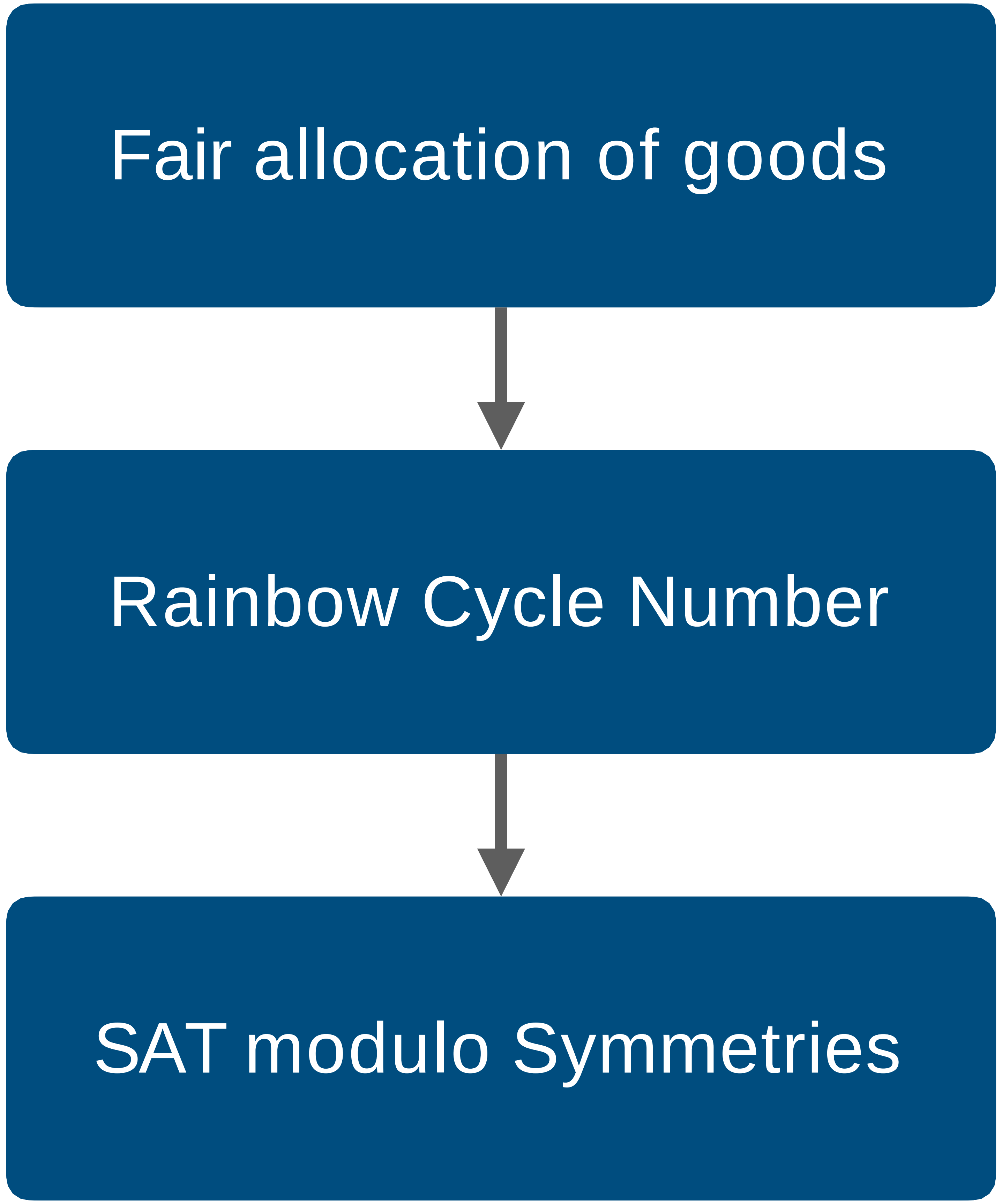
Computing small Rainbow Cycle Numbers with SAT modulo Symmetries

Markus Kirchweger
TU Wien, Vienna, Austria

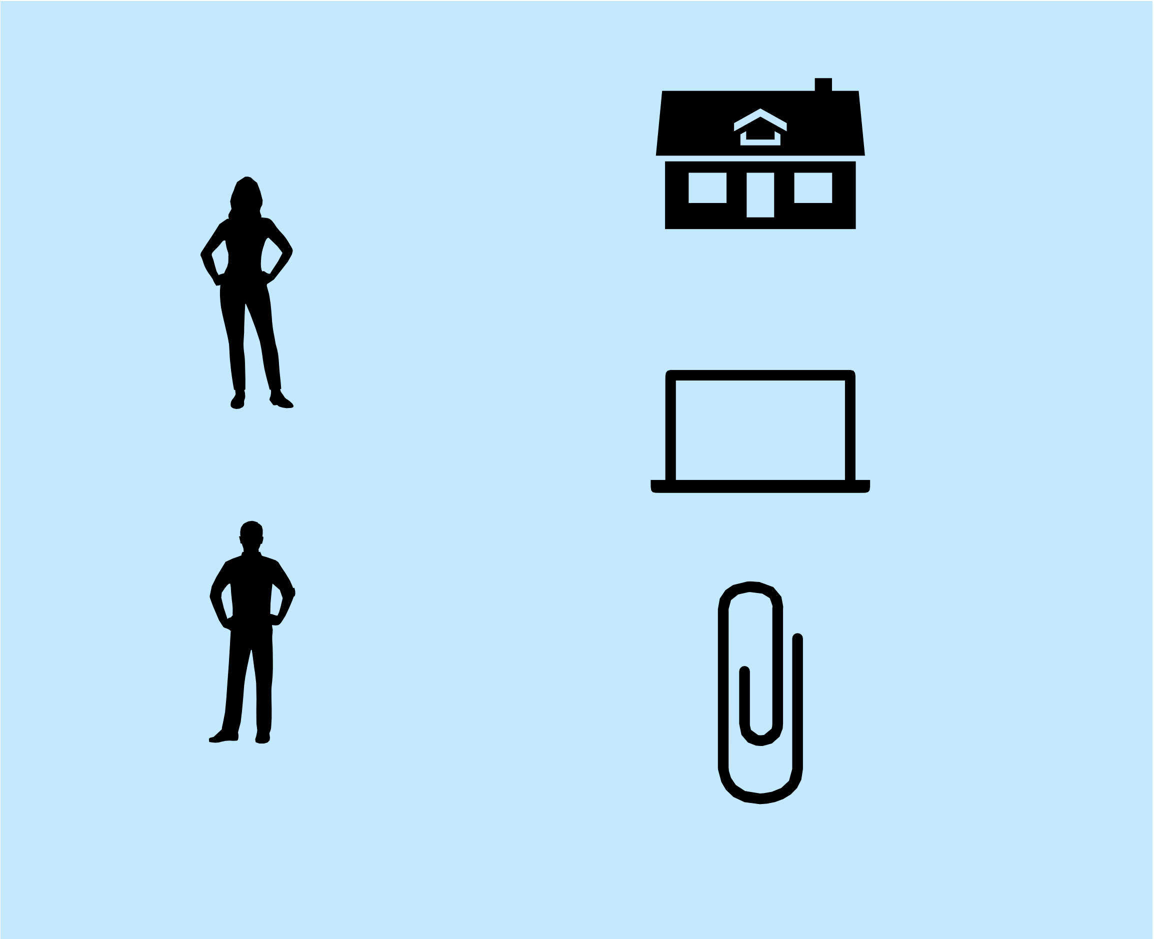
Stefan Szeider
TU Wien, Vienna, Austria

FWF Der Wissenschaftsfonds.
WWTF VIENNA SCIENCE AND TECHNOLOGY FUND
ac ALGORITHMS AND COMPLEXITY GROUP
CP 2024 GIRONA

Outline


Fair allocation of goods
Rainbow Cycle Number
SAT modulo Symmetries

Fair allocation of goods



agents $\left\{ 1, \vdots, n \right.$ **allocation** $\vdots$ set of goods

- allocate to each agent a set of goods $A_i \subseteq M$ such that

$$A_i \cap A_j = \emptyset \quad and \quad \cup_i A_i = M$$

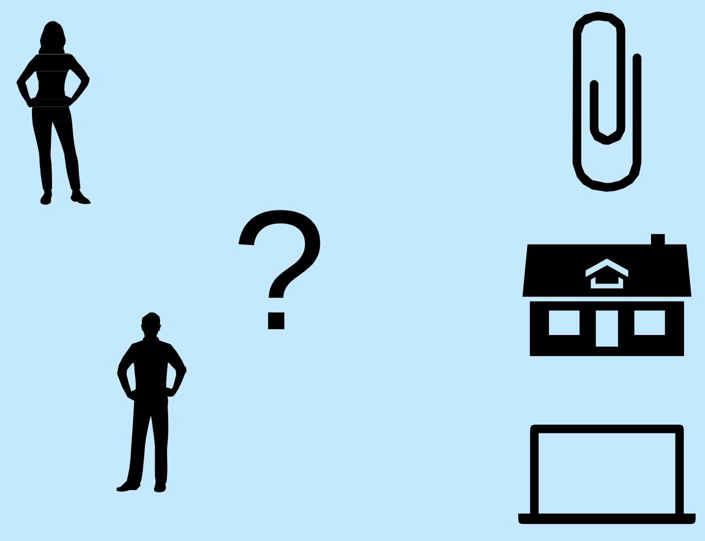
- a **partial** allocation allows some goods unassigned (or donated to *charity*)

- **valuation**: $v_i : 2^M \rightarrow \mathbb{R}_{\geq 0}$

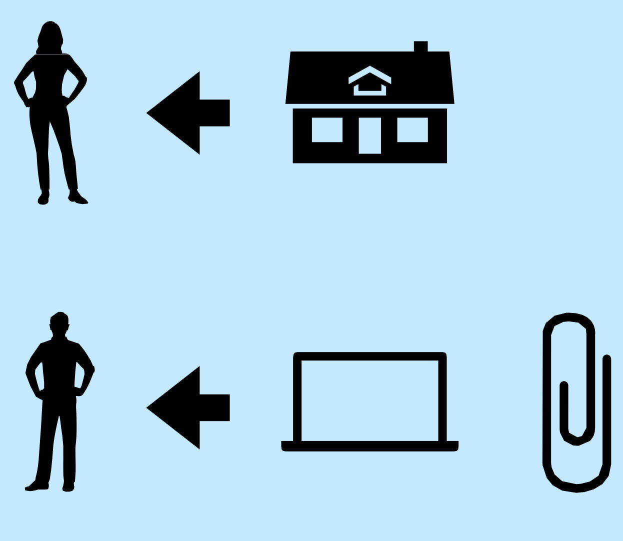
			0	0
		✓	1	1
	✓		1000	900
	✓	✓	1001	901
✓			1000000	1100000
✓		✓	1000001	1100001
✓	✓		1001000	1100900
✓	✓	✓	1001001	1100901

Envy-freeness

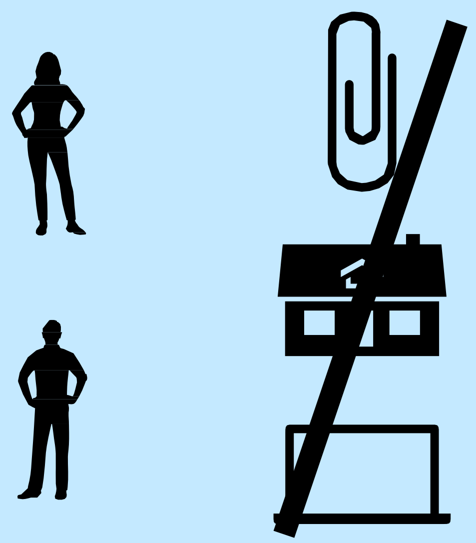
- EF … envy-free if $\forall i,j : v_i(A_i) \geq v_i(A_j)$
- EFX … envy-free up to **any** good: $\forall i,j,\ \forall g \in A_j : v_i(A_i) \geq v_i(A_j \setminus \{g\})$



EF ∄



EFX



EF with charity

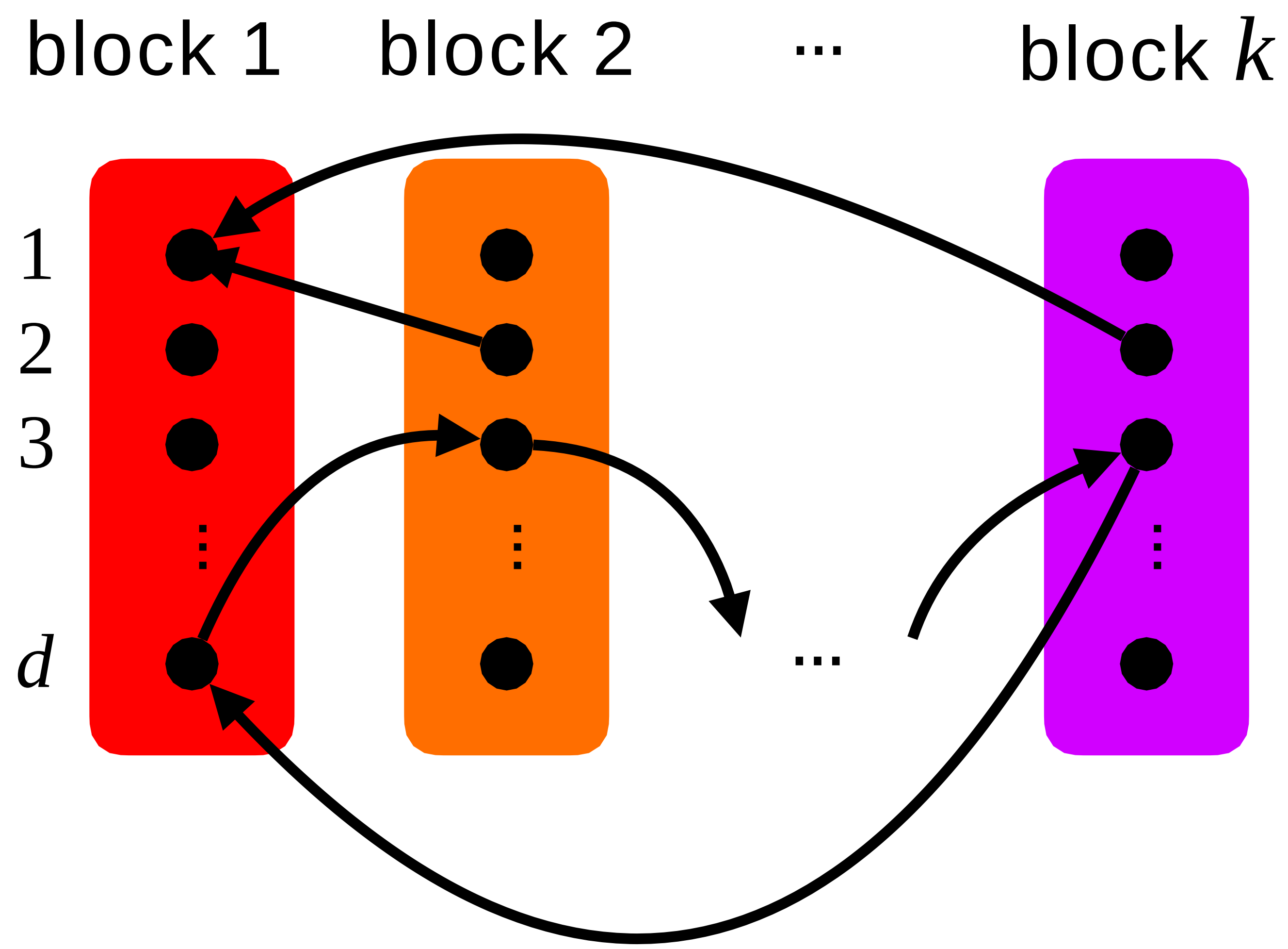
Does an EFX allocation always exist?

- **one of the most significant open questions in the field**
- Partial affirmative results include the cases
  - 2 agents [Plaut, Roughgarden 2018]
  - 3 agents, additive valuations [Chaudhury, Garg, Mehlhorn 2020]
- **Approximative EFX:**

α-EFX for $\alpha \in (0,1]$: $\forall i,j,\ \forall g \in A_j \quad : v_i(A_i) \geq \alpha \cdot v_i(A_j \setminus \{g\})$

The Rainbow Cycle Number

- The **rainbow cycle number** $R(d)$ is the largest integer k such that there exists a k-partite directed graph with each block of size d such that:
  - every vertex has an incoming edge from each other block (**in-property**)
  - there is **no rainbow cycle** (a cycle containing at most one vertex from each block)



α-EFX and the Rainbow Cycle Number

- Based on $R(d)$ one gets $(1-\varepsilon)$-EFX allocations with a sublinear number of unallocated items.

- **Theorem** [Chaudhury, Garg, Mehlhorn, Mehta, Misra 2021]

Let $\varepsilon \in (0,1/2]$ and let $g(y)$ be the smallest integer d such that $d \cdot R(d) \geq y$.

Then, there is a partial $(1-\varepsilon)$-EFX allocation with at most

$$\frac{4n}{\varepsilon \cdot g(2n/\varepsilon)}$$

many unallocated items.

Bounds on the Rainbow Cycle Number

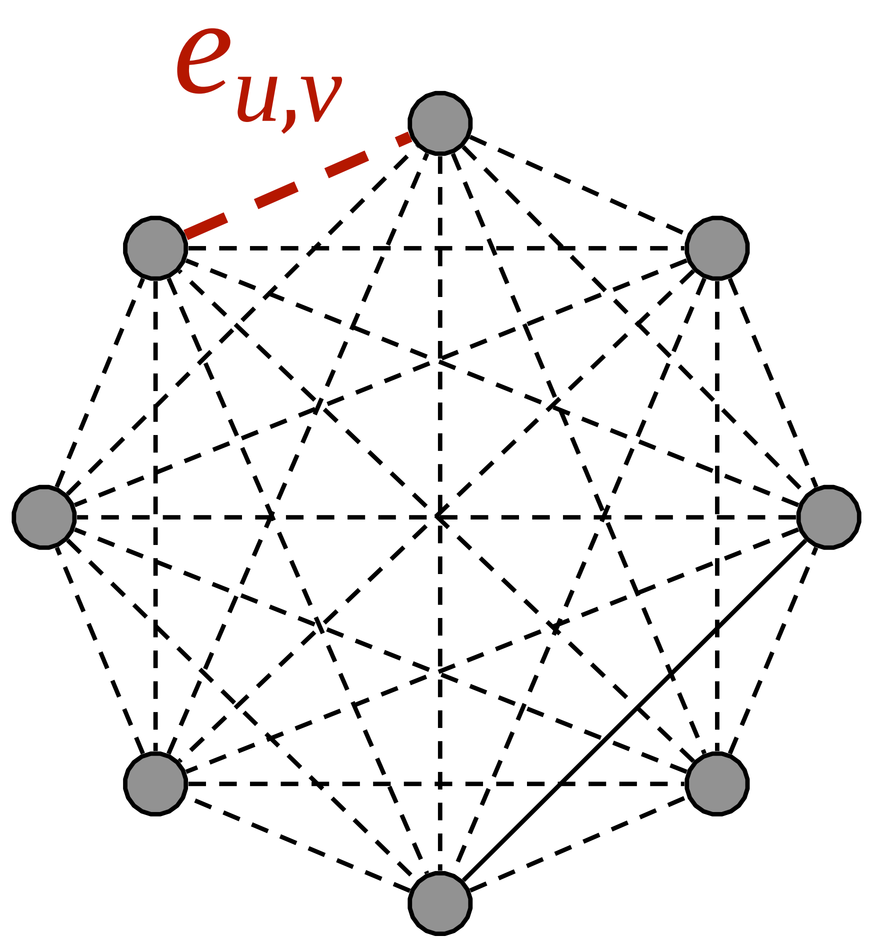
- [Chaudhury, Garg, Mehlhorn, Mehta, Misra 2021]
  - $d \leq R(d) \leq d^4 + d$
  - "We believe that finding better upper bounds on $R(d)$ is a natural combinatorial question"
  - "Better upper-bounds to $R(d)$ imply the existence of better relaxations of EFX allocations"
  - "Therefore investigating better upper bounds on the rainbow cycle number is of interest in its own right and we leave this as an interesting open problem"
- $R(2) = 2$
- $R(3) = 3$
- $R(4) \stackrel{?}{=} 4$ open

Conjecture: $R(d) = d$

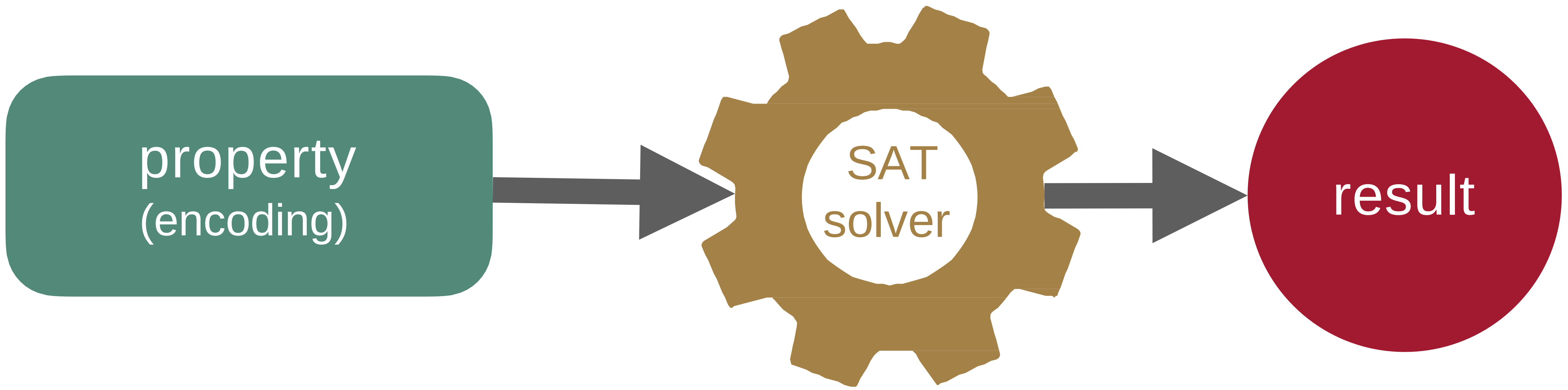
Showing $R(d) = d$ for small d

- checking that every $(d+1)$-partite graph with d vertices per block that satisfies the in-property contains a rainbow cycle.
- enumerate all such graphs modulo isomorphism, say with Nauty?
- $d = 4$ implies $n = 20$
- there are more than 7.03×10^{29} directed graphs with 14 vertices, modulo isomorphism
- generate-and-test not feasible!

Graph search as a synthesis problem

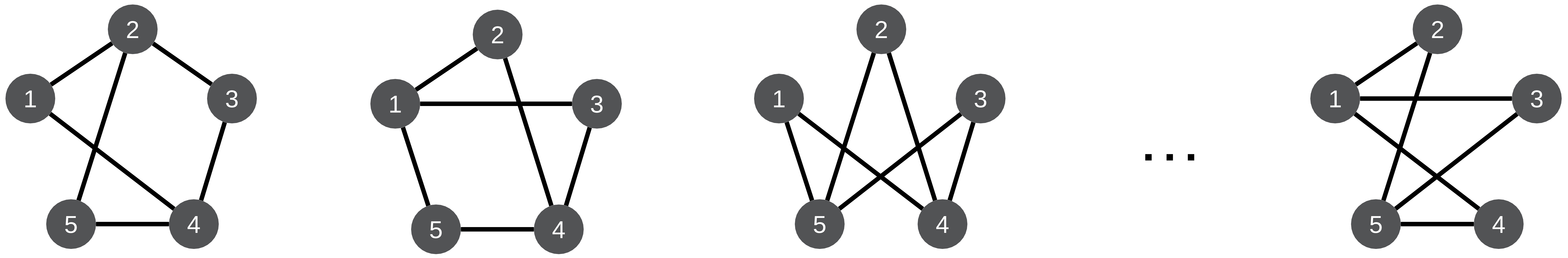


- We fix the number of vertices
- Each edge $\{u, v\}$ is represented by a propositional variable $e_{u,v}$ which is true iff the edge exists



Isomorph-Free Generation

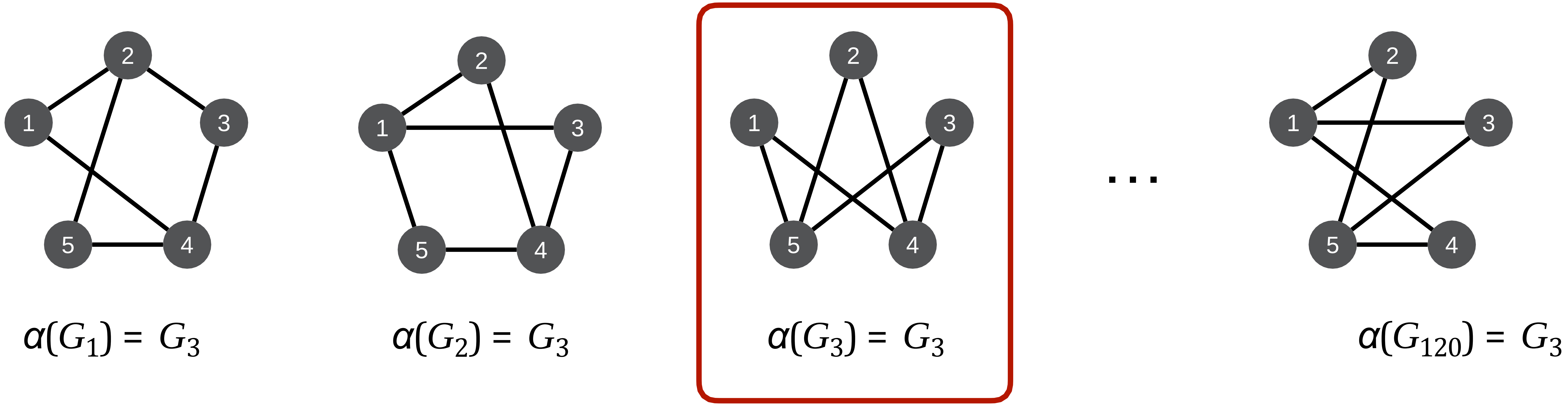
- **Isomorph-free generation:** Number of objects explode quickly
- **Canonization:** map each object to a unique representative $\alpha(G)$ of its isomorphism class



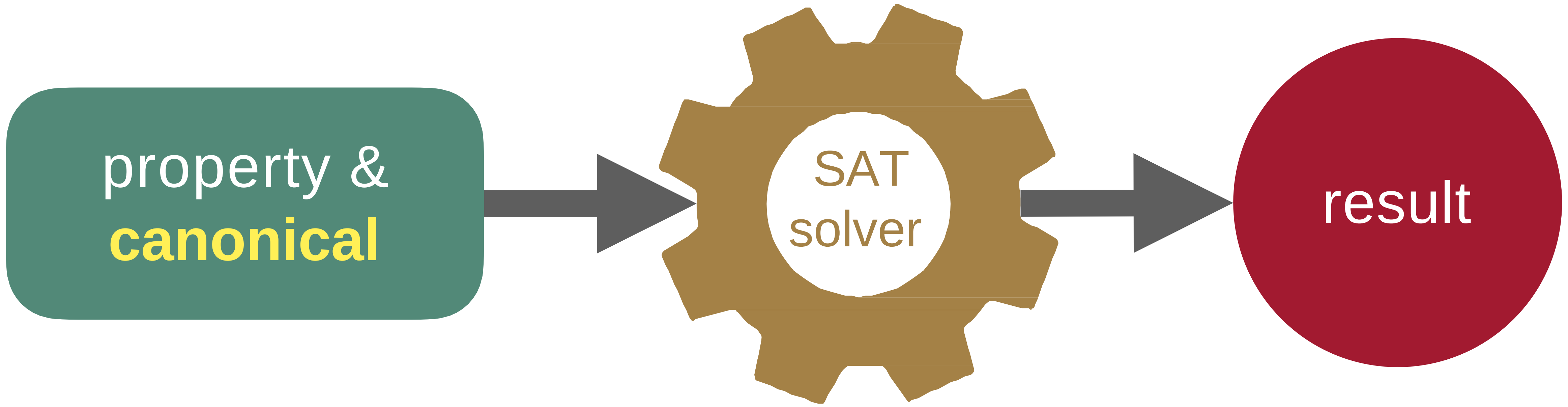
G_1 G_2 G_3 G_{120}

Isomorph-Free Generation

- **Isomorph-free generation:** Number of objects explode quickly
- **Canonization:** map each object to a unique representative $\alpha(G)$ of its isomorphism class
- **Canonical Objects:** Only generate objects with $\alpha(G) = G$

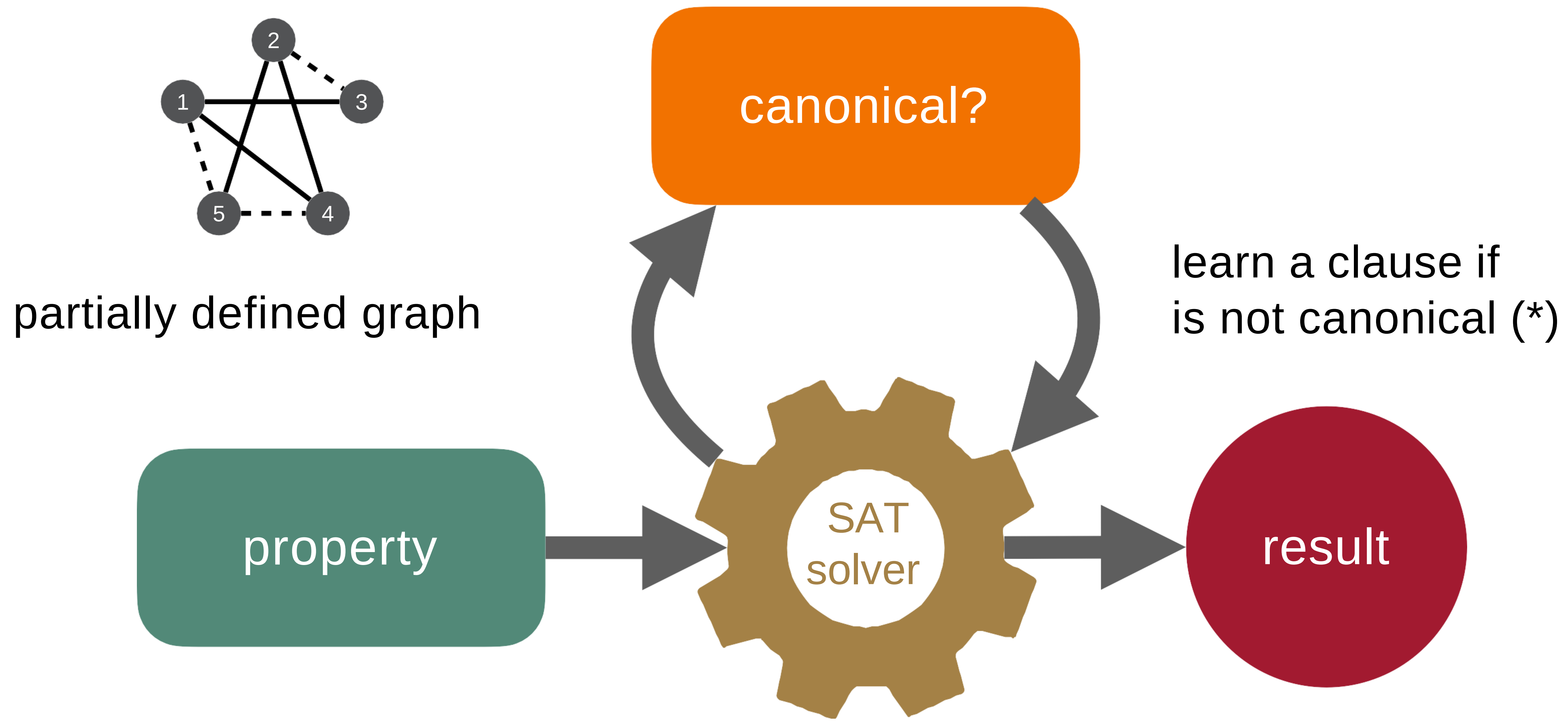


Static SAT approach



Problem: no polynomial size encoding for canonicity is known!

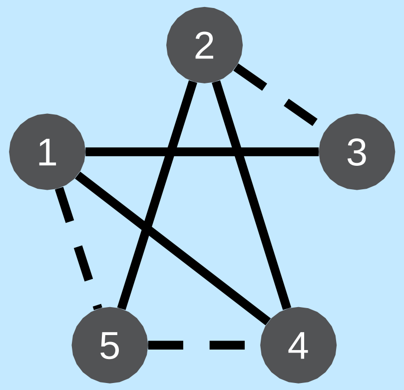
Dynamic SAT approach: SMS



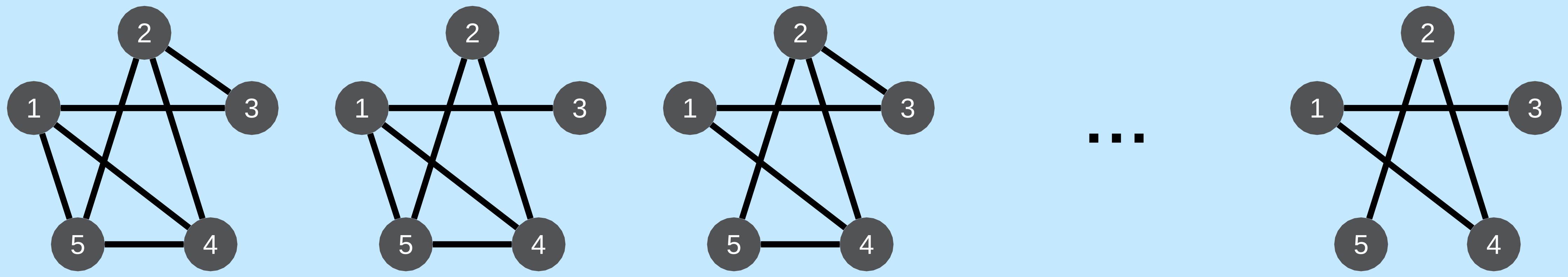
- SAT modulo Symmetries [Kirchweger, Szeider. CP 2021]
- IPASIR-UP interface [Fazekas et al. 2023]

Canonicity of partially defined graphs

partially defined graph P



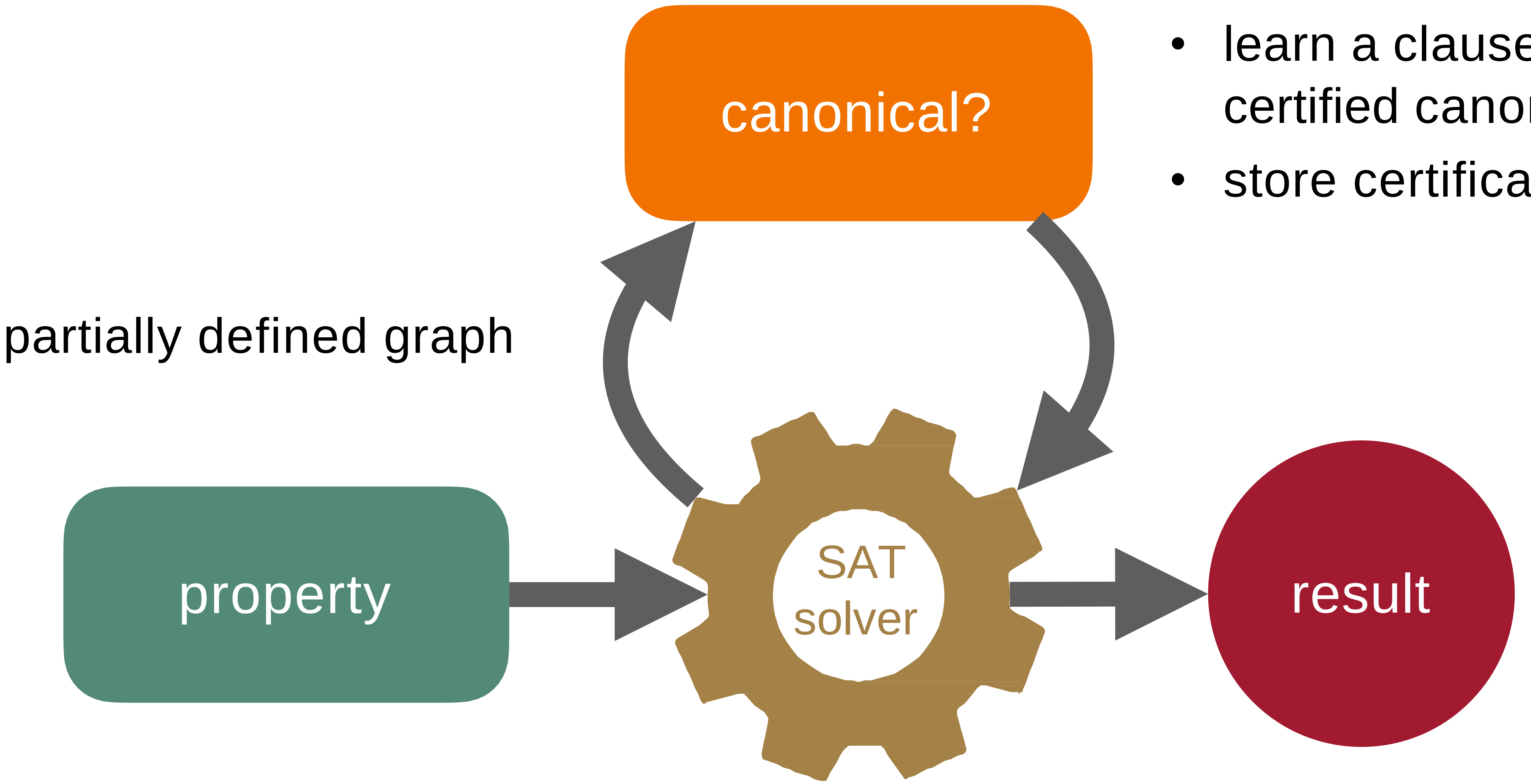
$X(P)$: set of all fully defined graphs P can be extended to



- is **non-canonical** if $\forall G \in X(P)\, \exists \pi : \pi(G) <_{\text{lex}} G$
- is **certified non-canonical** if $\exists \pi \forall G \in X(P) : \pi(G) <_{\text{lex}} G$

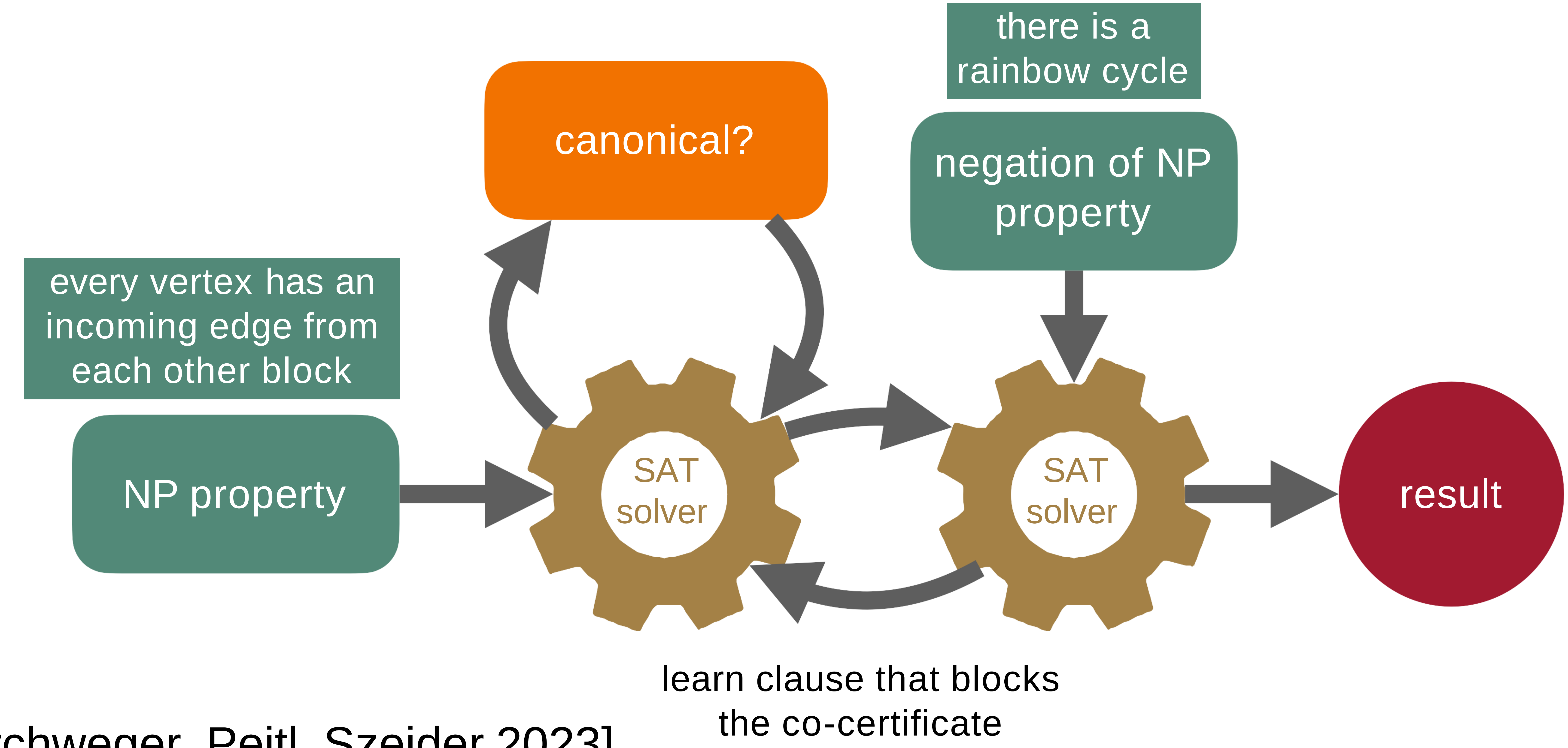
  we have an efficient constraint-propagation algorithm for computing π

Dynamic SAT approach: SMS



- learn a clause if is not certified canonical
- store certificate

SMS with co-certificate learning



- [Kirchweger, Peitl, Szeider 2023]

Results for showing $R(d) = d$

- “$R(3) \geq 4$” is unsatisfiable, within 1 second
- “$R(4) \geq 5$” is unsatisfiable, within 23 minutes
- “$R(5) \geq 6$” didn’t terminate within 300h

Invariant pruning

- assuming max indegree is $\Delta := d(k-1)$
- w.l.o.g., vertex 1 has indegree Δ
- if UNSAT, add constraints that limit indegrees to $\Delta - 1$ for all vertices

- assuming max indegree is $\Delta - 1$
- w.l.o.g., vertex 1 has indegree $\Delta - 1$
- if UNSAT, add constraints that limit indegrees to $\Delta - 2$

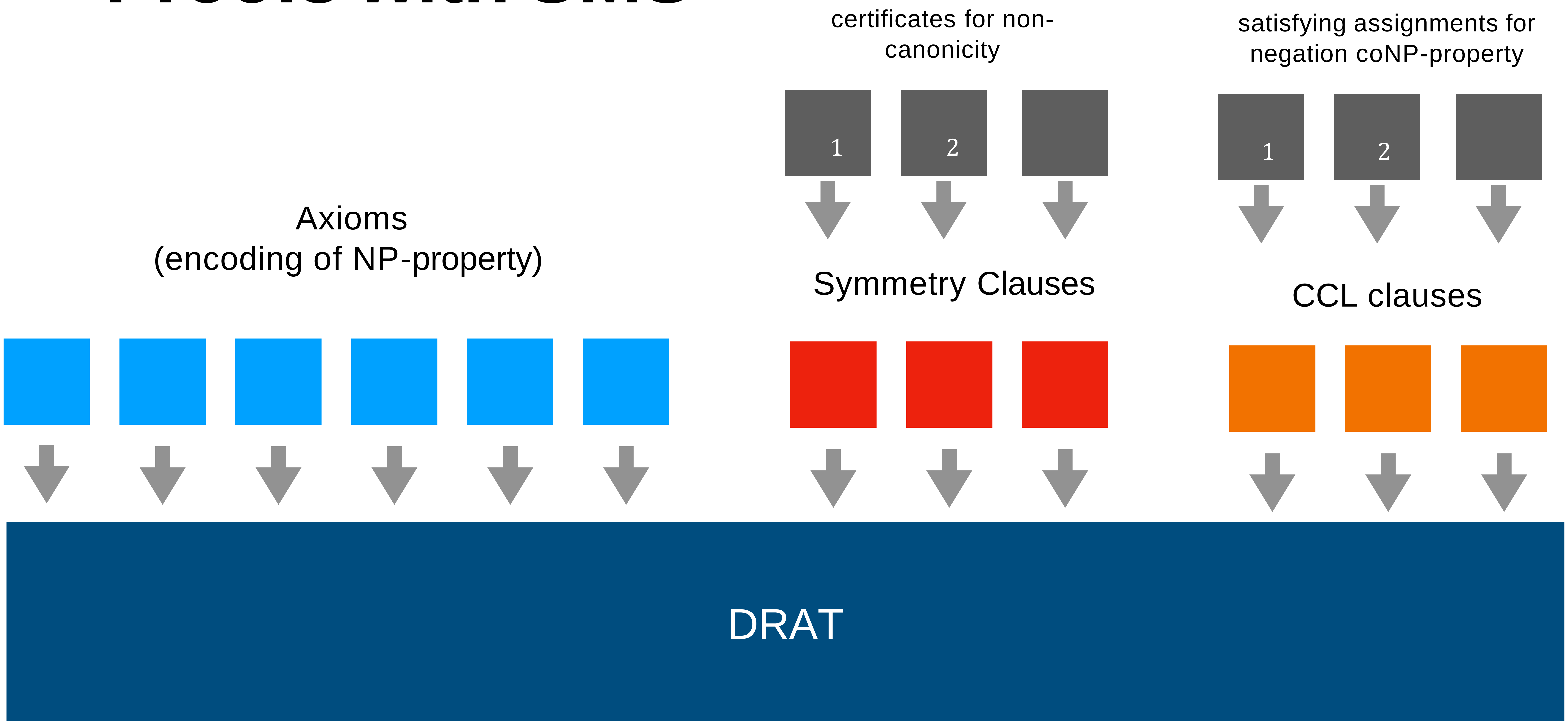
etc.

$R(d)$ with invariant pruning

- “$R(4) \geq 5$”
  - showing in-degree ≤ 4 within 3 seconds
  - showing unsatisfiability with in-degree ≤ 4 then takes half a minute
  - almost 50-fold speedup

- “$R(5) \geq 6$”
  - showing in-degree ≤ 6 within 105h
  - showing unsatisfiability with in-degree ≤ 6 didn’t terminate within 300h

Proofs with SMS



[Wetzler, Heule, Hunt 2014]

Summary

- Fair division of goods, EFX
- Connection between rainbow cycle numbers and α-EFX with charity
- Computing rainbow cycle numbers with SAT modulo Symmetries
- Determined $R(4) = 4$, with DRAT proof

Future Work:

- Settle case $R(5) = 5$ (mathematical insights?)
- Apply invariant pruning to other highly symmetric problems
- Try to compute counterexample to EFX

Resources

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

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