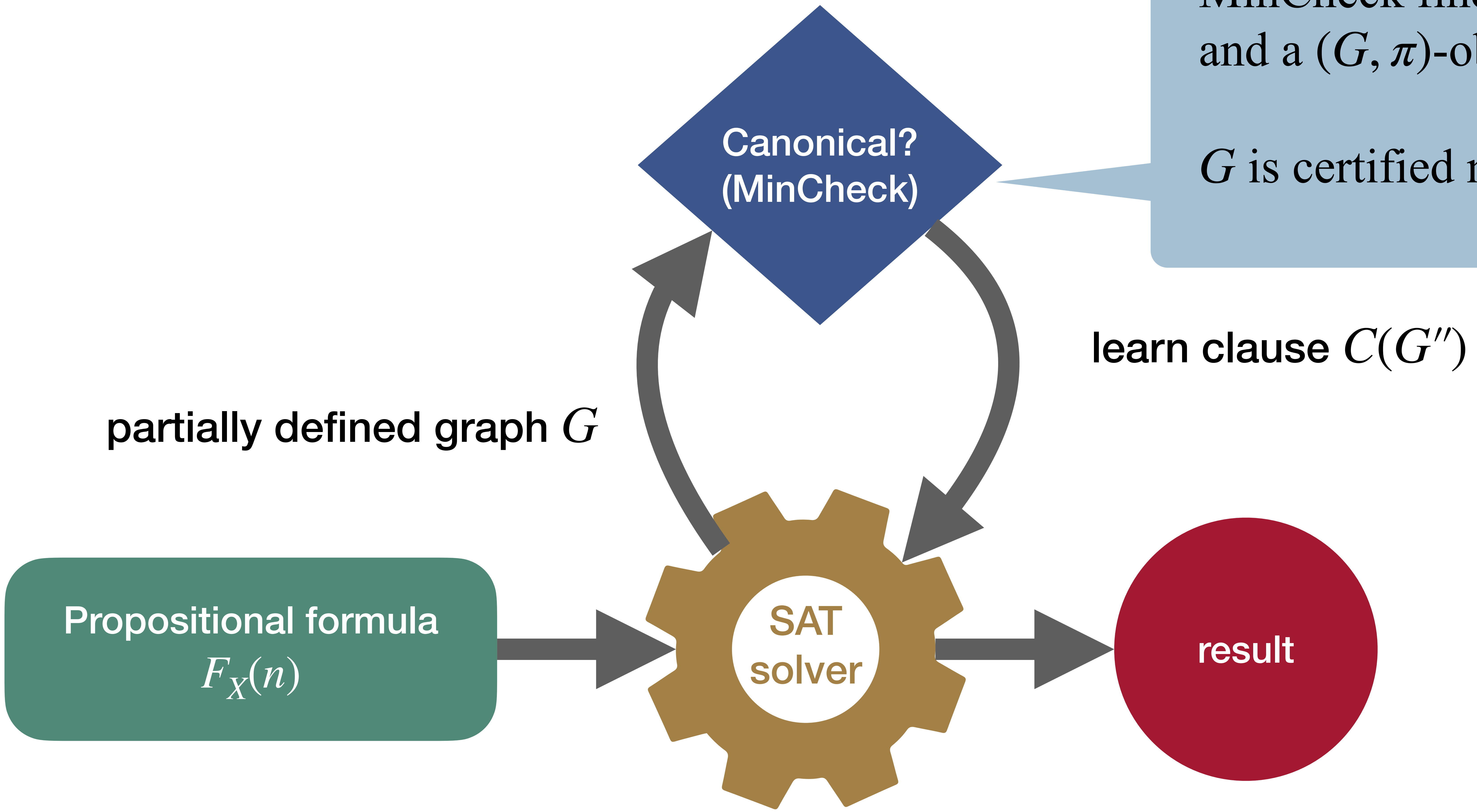
# Completeness

**Theorem:**

MinCheck finds a certificate $\pi$ and a $(G, \pi)$-obstruction $H$

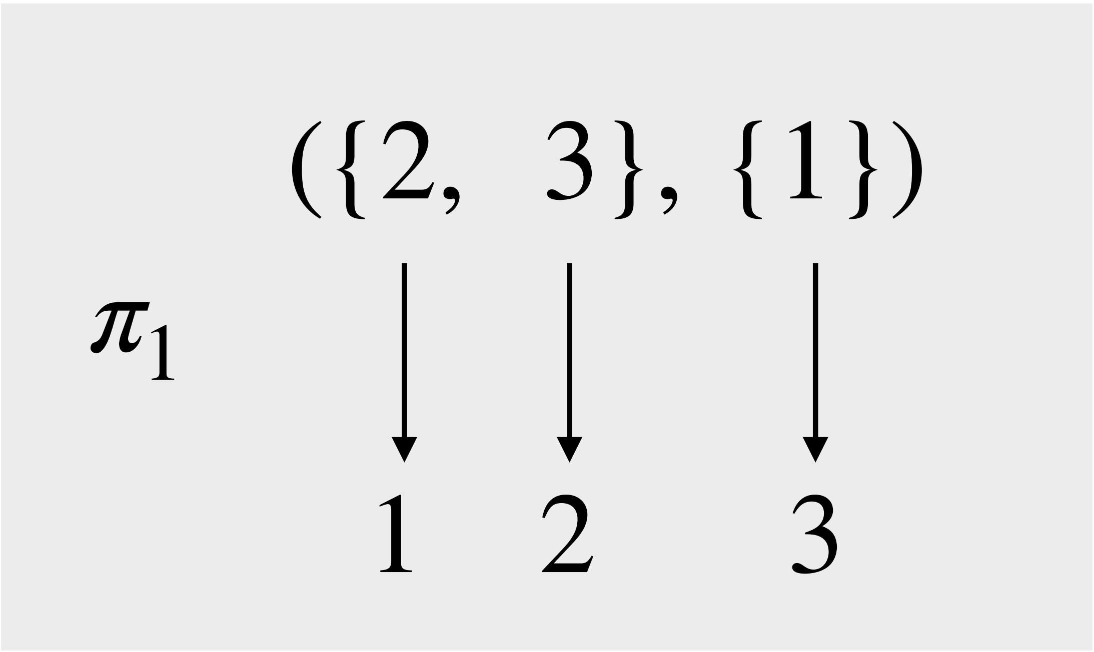
$$\Leftrightarrow$$

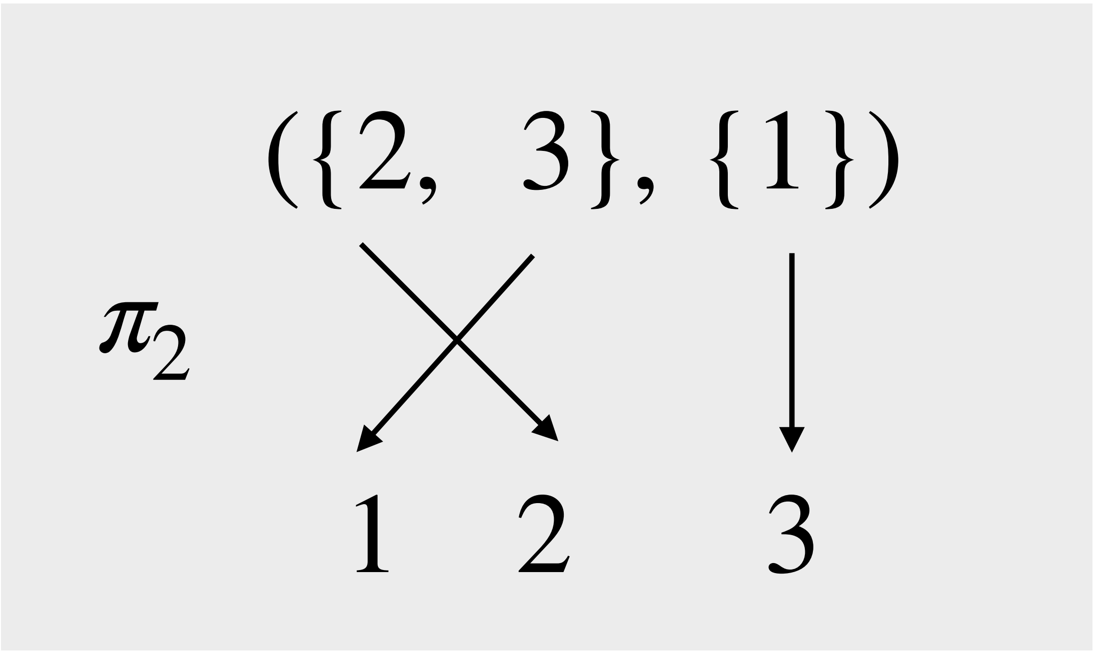
$G$ is certified non-canonical



# Ordered Partions

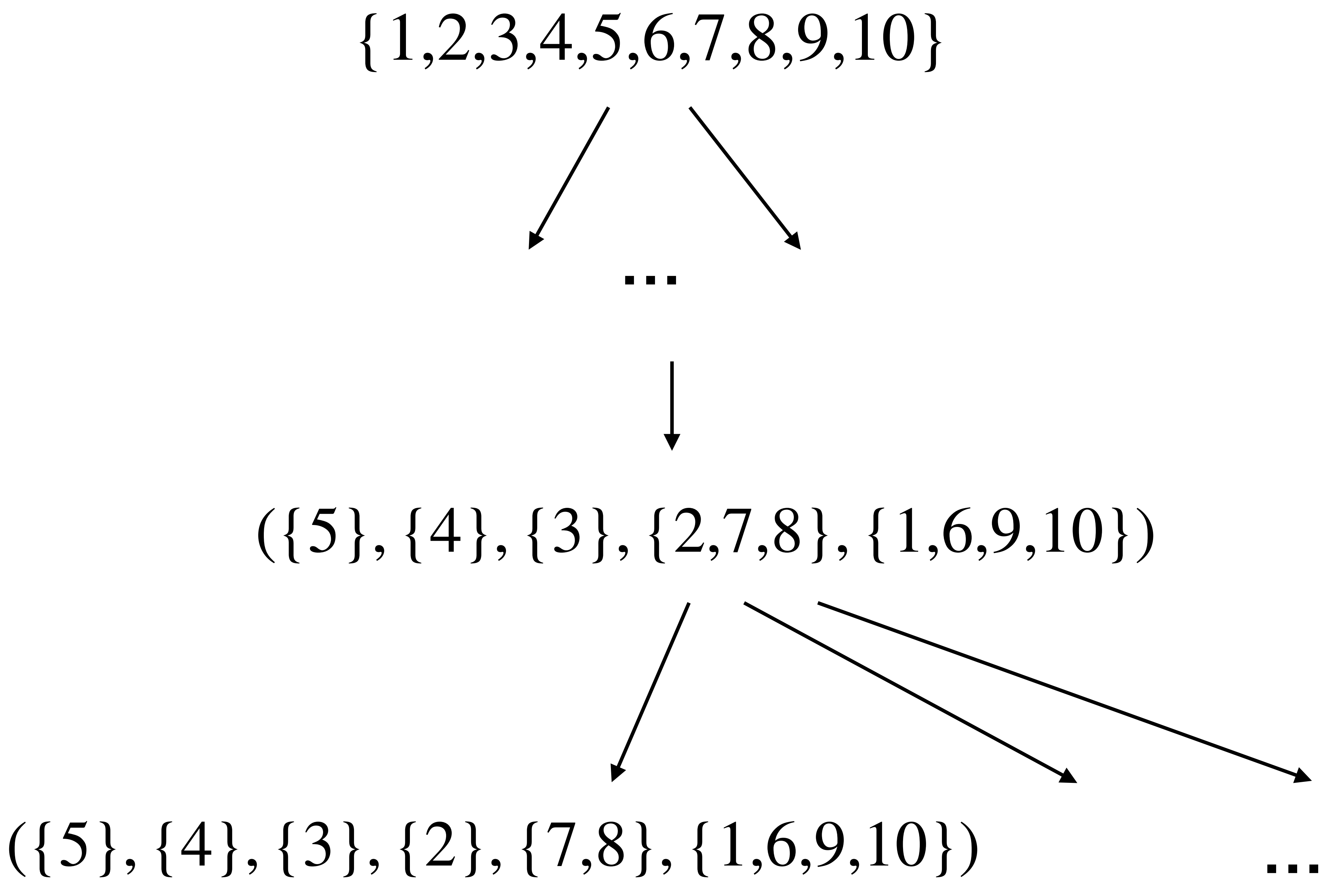
- MinCheck operates on **ordered partitions** of the vertex set, a partition whose equivalence classes are totally ordered
- An ordered partition $(V_1, \ldots, V_r)$ represents all permutations $\pi$ with the property that $u \in V_i, v \in V_j$ for $i < j$ implies $\pi(u) < \pi(v)$.



# Iterative Ordered Partition Refinement

- We start with $P = (V)$, and refine it iteratively from left to right, trying all possibilities of splitting a $V_i$ into a singleton and the rest.
- After each decision, we **propagate:** refine all other equivalence classes without loosing potential certificates



# Propagation

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 0 | 0 | 0 | 0 | 0 | * | * | 1 | 1 | 1 |
| **2** | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| **3** | 0 | 0 | 0 | * | 1 | * | * | * | * | * |
| **4** | 0 | 1 | * | 0 | 0 | * | * | * | * | * |
| **5** | 0 | 1 | 1 | 0 | 0 | 0 | * | * | * | * |
| **6** | * | 1 | * | * | 0 | 0 | * | * | * | * |
| **7** | * | 0 | * | * | * | * | 0 | * | * | * |
| **8** | 1 | 0 | * | * | * | * | * | 0 | * | * |
| **9** | 1 | 1 | * | * | * | * | * | * | 0 | * |
| **10** | 1 | 1 | * | * | * | * | * | * | * | 0 |

$(\{1,2,3,4,5,6,7,8,9,10\})$

$\big\downarrow$ decision $1 \mapsto 1$

$(\{1\}, \{2,3,4,5,6,7,8,9,10\})$

$\big\downarrow$ propagation

$(\{1\}, \{2,3,4,5\}, \{6\}, \{7\}, \{8,9,10\})$

# Propagation

| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 0 | 0 | 0 | 0 | 0 | * | * | 1 | 1 | 1 |
| **2** | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| **3** | 0 | 0 | 0 | * | 1 | * | * | * | * | * |
| **4** | 0 | 1 | * | 0 | 0 | * | * | * | * | * |
| **5** | 0 | 1 | 1 | 0 | 0 | 0 | * | * | * | * |
| **6** | * | 1 | * | * | 0 | 0 | * | * | * | * |
| **7** | * | 0 | * | * | * | * | 0 | * | * | * |
| **8** | 1 | 0 | * | * | * | * | * | 0 | * | * |
| **9** | 1 | 1 | * | * | * | * | * | * | 0 | * |
| **10** | 1 | 1 | * | * | * | * | * | * | * | 0 |

$(\{1,2,3,4,5,6,7,8,9,10\})$

$\downarrow$ decision $1 \mapsto 1$

$(\{1\}, \{2,3,4,5,6,7,8,9,10\})$

$\downarrow$ propagation

$(\{1\}, \{2,3,4,5\}, \{6\}, \{7\}, \{8,9,10\})$

$\downarrow$ decision $2 \mapsto 2$

$(\{1\}, \{2\}, \{3,4,5\}, \{6\}, \{7\}, \{8,9,10\})$

$\downarrow$ propagation

$(\{1\}, \{2\}, \{3\}, \{4,5\}, \{6\}, \{7\}, \{8\}, \{9,10\})$

$\downarrow$

...

# Propagation

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 0 | 0 | 0 | 0 | 0 | * | * | 1 | 1 | 1 |
| **2** | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| **3** | 0 | 0 | 0 | * | 1 | * | * | * | * | * |
| **4** | 0 | 1 | * | 0 | 0 | * | * | * | * | * |
| **5** | 0 | 1 | 1 | 0 | 0 | 0 | * | * | * | * |
| **6** | * | 1 | * | * | 0 | 0 | * | * | * | * |
| **7** | * | 0 | * | * | * | * | 0 | * | * | * |
| **8** | 1 | 0 | * | * | * | * | * | 0 | * | * |
| **9** | 1 | 1 | * | * | * | * | * | * | 0 | * |
| **10** | 1 | 1 | * | * | * | * | * | * | * | 0 |

$(\{1,2,3,4,5,6,7,8,9,10\})$

$\downarrow$ decision $1 \mapsto 1$

$(\{1\}, \{2,3,4,5,6,7,8,9,10\})$

$\downarrow$ propagation

$(\{1\}, \{2,3,4,5\}, \{6\}, \{7\}, \{8,9,10\})$

$\downarrow$ decision $2 \mapsto 5$

$(\{1\}, \{5\}, \{2,3,4\}, \{6\}, \{7\}, \{8,9,10\})$

$\downarrow$ propagation

$(\{1\}, \{5\}, \{4\}, \{2,3\}, \{6\}, \{7\}, \{8,9,10\})$

certifying permutation found!

# Propagation

$G$

| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 0 | 0 | 0 | 0 | 0 | * | * | 1 | 1 | 1 |
| **2** | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| **3** | 0 | 0 | 0 | * | 1 | * | * | * | * | * |
| **4** | 0 | 1 | * | 0 | 0 | * | * | * | * | * |
| **5** | 0 | 1 | 1 | 0 | 0 | 0 | * | * | * | * |

$\pi(G)$

| | **1** | **5** | **4** | **2** | **3** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 0 | 0 | 0 | 0 | 0 | * | * | 1 | 1 | 1 |
| **5** | 0 | 0 | 0 | 1 | 1 | 0 | * | * | * | * |
| **3** | 0 | 1 | * | 0 | 0 | * | * | * | * | * |
| **4** | 0 | 0 | 0 | 1 | * | * | * | * | * | * |
| **2** | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |

$(\{1,2,3,4,5,6,7,8,9,10\})$

decision $1 \mapsto 1$

$(\{1\}, \{2,3,4,5,6,7,8,9,10\})$

propagation

$(\{1\}, \{2,3,4,5\}, \{6\}, \{7\}, \{8,9,10\})$

decision $2 \mapsto 5$

$(\{1\}, \{5\}, \{2,3,4\}, \{6\}, \{7\}, \{8,9,10\})$

propagation

$(\{1\}, \{5\}, \{4\}, \{2,3\}, \{6\}, \{7\}, \{8,9,10\})$

**pick any $\pi$ that is compatible with the current ordered partition**

# Obstruction

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 0 | 0 | 0 | 0 | 0 | * | * | 1 | 1 | 1 |
| **2** | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| **3** | 0 | 0 | 0 | * | 1 | * | * | * | * | * |
| **4** | 0 | 1 | * | 0 | 0 | * | * | * | * | * |
| **5** | 0 | 1 | 1 | 0 | 0 | 0 | * | * | * | * |
| **6** | * | 1 | * | * | 0 | 0 | * | * | * | * |
| **7** | * | 0 | * | * | * | * | 0 | * | * | * |
| **8** | 1 | 0 | * | * | * | * | * | 0 | * | * |
| **9** | 1 | 1 | * | * | * | * | * | * | 0 | * |
| **10** | 1 | 1 | * | * | * | * | * | * | * | 0 |

Edges and non-edges of the obstruction graph:

- Include all edges/non edges that come before the indicator pair that are not stable under $\pi$.
- Include the indicator pair $(2{,}6)$ and its image $(\pi(2), \pi(6)) = (5{,}6)$.

As a clause:

$$e_{1,2} \vee e_{1,3} \vee e_{1,4} \vee e_{1,5} \vee e_{2,3} \vee \neg e_{2,4} \vee \neg e_{2,6} \vee e_{5,6}$$

# Performance of MinCheck

- In the worst case, MinCheck needs to consider all $n!$ permutations.
- In practice, the worst case is rarely attained, propagation excludes many cases.
- If MinCheck uses a significant amount of the solving time:
  - call MinCheck only every $k$'th time a decision on an edge variable has been made
  - call MinCheck only up to $\ell$ recursive calls
- Isomorph-freeness still guaranteed by either running unrestricted MinCheck at the end or check set of solutions for isomorphic copies with separate tool

# Implementation

- MinCheck implemented in C++
- variants for graphs, directed graphs, matroids, hypergraphs
- hosting solver: originally clasp (CDCL ASP solver)
- since recently CaDiCal (modern CDCL SAT solver with inprocessing)
- CaDiCal with IPASIR-UP interface [Fazekas et al. SAT 2023]
- For instances with many clauses, this gives an order-of-magnitude speedup
- Python wrapper for easy use, supports many graph properties from the command line

# Ressources

**Tool** https://github.com/markirch/sat-modulo-symmetries/

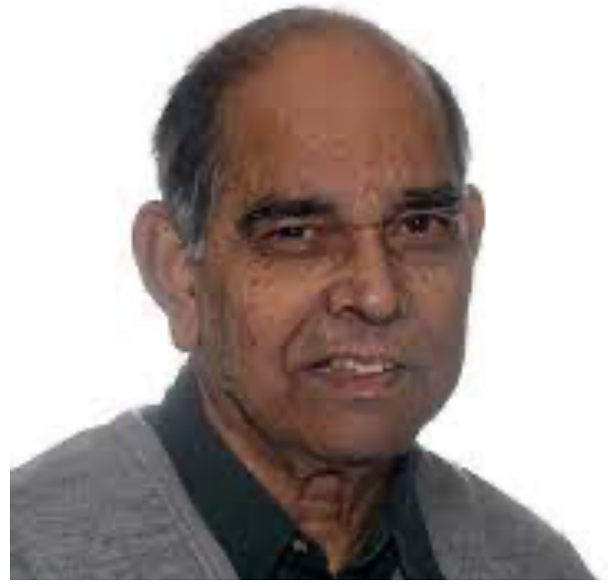
**Documentation** https://sat-modulo-symmetries.readthedocs.io/

# Applications

# Diameter-2-critical graphs: background

- the **diameter** of a graph is the longest distance between any two of its vertices.
- A graph is **diameter-$d$-critical** if its diameter is $d$ but deleting any edge decreases the diameter.
- The study of extremal properties of graphs of given diameter goes back to the 1960s [Erdős and Rényi], much work has been done on diameter-$d$-critical graphs.
- **Simon-Murty Conjecture 1979:** a diameter-2-critical graph with $n$ vertices has at most $\lfloor n^2/4 \rfloor$ edges, with equality holds exactly for the complete bipartite graph $K_{\lfloor n/2 \rfloor, \lceil n/2 \rceil}$.

Imre Simon



U.S.R. Murty

# Diameter-2-critical graphs: encoding

- We use SMS to enumerate diameter-2-critical graphs up to $n = 13$ and verify the Simon-Murty Conjecture up to $n = 19$
- Previous results only up to $n = 10$ with generate-and-test method based on Nauty [Radosavljević and Živković 2020]
- We use auxiliary variables $c_{i,j,k} \leftrightarrow e_{i,k} \wedge e_{j,k}$ to indicate that $i, j$ have a common neighbor $k$
- With these variables it is easy to express that (i) the diameter is 2 and that (ii) deleting any edge gives a diameter $> 2$
- We use a **frequency** parameter, calling MinCheck only every $k$'th time a decision on an object variable has been made ($k = 20$)

# Diameter-2-critical graphs: results

| | | SMS | | Static | |
|---|---|---|---|---|---|
| $n$ | #-graphs | #-sol | time | #-sol | time |
| 3 | 1 | 1 | 0.00(23%) | 1 | 0.00 |
| 4 | 2 | 2 | 0.00(22%) | 2 | 0.00 |
| 5 | 3 | 3 | 0.00(23%) | 4 | 0.00 |
| 6 | 5 | 5 | 0.00(32%) | 11 | 0.00 |
| 7 | 10 | 10 | 0.01(37%) | 32 | 0.01 |
| 8 | 30 | 30 | 0.05(47%) | 163 | 0.04 |
| 9 | 103 | 103 | 0.17(39%) | 1024 | 0.30 |
| 10 | 519 | 519 | 0.73(26%) | 9836 | 3.58 |
| **11** | **3746** | 3748 | 4.48(18%) | 135010 | 77.00 |
| **12** | **40866** | 40876 | 47.71(14%) | t.o. | t.o. |
| **13** | **688120** | 688143 | 1184.47(8%) | t.o. | t.o. |

limit for generate and test
[Radosavljević and Živković 2020]

- Enumeration of diam-2-critical graphs and verifying the Simon-Murty conjecture. (X%) gives time for MinCheck

# Diameter-2-critical graphs: results

| $n$ | total time | max-time | #-comb |
|---|---|---|---|
| 14 | 16 minutes | 9 sec | 406 |
| 15 | 2.2 hours | 20 sec | 1729 |
| 16 | 7.9 hours | 35 sec | 3480 |
| 17 | 19.9 hours | 74 sec | 5620 |
| 18 | 3.4 days | 132 sec | 12974 |
| 19 | 23.7 days | 312 sec | 50054 |

- For verifying the Simon-Murty Conjecture, we can utilize results that restrict the vertex degrees [Fan 1987, Haynes, Henning, Merwe, Yeo 2014], which allows us to scale the search even further.
- If we fix the degree of vertices, we can start MinCheck with an ordered partition where vertices of the same degree form equivalence classes.
- max-time: time for a single combination. #-comb: number of combinations

# Proofs with SMS

- With SMS we can certify the correctness of results by means of DRAT proofs.
- Adding the clauses generated by MinCheck to the original encoding, we can produce a DRAT proof by any CDCL solver that supports DRAT.
- The added clauses can also be verified independently when certificate permutation is logged.
- This gives an additional application domain for SMS: providing further confidence in known results obtained by other methods.
- For enumeration tasks, we add clauses excluding all the previously found solutions. This way, we can certify the completeness of the enumeration.

# Ramsey Sets: background

- For positive integers $x$ and $y$, the **Ramsey set** $\mathscr{R}(x, y, n)$ is the set of all $n$-vertex graphs up to isomorphism not containing an independent set of size $x$ nor a clique of size $y$.

- The **Ramsey number** $R(x, y)$ is the smallest integer such that $\mathscr{R}(x, y, R(x, y)) = \emptyset$ .

- Encoding the Ramsey property is straightforward (go over all subsets of vertices of size $x$ and require that the set induces an edge; go over all subsets of vertices of size $y$ and require that the set induces a non-edge)

Frank P. Ramsey
(1903-1930)

# Ramsey Sets: results

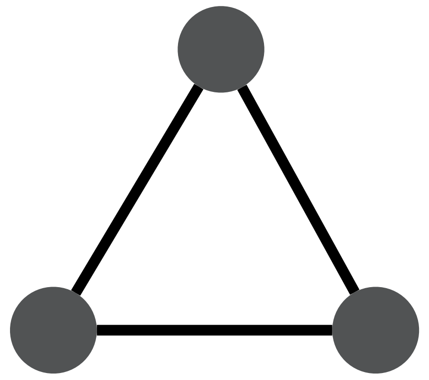
- $R(3,5) = 14$, $R(4,4) = 18$
- First proofs for the numbers

| | $\mathcal{R}(3,5,n)$ | | $\mathcal{R}(4,4,n)$ | |
|---|---|---|---|---|
| $n$ | #-sol | time | #-sol | time |
| 1 | 1 | 0.00 | 1 | 0.00 |
| 2 | 2 | 0.00 | 2 | 0.00 |
| 3 | 3 | 0.00 | 4 | 0.00 |
| 4 | 7 | 0.00 | 9 | 0.00 |
| 5 | 13 | 0.00 | 24 | 0.00 |
| 6 | 32 | 0.00 | 84 | 0.01 |
| 7 | 71 | 0.01 | 362 | 0.03 |
| 8 | 179 | 0.03 | 2079 | 0.16 |
| 9 | 290 | 0.05 | 14701 | 1.25 |
| 10 | 313 | 0.05 | 103706 | 11.99 |
| 11 | 105 | 0.05 | 546356 | 80.92 |
| 12 | 12 | 0.03 | 1449166 | 531.44 |
| 13 | 1 | 0.02 | 1184231 | 227.95 |
| 14 | | | 130816 | 28.70 |
| 15 | | | 640 | 0.66 |
| 16 | | | 2 | 0.15 |
| 17 | | | 1 | 0.14 |
| total | 1029 | 0.24 | 3432184 | 883.41 |

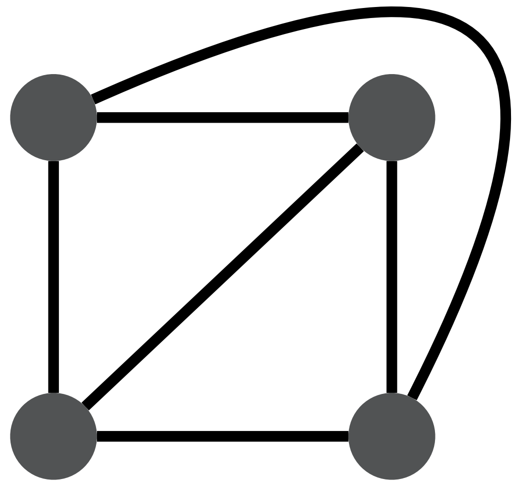
# Generating Planar Graphs with SMS

- A graph is **planar** if it can be drawn on the plane such that edges are represented by continuous curves that do intersect in the interior.

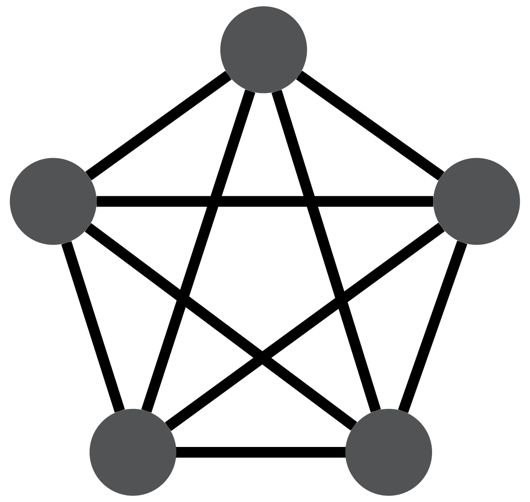
$K_3$



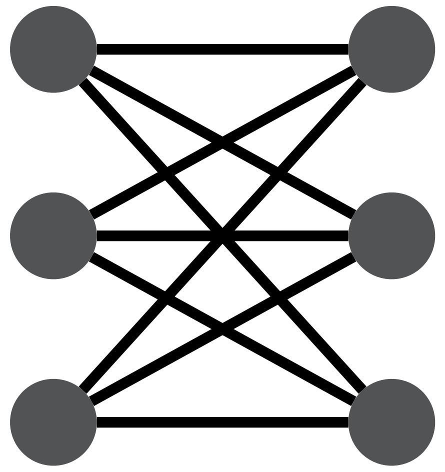
$K_4$



$K_5$



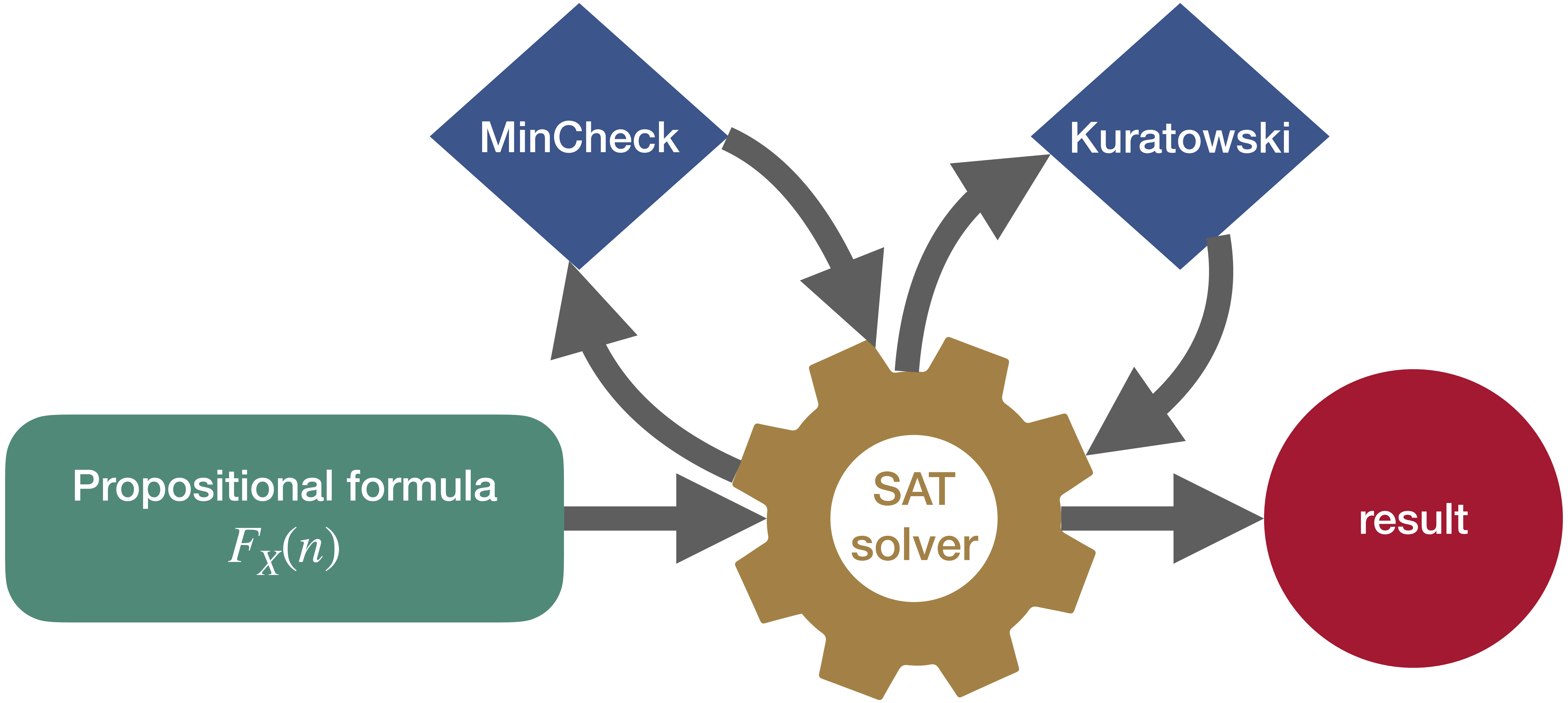
$K_{3,3}$



- The best known planarity criterium is **Kuratowski's Theorem.** It is a **negative criterium:** a graph is planar iff it does not contain a subdivision of $K_{3,3}$ or $K_5$ as subgraph. We implemented this with a **propagator**.
- Similar to they **co-certificate learning (CCL)**, used for finding lower bounds for the size of Kochen-Specker Systems [Kirchweger, Peitl, S. IJCAI 2023]
- One can encode planarity in CNF with a polynomial number of clauses. We tried two positive planarity criteria for that **Schnyder Orders** and **Universal Sets.**

Kazimierz Kuratowski
(1896-1980)

# Lazy planarity encoding based on Kuratowski




- Planarity testing algorithm that produces $K_{3,3}$, $K_5$ subdivisions [Boyer and Myrvold 2004]
- Run it on the fully defined graph obtained from the partially defined graph by considering all undefined edges as non-edges
- Clearly outperforms the eager encodings.
- Provides a competitive alternative to **plantri** [Brinkmann and McKay 2007]

# Planar Turán Numbers: background

- **Turán numbers:** $\mathrm{ex}(n, H)$ maximum number of edges in an $n$-vertex graph that excludes a subgraph $H$ . Turán's Theorem [1941] covers the case where $H$ is a clique.

- **Planar Turán numbers:** $\mathrm{ex}_P(n, H)$ = maximal number of edges in an $n$-vertex **planar** graph that excludes $H$.

- Has been studied for $H = C_4,\ H = C_5$ [Dowden 2016]

- Encoding: simply go over all sets of $k = 4{,}5$ vertices and add a clause that requires the vertices do not generate a cycle

Pál Turán
(1910-1976)

# Planar Turán Numbers: results $k = 4$

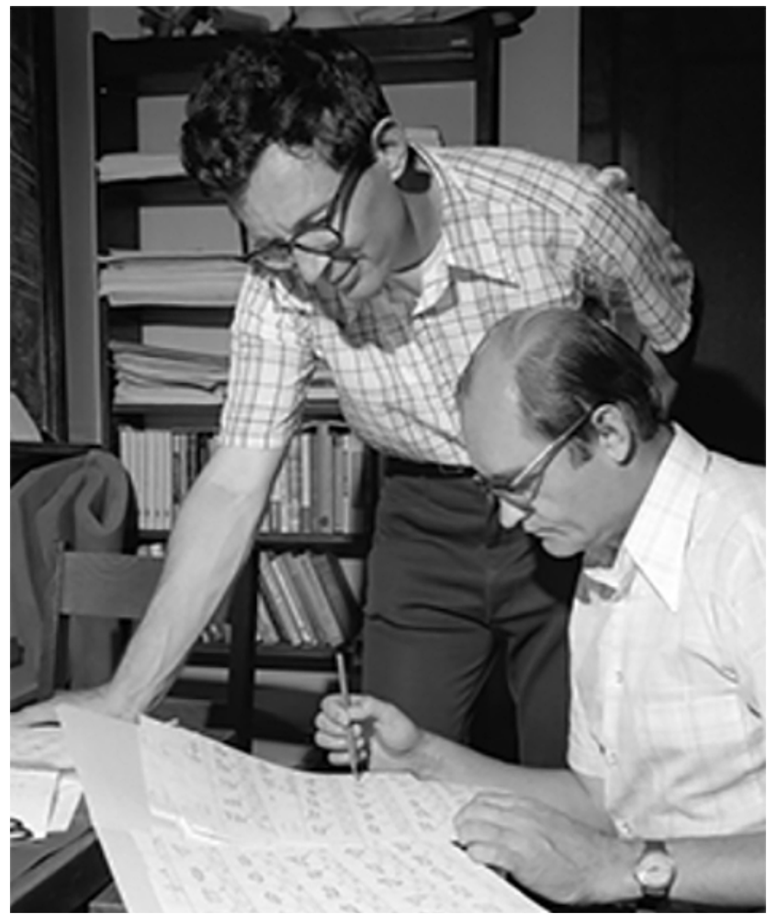
| | | SAT | | UNSAT | |
|---|---|---|---|---|---|
| $n$ | $\mathrm{ex}_P(n, C_4)$ | Kura | Ord | Kura | Ord |
| 4 | 4 | 0.00 | 0.00 | 0.00 | 0.00 |
| 5 | 6 | 0.00 | 0.00 | 0.00 | 0.00 |
| 6 | 7 | 0.00 | 0.00 | 0.00 | 0.01 |
| 7 | 9 | 0.01 | 0.01 | 0.01 | 0.02 |
| 8 | 11 | 0.01 | 0.02 | 0.02 | 0.03 |
| 9 | 13 | 0.03 | 0.04 | 0.05 | 0.05 |
| 10 | 16 | 0.04 | 0.07 | 0.07 | 0.06 |
| 11 | 18 | 0.16 | 0.44 | 0.16 | 0.23 |
| 12 | 20 | 0.27 | 0.98 | 0.56 | 2.29 |
| 13 | 22 | 0.23 | 0.14 | 1.96 | 15.27 |
| 14 | 24 | 0.20 | 0.44 | 6.46 | 340.11 |
| 15 | 27 | 1.00 | 0.85 | 21.39 | 294.07 |
| 16 | 29 | 5.87 | 24.90 | 172.90 | 31142.08 |
| 17 | 31 | 5.19 | 83.59 | 3479.65 | t.o. |
| 18 | 33 | 14.69 | 14.85 | 59862.72 | t.o. |

# Planar Turán Numbers: results $k = 5$

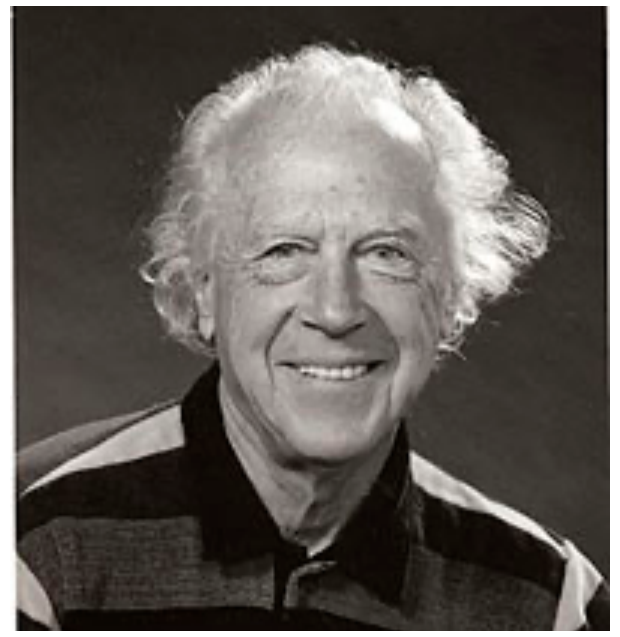
| | | SAT | | UNSAT | |
|---|---|---|---|---|---|
| $n$ | $\mathrm{ex}_P(n, C_5)$ | Kura | Ord | Kura | Ord |
| 5 | 7 | 0.00 | 0.01 | 0.00 | 0.00 |
| 6 | 9 | 0.00 | 0.01 | 0.01 | 0.01 |
| 7 | 12 | 0.01 | 0.02 | 0.02 | 0.01 |
| 8 | 13 | 0.02 | 0.07 | 0.05 | 0.11 |
| 9 | 15 | 0.03 | 0.06 | 0.07 | 0.38 |
| 10 | 18 | 0.10 | 0.29 | 0.23 | 1.67 |
| 11 | 19 | 0.12 | 0.30 | 0.57 | 4.89 |
| 12 | 22 | 1.83 | 1.72 | 1.99 | 33.08 |
| 13 | 24 | 0.48 | 1.61 | 11.45 | 271.18 |
| 14 | 27 | 3.18 | 7.63 | 35.24 | 1174.85 |
| 15 | 29 | 2.24 | 10.82 | 277.78 | 15459.24 |
| 16 | 31 | 4.71 | 59.09 | 3172.27 | 235353.58 |
| 17 | 34 | 207.49 | 890.98 | 29023.55 | t.o. |
| 18 | 36 | 1851.84 | 1249.38 | t.o. | t.o. |

# Earth-Moon problem: background

- **Four-Color Theorem:** The most famous computer assisted mathematical proof [Appel, Haken 1977]. Every planar graph is 4-colorable.
- **Ringel's Earth-Moon problem 1959:** How many colors are sufficient for a **biplanar** graph (edges can be partitioned into two planar graphs): every country has a colony on the moon. Each country gets the same color as its colony.
- The Earth-Moon problem is "hard as two or three four-color theorems" [Hutchinson 2016]

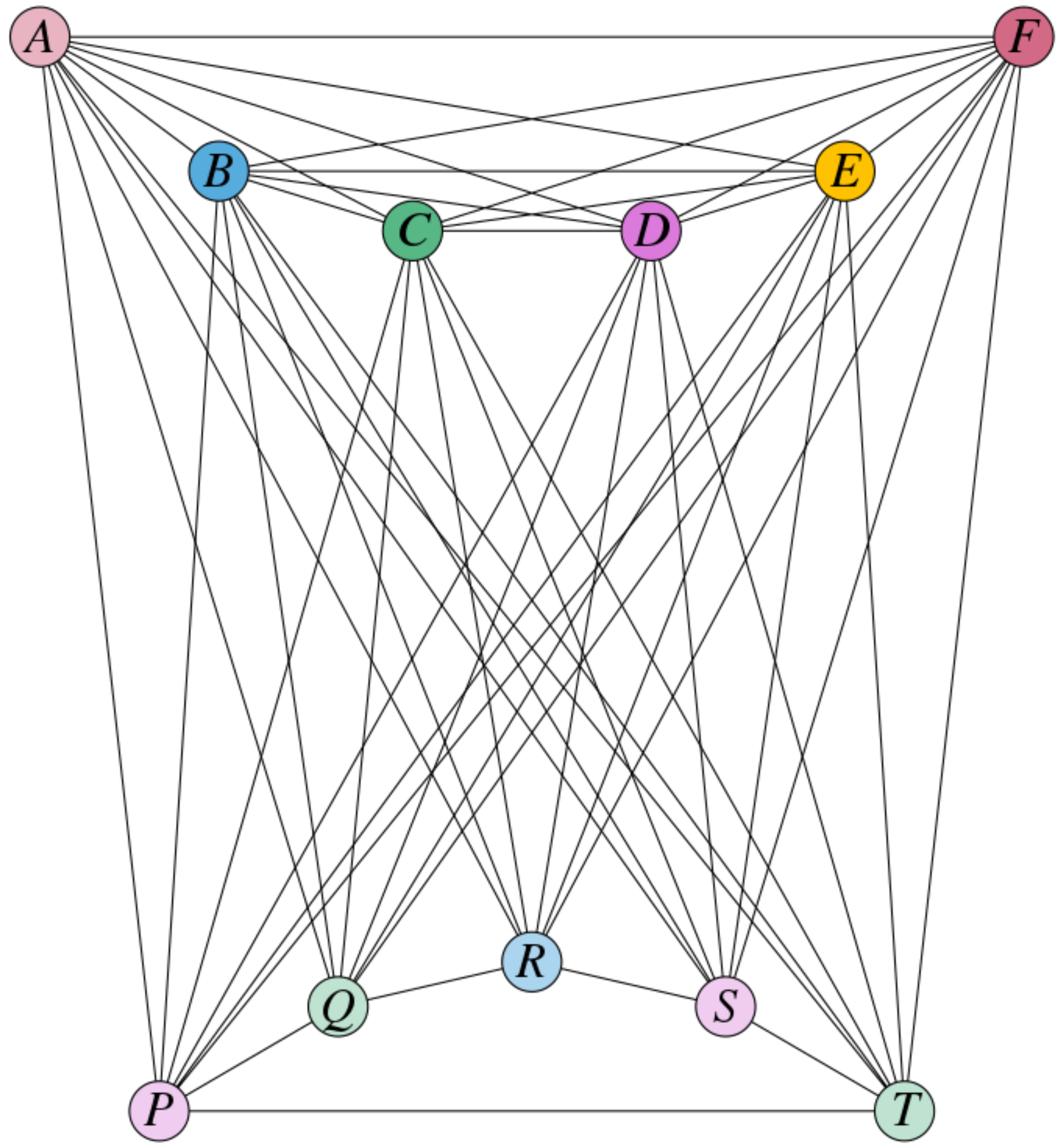


Kenneth Appel
Wolfgang Haken

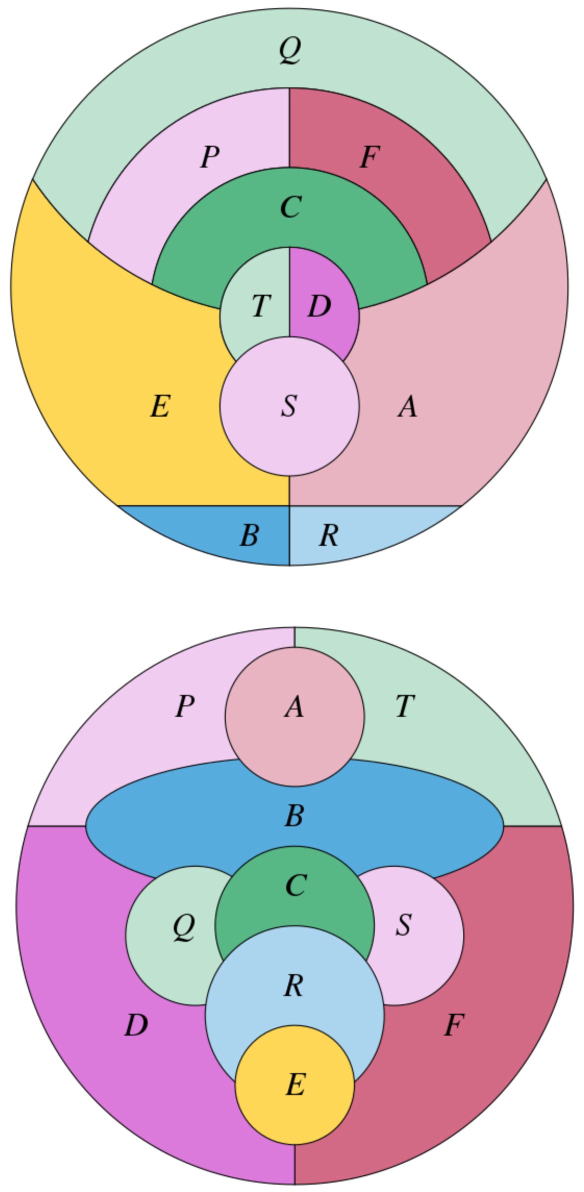


Gerhard Ringel

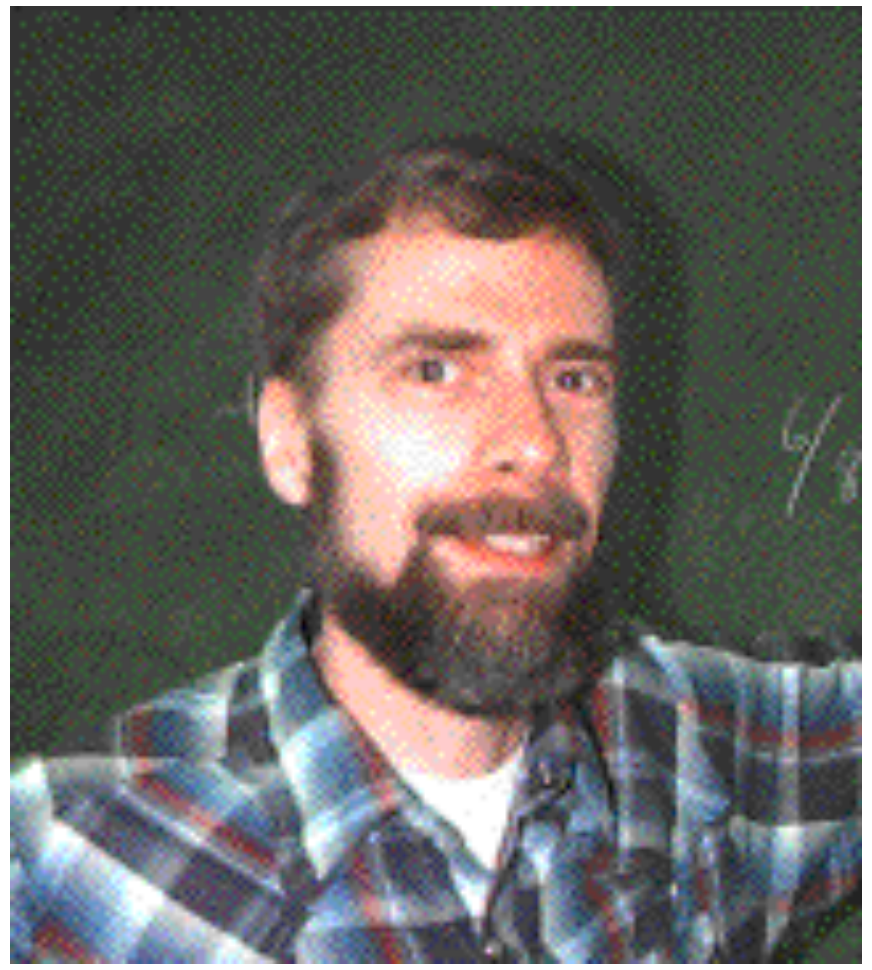
# Sulanke's Graph




Moon




Earth

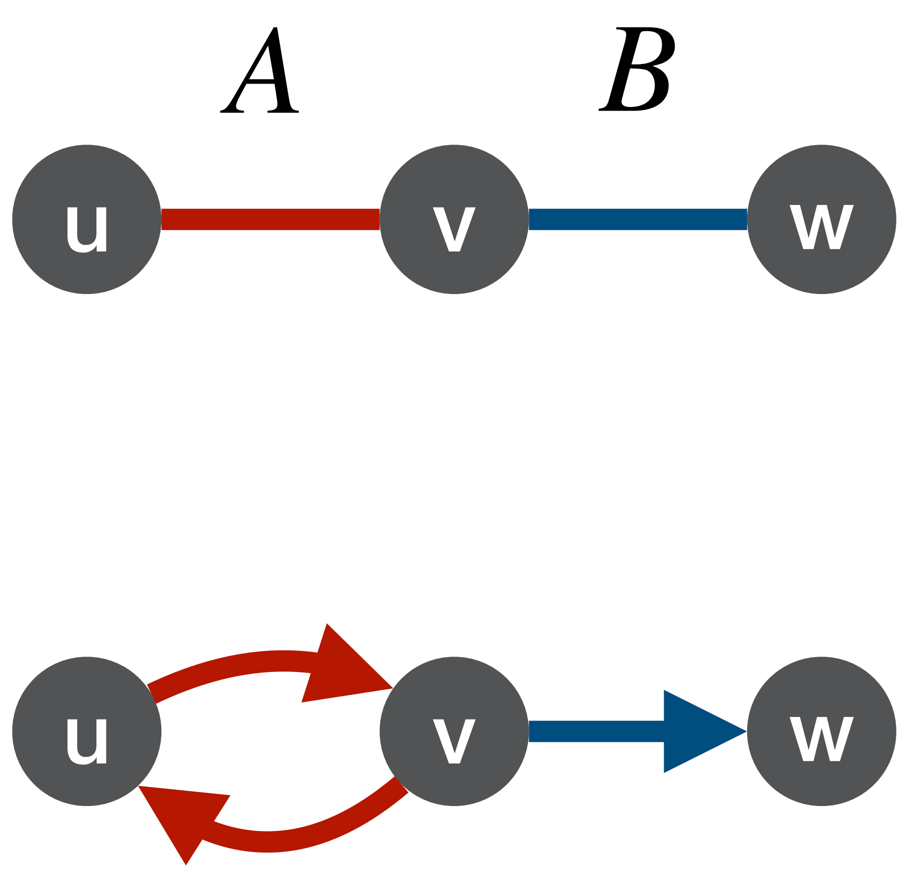


Thom Sulanke

- **Known:** answer lies between 9 and 12
  9 is due to Sulanke [1973], 12 follows by Euler's formula.
  Biplanar graph with 11 vertices that needs 9 colors, found 1973 by Sulanke.

# Earth-Moon problem: encoding

- We encode the bipartition of the edge set $E = A \uplus B$ as a directed graph
- Developed a MinCheck variant for directed graphs
- Planarity of graphs $A$ and $B$ checked with Kuratowski's criterium
- non $k$-colorability checked by coloring clauses (consider all possible partitions of $V$ into $k$ color classes)
- feasible since $n$ is small



# Earth-Moon problem: results

| $n$ | # colors: 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|
| 8 | $K_8$ | | | | | |
| 9 | | $K_9$ | | | | |
| 10 | | new | $K_{10}$ | | | |
| 11 | | Sulanke | new | $K_{11}$ | | |
| 12 | | | new | new | $K_{12}$ | |
| 13 | | | new | new | new | $K_{13}$ |
| 14 | | | open | open | open | |

- Theorem: All biplanar graphs on $n \leq 13$ vertices are 9-colorable.

# SMS for Hypergraphs

- **Hypergraphs**: reuse graph MinCheck by starting with a special ordered partition.

- Verified the *Erdős-Faber-Lovász Conjecture* [1072] for $n \leq 12$ and several cases of $13 \leq n \leq 18$.

Erdős

Faber

Lovász

| | $e_1$ | $e_2$ | $e_3$ | $e_4$ |
|---|---|---|---|---|
| $v_0$ | 1 | 1 | 0 | 0 |
| $v_1$ | 1 | 0 | 1 | 0 |
| $v_2$ | 1 | 0 | 0 | 1 |
| $v_3$ | 0 | 1 | 1 | 1 |

incidence matrix

| | $v_0$ | $v_1$ | $v_2$ | $v_3$ | $e_1$ | $e_2$ | $e_3$ | $e_4$ |
|---|---|---|---|---|---|---|---|---|
| $v_0$ | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| $v_1$ | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| $v_2$ | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| $v_3$ | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| $e_1$ | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| $e_2$ | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| $e_3$ | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| $e_4$ | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |

adjacency matrix of incidence graph

**start with ordered partition**

$$(\{v_0, v_1, v_2, v_3\}, \{e_1, e_2, e_3, e_4\})$$

# SMS for Matroids

- **Matroids**: we have developed a new variant of MinCheck for matroids of bounded rank.

- With SMS we could verify *Rota's Basis Conjecture* [1994] for matroids of rank 4 and matroids of rank 5 and girth 4.

  [Kirchweger, Scheucher, Sz. 2022]

Gian Carlo Rota
(1932-1999)

# Summary

- SMS provides a powerful framework for isomorph-free generation of combinatorial objects
- At its heart is an efficient algorithm that checks for certified non-canonicity of partially defined objects through an iterative refinement of ordered partitions
- Utilizes the power of modern CDCL SAT solver through the
IPASIR-UP interface
- Generates DRAT proofs for independent verification
- Several applications: graphs (directed, undirected, planar), hypergraphs, and matroids of bounded rank.

- Better Minimality Check? CSP approach?
- Combine SMS with static symmetry breaking?
- SMS for QBF?

**Thanks!**

**Tool** https://github.com/markirch/sat-modulo-symmetries/

**Documentation** https://sat-modulo-symmetries.readthedocs.io/