Theory of Constraint Satisfaction Problems

Antoine Wiehe

Colloquium Computational Discrete Mathematics

TU Graz

Instance:

- a domain A
- variables $x, y, z, \ldots$
- constraints: $x \vee y \vee z, z \leq x, 2x + y = 0, \ldots$

Question: does there exist $h\colon \mathrm{Vars} \to A$ satisfying every constraint?

Instance:

- a domain A
- variables $x, y, z, \ldots$
- constraints: $x \vee y \vee z, z \leq x, 2x + y = 0, \ldots$

Question: does there exist $h\colon \mathrm{Vars} \to A$ satisfying every constraint?

Examples. Boolean Satisfiability

Is $(x \vee y \vee z) \wedge (\neg z \vee t \vee p) \wedge \ldots$ satisfiable?

Instance:

- a domain A
- variables $x, y, z, \ldots$
- constraints: $x \vee y \vee z, z \leq x, 2x + y = 0, \ldots$

Question: does there exist $h\colon \text{Vars} \to A$ satisfying every constraint?

Examples. Boolean Satisfiability

Is $(x \vee y \vee z) \wedge (\neg z \vee t \vee p) \wedge \ldots$ satisfiable?

Solving Linear Equations

$$\text{Is } \left\{ \begin{array}{cccccc} 2x & -y & & = & 0 \\ & y & +z & = & 1 \\ x & & -z & = & 2 \end{array} \right. \text{ satisfiable?}$$

Instance:

- a domain A
- variables $x, y, z, \ldots$
- constraints: $x \vee y \vee z, z \leq x, 2x + y = 0, \ldots$

Question: does there exist $h\colon \text{Vars} \to A$ satisfying every constraint?

Examples. Boolean Satisfiability

Is $(x \vee y \vee z) \wedge (\neg z \vee t \vee p) \wedge \ldots$ satisfiable?

NP-complete

Solving Linear Equations

Is $\left\{ \begin{array}{cccccc} 2x & -y & & = & 0 \\ & y & +z & = & 1 \\ x & & -z & = & 2 \end{array} \right.$ satisfiable?

Solvable in polynomial time

Instance:
- a domain A
- variables $x, y, z, \ldots$
- constraints: $x \vee y \vee z, z \leq x, 2x + y = 0, \ldots$

Question: does there exist $h\colon \text{Vars} \to A$ satisfying every constraint?

Examples. Boolean Satisfiability

Is $(x \vee y \vee z) \wedge (\neg z \vee t \vee p) \wedge \ldots$ satisfiable?

NP-complete

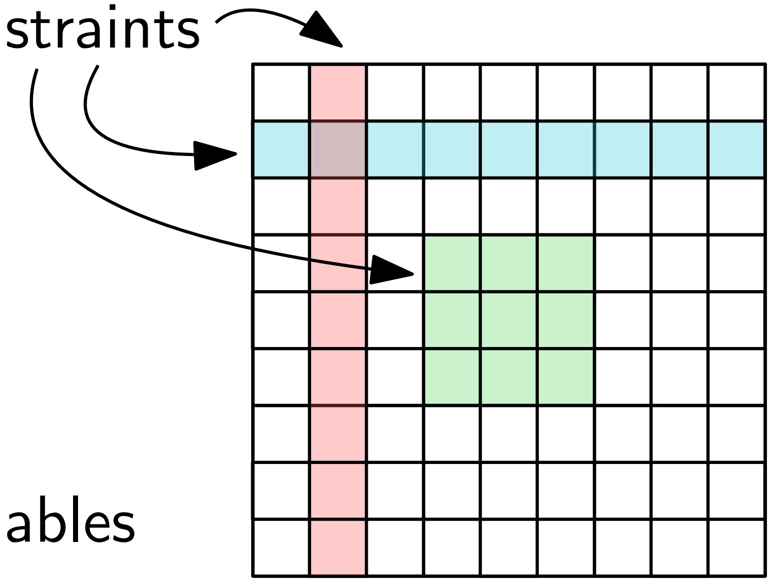
Solving Linear Equations

Is $\left\{\begin{array}{rrrcl} 2x & -y & & = & 0 \\ & y & +z & = & 1 \\ x & & -z & = & 2 \end{array}\right.$ satisfiable?

Solvable in polynomial time

Sudoku

9-ary constraints



9^2 variables

Linear Programming

Is $\mathbf{A}x = \mathbf{b}$ satisfiable in $\mathbb{R}_{\geq 0}$?

Instance:

- a domain A
- variables $x, y, z, \ldots$
- constraints: $x \vee y \vee z, z \leq x, 2x + y = 0, \ldots$

Question: does there exist $h\colon \mathrm{Vars} \to A$ satisfying every constraint?

- Non-uniform CSP: domain is fixed and only certain types of constraints are allowed

  Constraint with r variables $\leftrightarrow$ relation $R \subseteq A^r$ listing all valid assignments

  Constraint language / template: $\mathbb{A} = (A; R_1, \ldots, R_k)$

Instance:

- ~~a domain A~~
- variables $x, y, z, \ldots$
- constraints: $R_1(x, y, z), R_2(z, x), R_3(x, y), R_2(z, y), \ldots$

Question: does there exist $h\colon \text{Vars} \to A$ satisfying every constraint?

- Non-uniform CSP: domain is fixed and only certain types of constraints are allowed

  Constraint with r variables $\leftrightarrow$ relation $R \subseteq A^r$ listing all valid assignments

  Constraint language / template: $\mathbb{A} = (A; R_1, \ldots, R_k)$

Instance:
- ~~a domain A~~
- variables $x, y, z, \ldots$
- constraints: $R_1(x, y, z), R_2(z, x), R_3(x, y), R_2(z, y), \ldots$

Question: does there exist $h\colon \text{Vars} \to A$ satisfying every constraint?

- Non-uniform CSP: domain is fixed and only certain types of constraints are allowed

  Constraint with r variables $\leftrightarrow$ relation $R \subseteq A^r$ listing all valid assignments

  Constraint language / template: $\mathbb{A} = (A; R_1, \ldots, R_k)$

$\textsc{1-In-3-Sat} : (\{0, 1\}; \{(1, 0, 0), (0, 1, 0), (0, 0, 1)\})$
$n\text{-}\textsc{Coloring} : (\{1, \ldots, n\}, \neq)$
$\textsc{Acyclicity} : (\mathbb{Q}; <)$

Instance:

- ~~a domain A~~
- variables $x, y, z, \ldots$
- constraints: $R_1(x, y, z), R_2(z, x), R_3(x, y), R_2(z, y), \ldots$

Question: does there exist $h\colon \mathrm{Vars} \to A$ satisfying every constraint?

- Non-uniform CSP: domain is fixed and only certain types of constraints are allowed

  Constraint with r variables $\leftrightarrow$ relation $R \subseteq A^r$ listing all valid assignments

  Constraint language / template: $\mathbb{A} = (A; R_1, \ldots, R_k)$

$\textsc{1-In-3-Sat} : (\{0, 1\}; \{(1, 0, 0), (0, 1, 0), (0, 0, 1)\})$

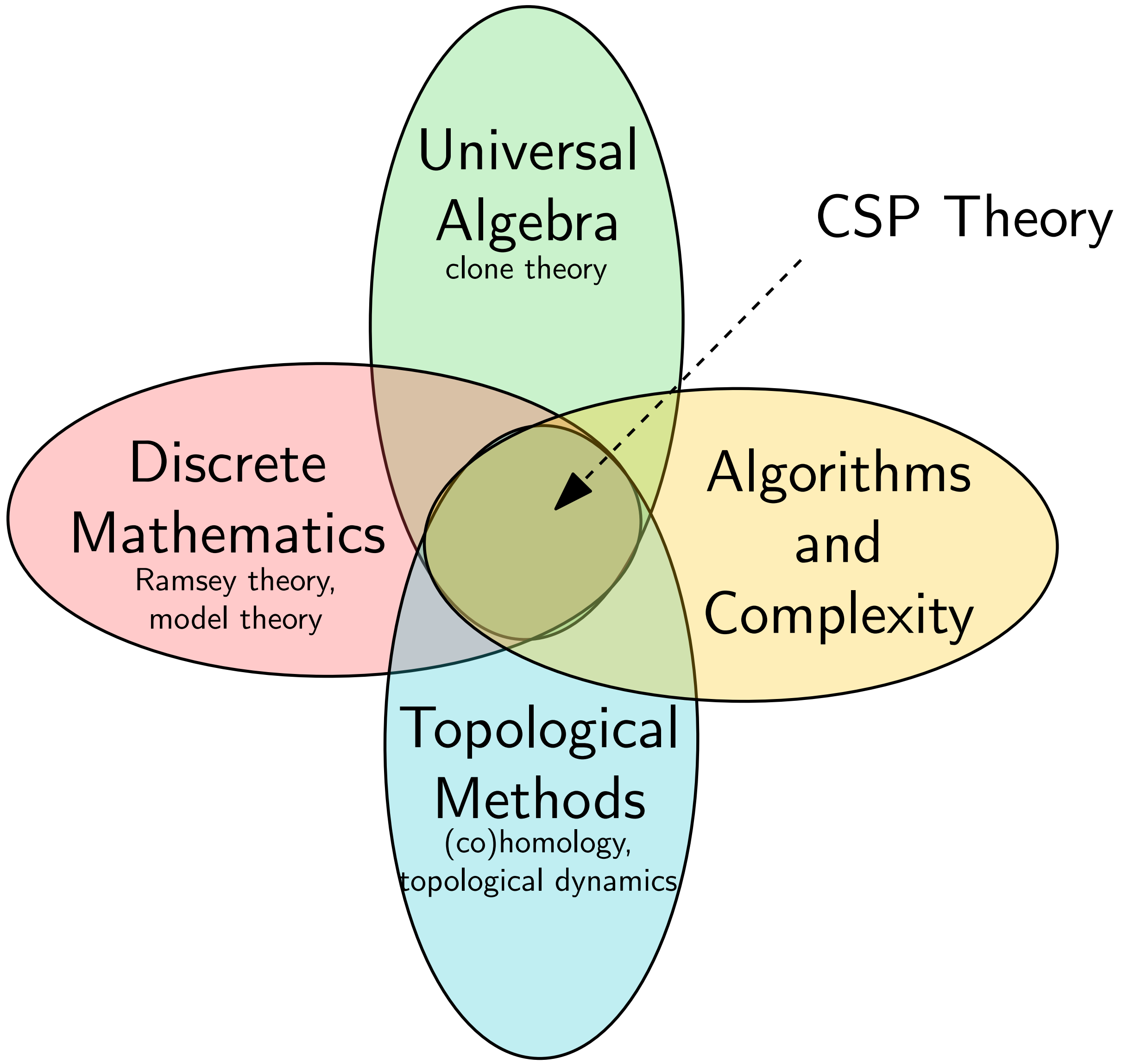
$n\textsc{-Coloring} : (\{1, \ldots, n\}, \neq)$

$\textsc{Acyclicity} : (\mathbb{Q}; <)$

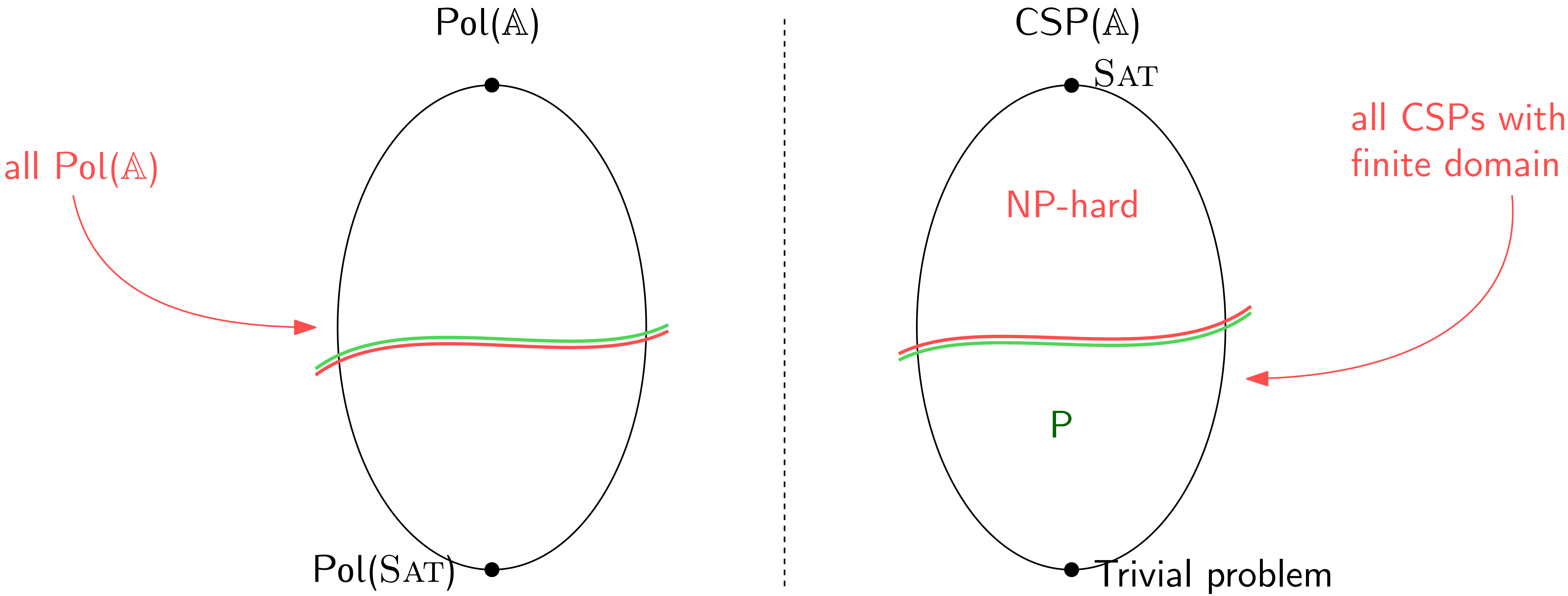
Theorem (Bulatov // Zhuk '17)**.** For every template $\mathbb{A}$ with a finite domain, CSP($\mathbb{A}$) is solvable in polynomial time or NP-complete.

Finite-domain dichotomy result built on polymorphisms $\mathrm{Pol}(\mathbb{A}) = \{f \colon \mathbb{A}^n \to \mathbb{A}, n \geq 1\}$

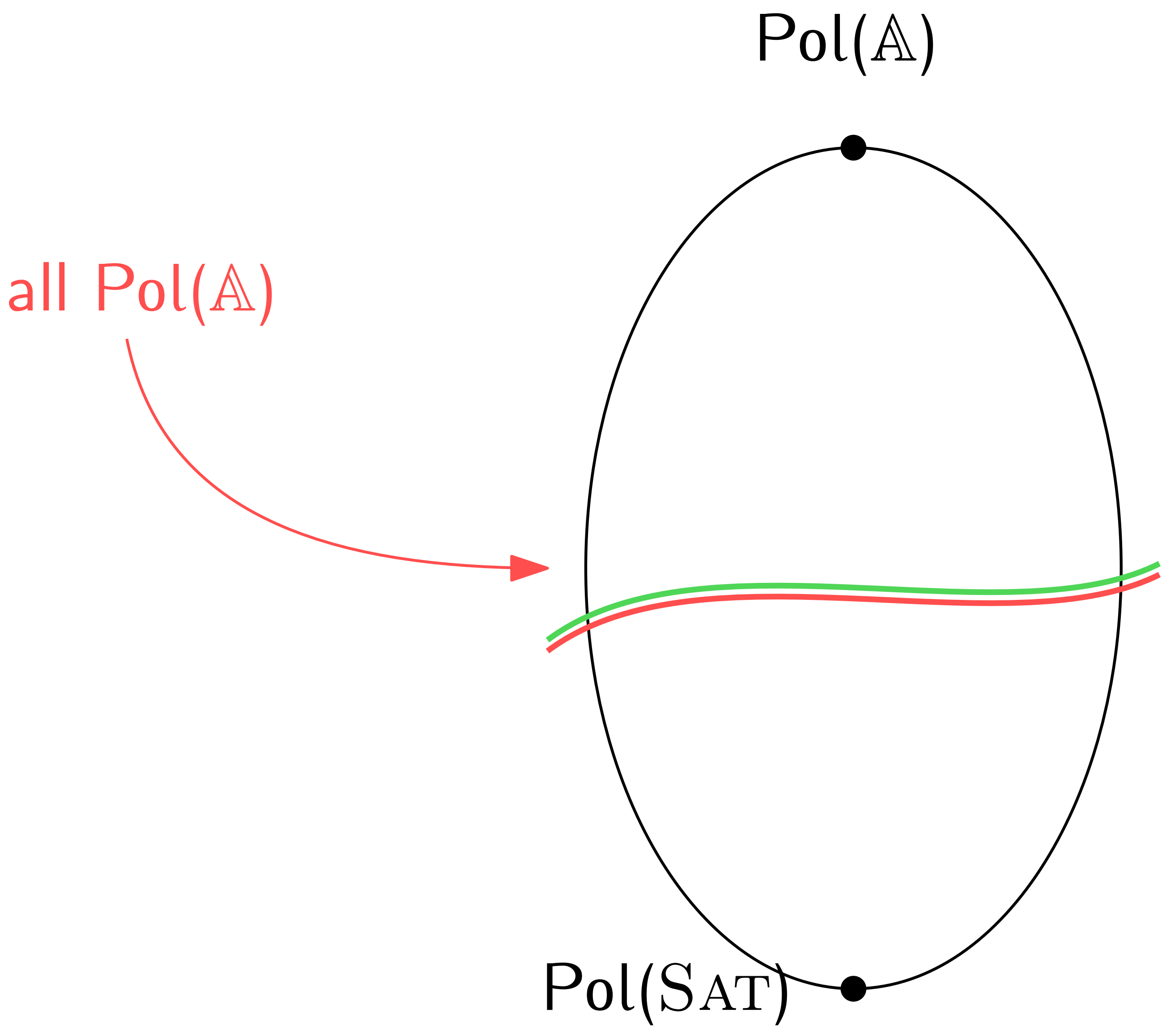
- Clones: closed under composition, contain projections
- Standard object in universal algebra
- Clone actions of $\mathrm{Pol}(\mathbb{A})$ tell us about the complexity of $\mathrm{CSP}(\mathbb{A})$

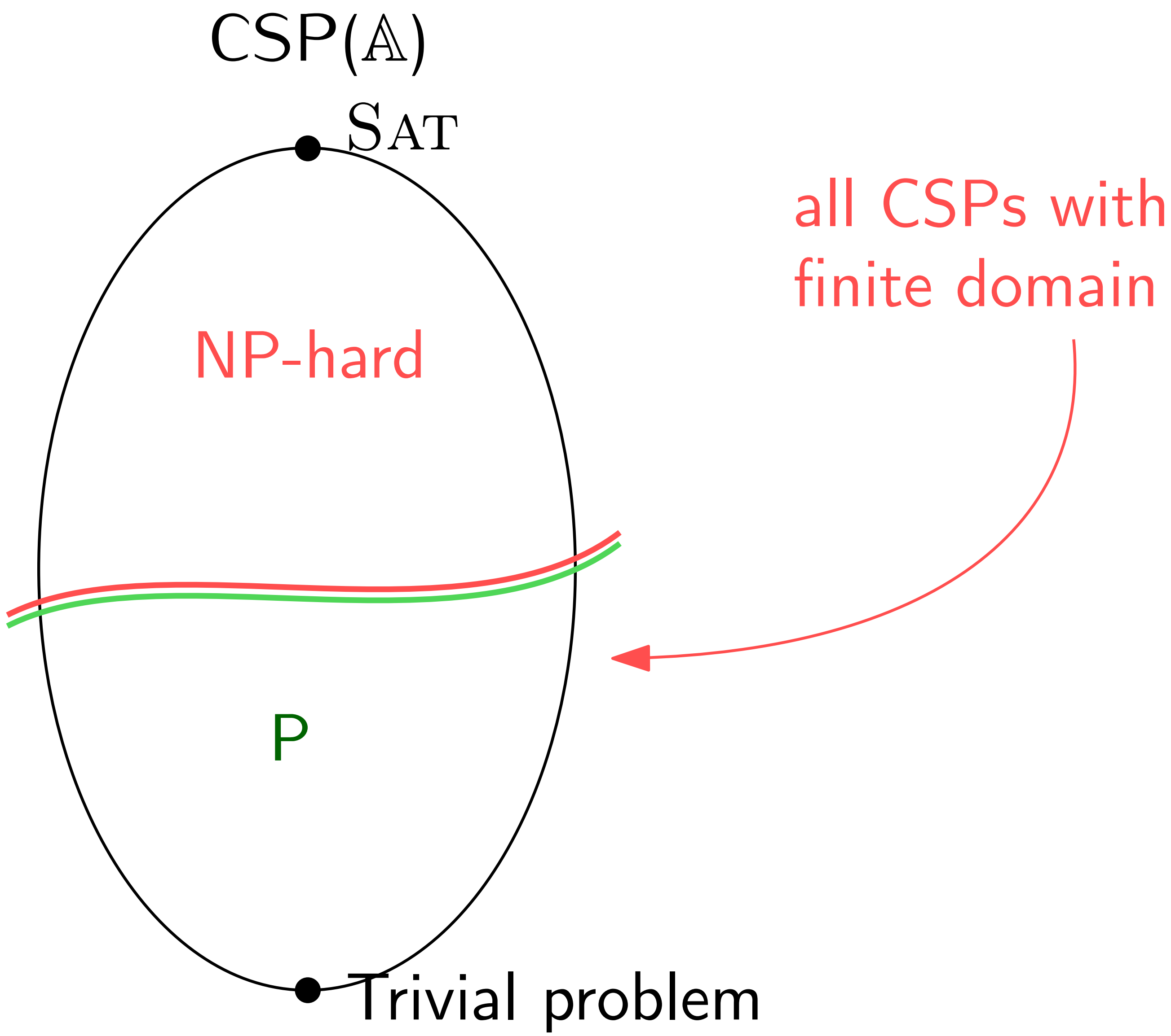


Finite-domain dichotomy result built on polymorphisms $\mathrm{Pol}(\mathbb{A}) = \{f\colon \mathbb{A}^n \to \mathbb{A}, n \geq 1\}$



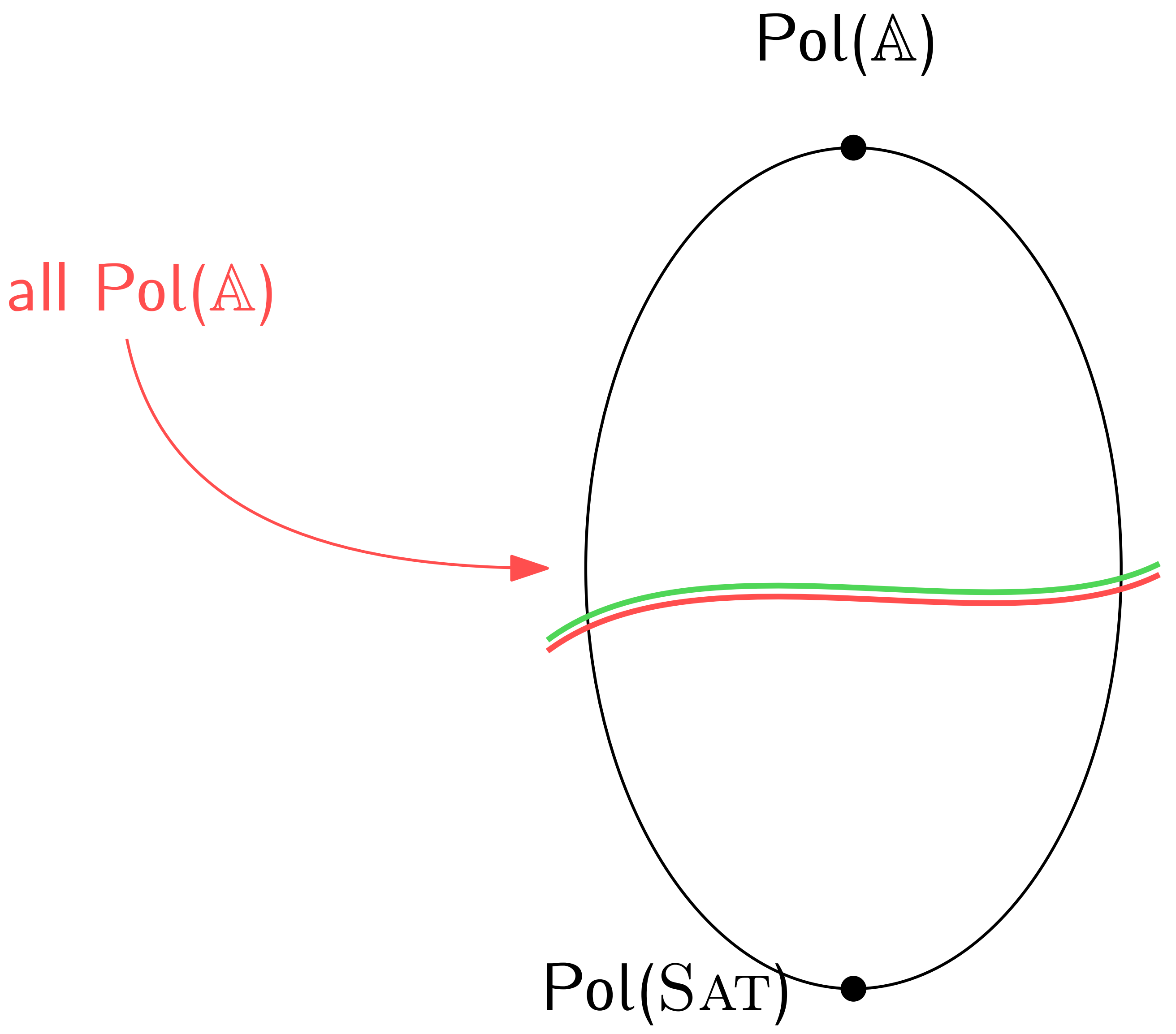
Finite-domain dichotomy result built on polymorphisms $\mathrm{Pol}(\mathbb{A}) = \{f : \mathbb{A}^n \to \mathbb{A}, n \geq 1\}$

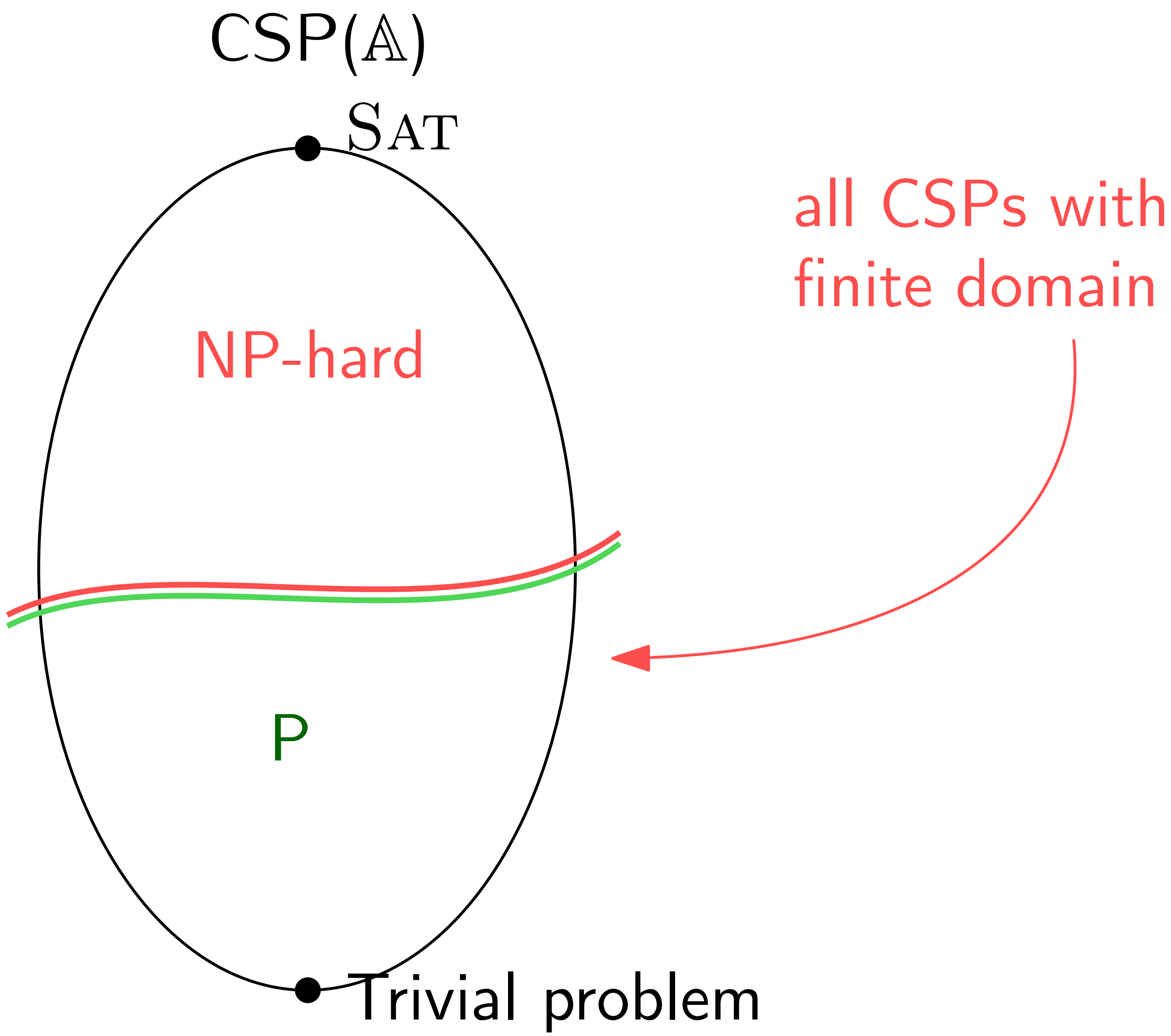




- $\mathrm{Pol}(\mathbb{A}) = \mathrm{Pol}(\mathbb{B})$ implies $\mathrm{CSP}(\mathbb{A}) \equiv \mathrm{CSP}(\mathbb{B})$ (Bulatov, Jeavons, Krokhin)

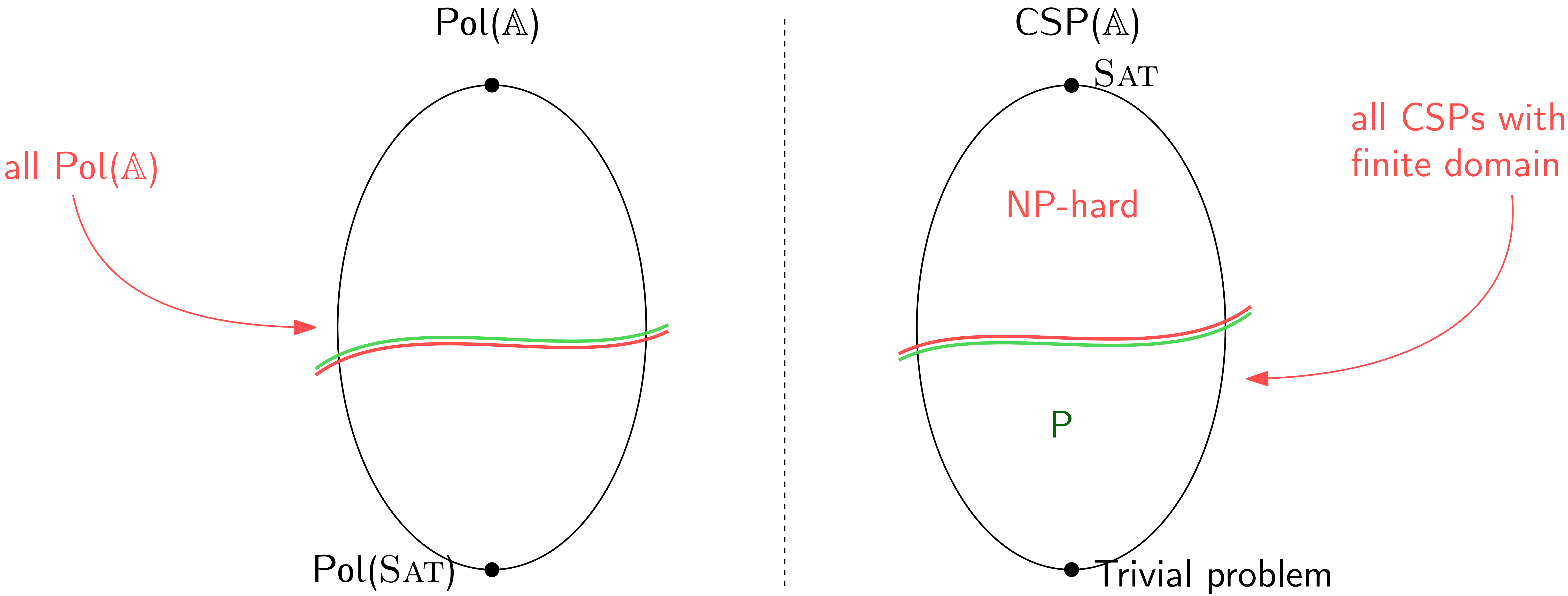
Finite-domain dichotomy result built on polymorphisms $\mathrm{Pol}(\mathbb{A}) = \{f : \mathbb{A}^n \to \mathbb{A}, n \geq 1\}$





- $\mathrm{Pol}(\mathbb{A}) \subseteq \mathrm{Pol}(\mathbb{B})$ implies $\mathrm{CSP}(\mathbb{B}) \leq \mathrm{CSP}(\mathbb{A})$ (Bulatov, Jeavons, Krokhin)

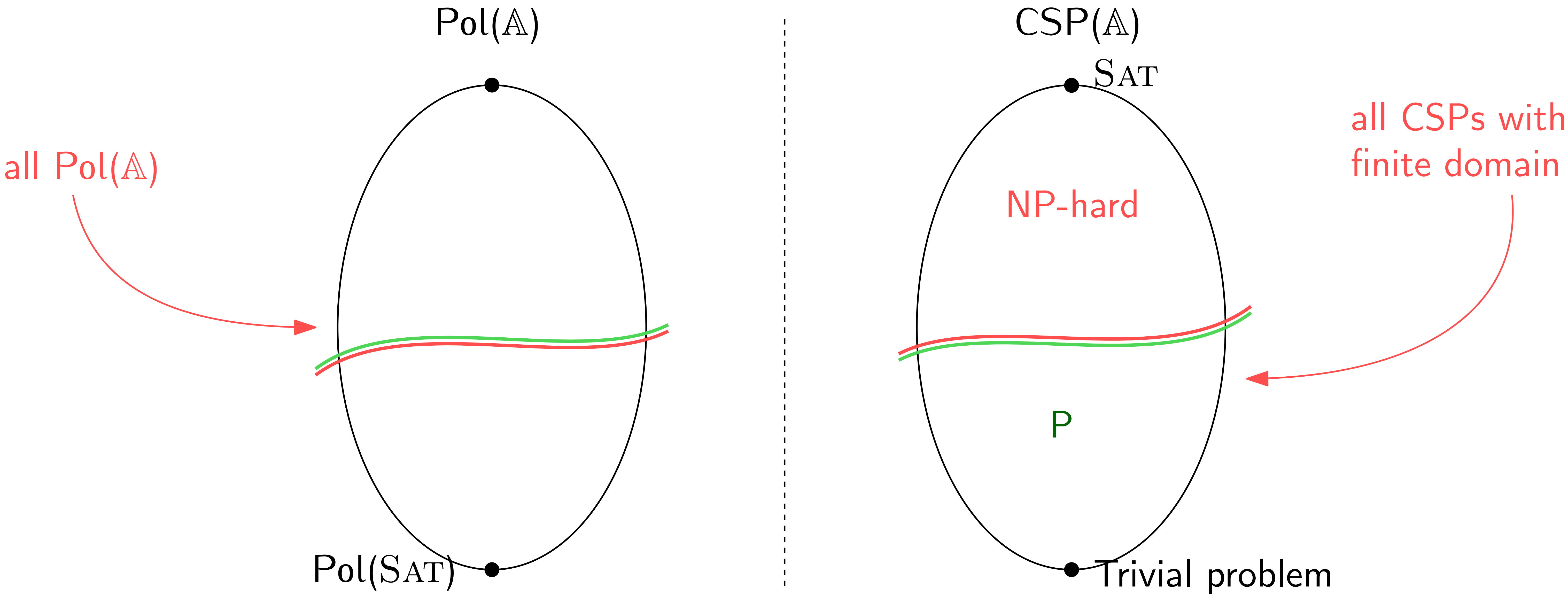
Finite-domain dichotomy result built on polymorphisms $\mathrm{Pol}(\mathbb{A}) = \{f \colon \mathbb{A}^n \to \mathbb{A}, n \geq 1\}$



- $\mathrm{Pol}(\mathbb{A}) \to \mathrm{Pol}(\mathbb{B})$ implies $\mathrm{CSP}(\mathbb{B}) \leq \mathrm{CSP}(\mathbb{A})$

(Barto, Opršal, Pinsker)

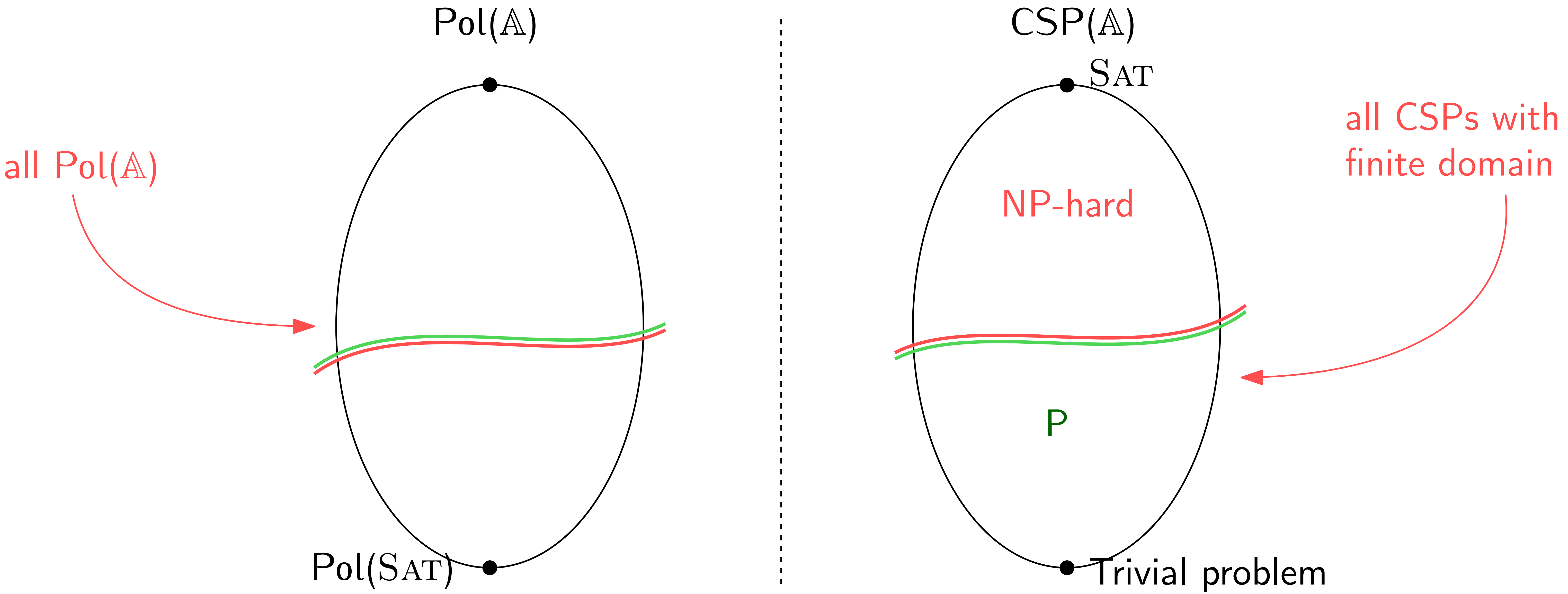
Finite-domain dichotomy result built on polymorphisms $\mathrm{Pol}(\mathbb{A}) = \{f \colon \mathbb{A}^n \to \mathbb{A}, n \geq 1\}$



- $\mathrm{Pol}(\mathbb{A}) \to \mathrm{Pol}(\mathbb{B})$ implies $\mathrm{CSP}(\mathbb{B}) \leq \mathrm{CSP}(\mathbb{A})$
- $\mathrm{Pol}(\mathbb{A}) \to \mathrm{Pol}(\textsc{Sat})$ implies $\mathrm{CSP}(\mathbb{A})$ is NP-hard

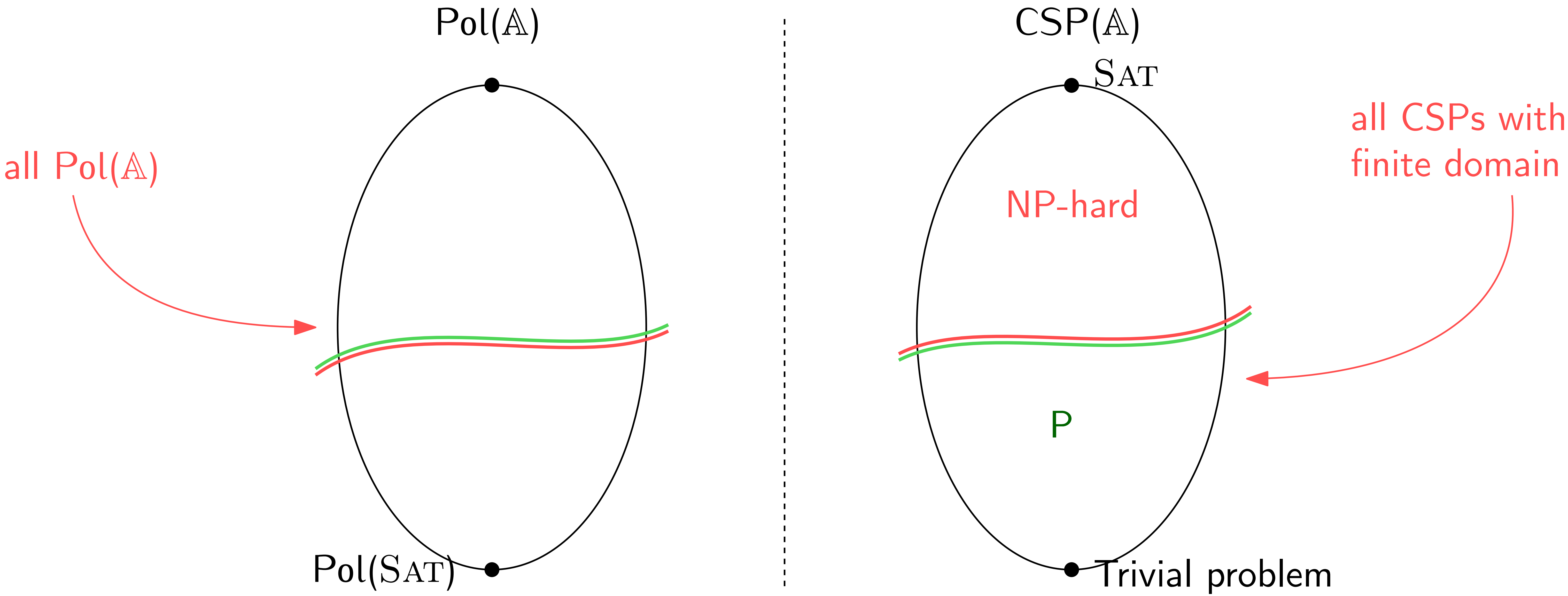
(Barto, Opršal, Pinsker)

Finite-domain dichotomy result built on polymorphisms $\mathrm{Pol}(\mathbb{A}) = \{f : \mathbb{A}^n \to \mathbb{A}, n \geq 1\}$



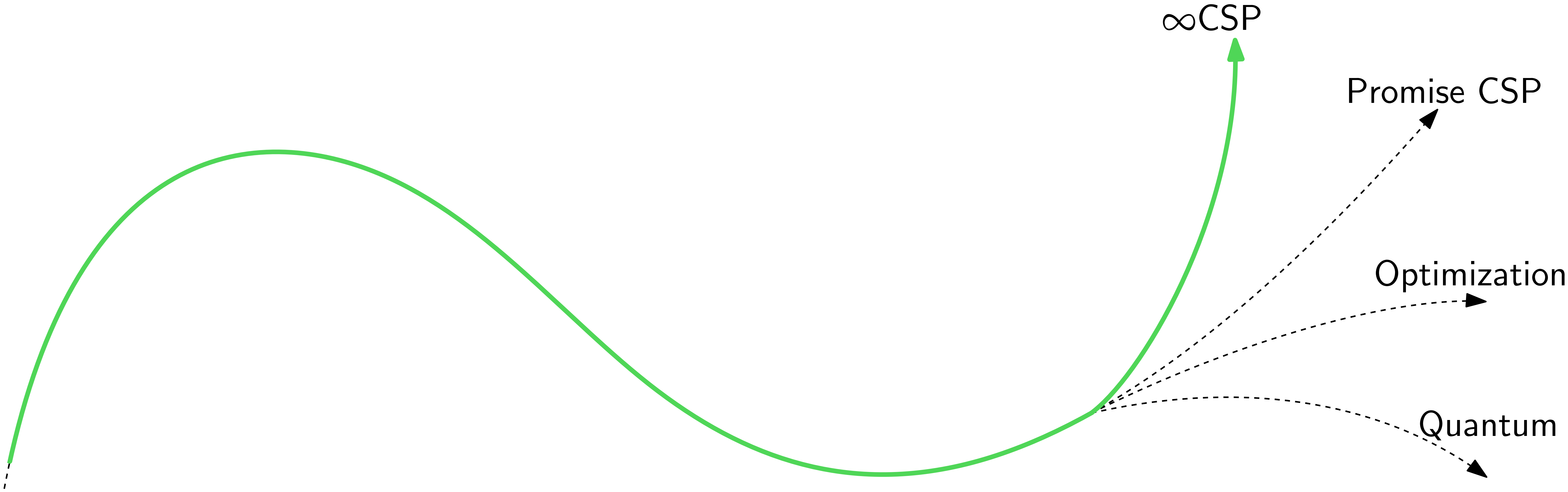
- $\mathrm{Pol}(\mathbb{A}) \to \mathrm{Pol}(\mathbb{B})$ implies $\mathrm{CSP}(\mathbb{B}) \leq \mathrm{CSP}(\mathbb{A})$ (Barto, Opršal, Pinsker)
- $\mathrm{Pol}(\mathbb{A}) \to \mathrm{Pol}(\textsc{Sat})$ implies $\mathrm{CSP}(\mathbb{A})$ is NP-hard
- If $\mathrm{Pol}(\mathbb{A}) \not\to \mathrm{Pol}(\textsc{Sat})$, then $\mathrm{CSP}(\mathbb{A})$ is solvable in polynomial time (Bulatov // Zhuk)

Finite-domain dichotomy result built on polymorphisms $\mathrm{Pol}(\mathbb{A}) = \{f\colon \mathbb{A}^n \to \mathbb{A}, n \geq 1\}$



- $\mathrm{Pol}(\mathbb{A}) \to \mathrm{Pol}(\mathbb{B})$ implies $\mathrm{CSP}(\mathbb{B}) \leq \mathrm{CSP}(\mathbb{A})$ (Barto, Opršal, Pinsker)
- $\mathrm{Pol}(\mathbb{A}) \to \mathrm{Pol}(\textsc{Sat})$ implies $\mathrm{CSP}(\mathbb{A})$ is NP-hard
- If $\mathrm{Pol}(\mathbb{A}) \not\to \mathrm{Pol}(\textsc{Sat})$, then $\mathrm{CSP}(\mathbb{A})$ is solvable in polynomial time (Bulatov // Zhuk)
- Can the algebraic approach be leveraged for other computational problems?

Research in infinite-domain CSPs



- Reasoning: given a collection Φ of facts and an atom ψ, does Φ imply ψ?

- Reasoning: given a collection Φ of facts and an atom ψ, does Φ imply ψ?
- Same as unsatisfiability of $\Phi \cup \{\neg\psi\}$ $\quad\leftrightsquigarrow\quad$ constraint satisfaction problem

- Reasoning: given a collection Φ of facts and an atom ψ, does Φ imply ψ?
- Same as unsatisfiability of $\Phi \cup \{\neg\psi\} \quad \leftrightsquigarrow \quad$ constraint satisfaction problem
- In spatial/temporal reasoning, variables are coordinates and the facts/atom are of the form $x < y, y = x + 5, \ldots$

- Reasoning: given a collection Φ of facts and an atom ψ, does Φ imply ψ?
- Same as unsatisfiability of $\Phi \cup \{\neg\psi\}$ $\leftrightsquigarrow$ constraint satisfaction problem
- In spatial/temporal reasoning, variables are coordinates and the facts/atom are of the form $x < y, y = x + 5, \ldots$

Examples.

Ordering constraints	AND/OR scheduling	Betweenness

- Reasoning: given a collection Φ of facts and an atom ψ, does Φ imply ψ?
- Same as unsatisfiability of $\Phi \cup \{\neg\psi\}$ $\leftrightsquigarrow$ constraint satisfaction problem
- In spatial/temporal reasoning, variables are coordinates and the facts/atom are of the form $x < y, y = x + 5, \ldots$

Examples.

Ordering constraints

$$\begin{cases} x & < & y \\ y & < & z \\ z & < & t \\ t & < & x \end{cases}$$

AND/OR scheduling

Betweenness

- Reasoning: given a collection Φ of facts and an atom ψ, does Φ imply ψ?
- Same as unsatisfiability of $\Phi \cup \{\neg\psi\}$ $\leftrightsquigarrow$ constraint satisfaction problem
- In spatial/temporal reasoning, variables are coordinates and the facts/atom are of the form $x < y, y = x + 5, \ldots$

Examples.

Ordering constraints

$$\begin{cases} x & < & y \\ y & < & z \\ z & < & t \\ t & < & x \end{cases}$$

In P: check if constraint graph is acyclic

AND/OR scheduling

Betweenness

- Reasoning: given a collection Φ of facts and an atom ψ, does Φ imply ψ?
- Same as unsatisfiability of $\Phi \cup \{\neg\psi\}$ $\quad\leftrightsquigarrow\quad$ constraint satisfaction problem
- In spatial/temporal reasoning, variables are coordinates and the facts/atom are of the form $x < y, y = x + 5, \ldots$

Examples.

Ordering constraints

$$\begin{cases} x & < & y \\ y & < & z \\ z & < & t \\ t & < & x \end{cases}$$

In P: check if constraint graph is acyclic

AND/OR scheduling

$$\begin{cases} x & < & \max(y, z) \\ y & < & t \\ t & < & \max(y, x) \end{cases}$$

Betweenness

- Reasoning: given a collection Φ of facts and an atom ψ, does Φ imply ψ?
- Same as unsatisfiability of $\Phi \cup \{\neg\psi\}$ $\leftrightsquigarrow$ constraint satisfaction problem
- In spatial/temporal reasoning, variables are coordinates and the facts/atom are of the form $x < y, y = x + 5, \ldots$

Examples.

Ordering constraints

$$\begin{cases} x & < & y \\ y & < & z \\ z & < & t \\ t & < & x \end{cases}$$

In P: check if constraint graph is acyclic

AND/OR scheduling

$$\begin{cases} x & < & \max(y, z) \\ y & < & t \\ t & < & \max(y, x) \end{cases}$$

In P: essentially arc consistency

Betweenness

- Reasoning: given a collection Φ of facts and an atom ψ, does Φ imply ψ?
- Same as unsatisfiability of $\Phi \cup \{\neg\psi\}$ $\leftrightsquigarrow$ constraint satisfaction problem
- In spatial/temporal reasoning, variables are coordinates and the facts/atom are of the form $x < y, y = x + 5, \ldots$

Examples.

Ordering constraints

$$\begin{cases} x & < & y \\ y & < & z \\ z & < & t \\ t & < & x \end{cases}$$

In P: check if constraint graph is acyclic

AND/OR scheduling

$$\begin{cases} x & < & \max(y, z) \\ y & < & t \\ t & < & \max(y, x) \end{cases}$$

In P: essentially arc consistency

Betweenness

$$\begin{cases} x < y < z & \vee & z < y < x \\ y < z < t & \vee & t < z < y \\ x < z < t & \vee & t < z < y \end{cases}$$

- Reasoning: given a collection Φ of facts and an atom ψ, does Φ imply ψ?
- Same as unsatisfiability of $\Phi \cup \{\neg\psi\}$ $\leftrightsquigarrow$ constraint satisfaction problem
- In spatial/temporal reasoning, variables are coordinates and the facts/atom are of the form $x < y, y = x + 5, \ldots$

Examples.

Ordering constraints

$$\begin{cases} x & < & y \\ y & < & z \\ z & < & t \\ t & < & x \end{cases}$$

In P: check if constraint graph is acyclic

AND/OR scheduling

$$\begin{cases} x & < & \max(y,z) \\ y & < & t \\ t & < & \max(y,x) \end{cases}$$

In P: essentially arc consistency

Betweenness

$$\begin{cases} x < y < z & \vee & z < y < x \\ y < z < t & \vee & t < z < y \\ x < z < t & \vee & t < z < y \end{cases}$$

NP-complete

- Reasoning: given a collection Φ of facts and an atom ψ, does Φ imply ψ?
- Same as unsatisfiability of $\Phi \cup \{\neg\psi\}$ $\leftrightsquigarrow$ constraint satisfaction problem
- In spatial/temporal reasoning, variables are coordinates and the facts/atom are of the form $x < y, y = x + 5, \ldots$

Examples.

Ordering constraints

$$\begin{cases} x & \leq & y+1 \\ y & \leq & z-1 \\ z & \leq & t+3 \\ t & \leq & x-4 \end{cases}$$

In P: check if constraint graph has negative cycles

AND/OR scheduling

$$\begin{cases} x & < & \max(y,z) \\ y & < & t \\ t & < & \max(y,x) \end{cases}$$

In P: essentially arc consistency

Betweenness

$$\begin{cases} x < y < z & \vee & z < y < x \\ y < z < t & \vee & t < z < y \\ x < z < t & \vee & t < z < y \end{cases}$$

NP-complete

- Reasoning: given a collection Φ of facts and an atom ψ, does Φ imply ψ?
- Same as unsatisfiability of $\Phi \cup \{\neg\psi\}$ $\leftrightsquigarrow$ constraint satisfaction problem
- In spatial/temporal reasoning, variables are coordinates and the facts/atom are of the form $x < y, y = x + 5, \ldots$

Examples.

Ordering constraints

$$\begin{cases} x & \leq & y+1 \\ y & \leq & z-1 \\ z & \leq & t+3 \\ t & \leq & x-4 \end{cases}$$

In P: check if constraint graph has negative cycles

AND/OR scheduling

$$\begin{cases} x & < & \max(y,z) \\ y & < & t \\ t & < & \max(y,x) \end{cases}$$

In P: essentially arc consistency

Betweenness

$$\begin{cases} x < y < z & \vee & z < y < x \\ y < z < t & \vee & t < z < y \\ x < z < t & \vee & t < z < y \end{cases}$$

NP-complete

Temporal template: domain is $\mathbb{Z}$, relations are definable by boolean combinations of $<$ and $y = x + c$

- Reasoning: given a collection Φ of facts and an atom ψ, does Φ imply ψ?
- Same as unsatisfiability of $\Phi \cup \{\neg\psi\}$ $\leftrightsquigarrow$ constraint satisfaction problem
- In spatial/temporal reasoning, variables are coordinates and the facts/atom are of the form $x < y, y = x + 5, \ldots$

Examples.

Ordering constraints

$$\begin{cases} x & \leq & y+1 \\ y & \leq & z-1 \\ z & \leq & t+3 \\ t & \leq & x-4 \end{cases}$$

In P: check if constraint graph has negative cycles

AND/OR scheduling

$$\begin{cases} x & < & \max(y,z) \\ y & < & t \\ t & < & \max(y,x) \end{cases}$$

In P: essentially arc consistency

Betweenness

$$\begin{cases} x < y < z & \vee & z < y < x \\ y < z < t & \vee & t < z < y \\ x < z < t & \vee & t < z < y \end{cases}$$

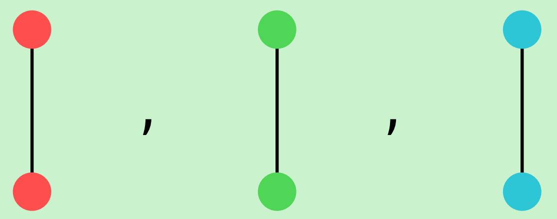
NP-complete

Temporal template: domain is $\mathbb{Z}$, relations are definable by boolean combinations of $<$ and $y = x + c$

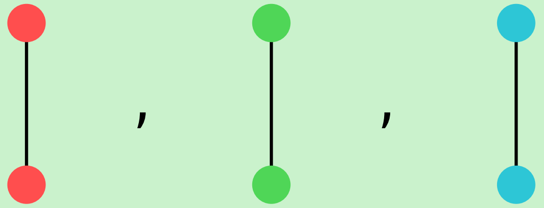
Theorem (Bodirsky-Martin-**M.**, *J. ACM*'18)**.** Let $\mathbb{A}$ be a temporal template.
Then CSP($\mathbb{A}$) is in P or NP-complete. Given $\mathbb{A}$, it is decidable which case applies.

Definition. Fix a set $\mathcal{F}$ of vertex-colored graphs. The vertex partitioning problem for $\mathcal{F}$ is:
Input: non-colored graph G,
Question: does there exist a vertex-coloring χ such that (G, χ) contains no element of $\mathcal{F}$.

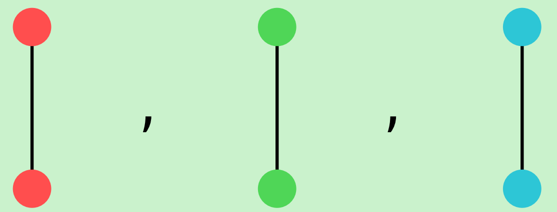
Definition. Fix a set $\mathcal{F}$ of vertex-colored graphs. The vertex partitioning problem for $\mathcal{F}$ is:
Input: non-colored graph G,
Question: does there exist a vertex-coloring χ such that (G, χ) contains no element of $\mathcal{F}$.

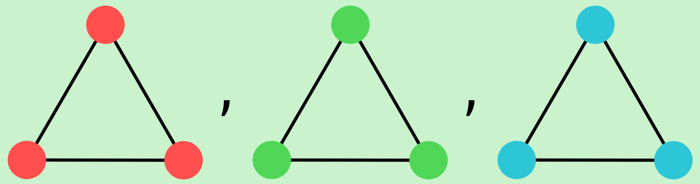
Examples. $\mathcal{F} = \Big\{$  $\Big\}$

Definition. Fix a set $\mathcal{F}$ of vertex-colored graphs. The vertex partitioning problem for $\mathcal{F}$ is:
Input: non-colored graph G,
Question: does there exist a vertex-coloring χ such that (G, χ) contains no element of $\mathcal{F}$.

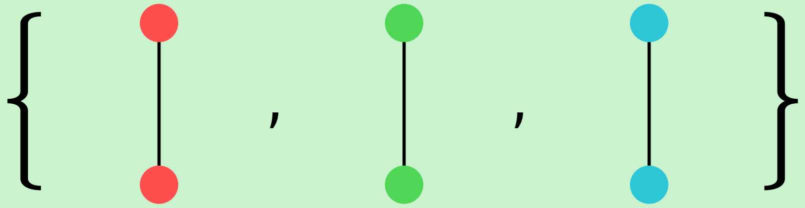
Examples. $\mathcal{F} = \Big\{$  $\Big\}$ $\rightsquigarrow$ 3-coloring problem

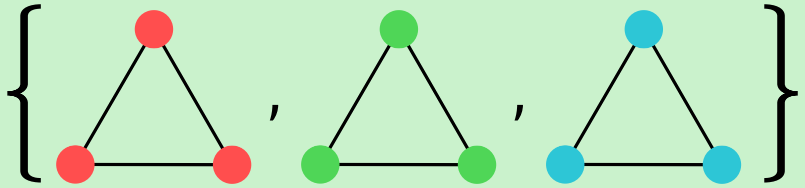
Definition. Fix a set $\mathcal{F}$ of vertex-colored graphs. The vertex partitioning problem for $\mathcal{F}$ is:
Input: non-colored graph G,
Question: does there exist a vertex-coloring χ such that (G, χ) contains no element of $\mathcal{F}$.

Examples. $\mathcal{F} = \{$  $\}$ $\rightsquigarrow$ 3-coloring problem

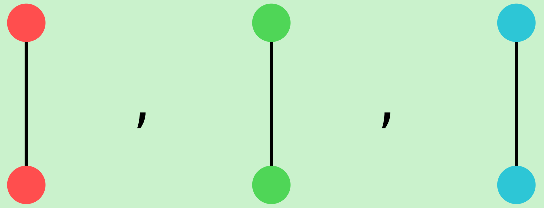
$\mathcal{F} = \{$  $\}$

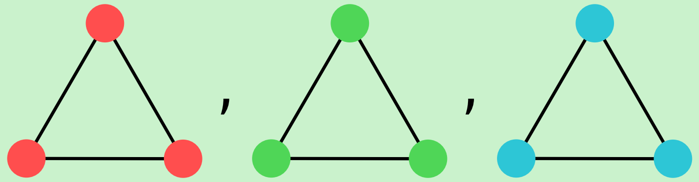
Definition. Fix a set $\mathcal{F}$ of vertex-colored graphs. The vertex partitioning problem for $\mathcal{F}$ is:
Input: non-colored graph G,
Question: does there exist a vertex-coloring χ such that (G, χ) contains no element of $\mathcal{F}$.

Examples. $\mathcal{F} =$  $\rightsquigarrow$ 3-coloring problem

$\mathcal{F} =$  $\rightsquigarrow$ no-monochromatic-triangle problem

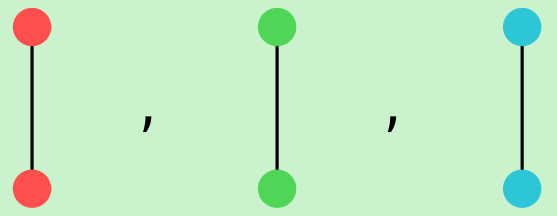
Definition. Fix a set $\mathcal{F}$ of vertex-colored graphs. The vertex partitioning problem for $\mathcal{F}$ is:
Input: non-colored graph G,
Question: does there exist a vertex-coloring χ such that (G, χ) contains no element of $\mathcal{F}$.

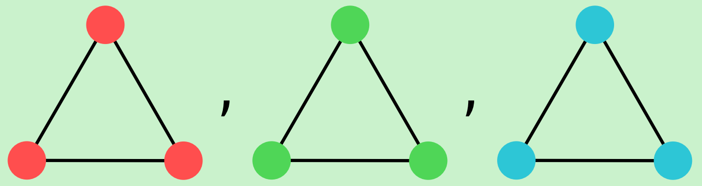
Examples. $\mathcal{F} = \{$  $\}$ $\rightsquigarrow$ 3-coloring problem

$\mathcal{F} = \{$  $\}$ $\rightsquigarrow$ no-monochromatic-triangle problem

Observation. There is $\mathbb{A}$ such that $\mathrm{CSP}(\mathbb{A})$ is the no-mono-triangle problem, but no finite $\mathbb{A}$.

Definition. Fix a set $\mathcal{F}$ of vertex-colored graphs. The vertex partitioning problem for $\mathcal{F}$ is:
Input: non-colored graph G,
Question: does there exist a vertex-coloring χ such that (G, χ) contains no element of $\mathcal{F}$.

Examples. $\mathcal{F} = \{$  $\}$ $\rightsquigarrow$ 3-coloring problem

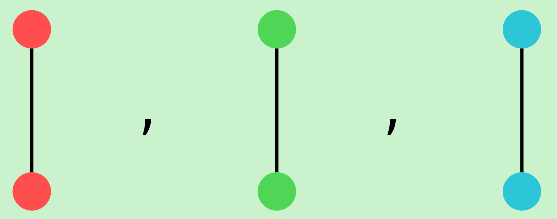
$\mathcal{F} = \{$  $\}$ $\rightsquigarrow$ no-monochromatic-triangle problem

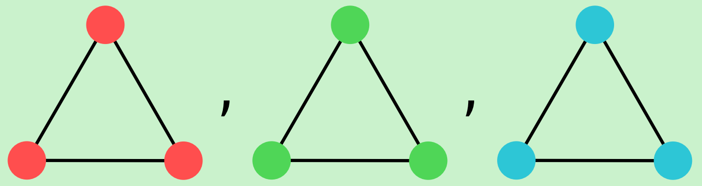
Observation. There is $\mathbb{A}$ such that CSP($\mathbb{A}$) is the no-mono-triangle problem, but no finite $\mathbb{A}$.

Theorem (Bodirsky-Madelaine-**M.**, *SIAM J. Comp.* '21)**.** Every vertex partitioning problem is in P or NP-complete. Given $\mathcal{F}$, it is decidable which case applies.

The proof uses techniques from infinite-domain constraint satisfaction.

Definition. Fix a set $\mathcal{F}$ of vertex-colored graphs. The vertex partitioning problem for $\mathcal{F}$ is:
Input: non-colored graph G,
Question: does there exist a vertex-coloring χ such that (G, χ) contains no element of $\mathcal{F}$.

Examples. $\mathcal{F} = \{$  $\}$ $\rightsquigarrow$ 3-coloring problem

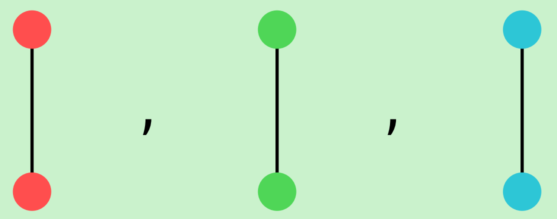
$\mathcal{F} = \{$  $\}$ $\rightsquigarrow$ no-monochromatic-triangle problem

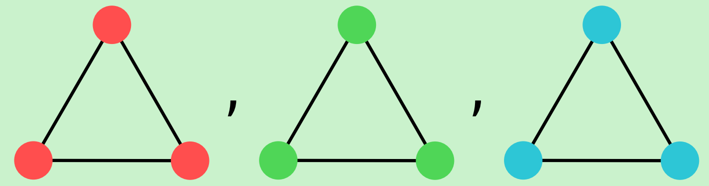
Observation. There is $\mathbb{A}$ such that CSP($\mathbb{A}$) is the no-mono-triangle problem, but no finite $\mathbb{A}$.

Theorem (M.-Nagy-Pinsker-Wrona, *SIAM J. Comp.* '24). Characterization of the $\mathcal{F}$ that have bounded treewidth duality.

The proof uses techniques from infinite-domain constraint satisfaction.

Definition. Fix a set $\mathcal{F}$ of vertex-colored graphs. The vertex partitioning problem for $\mathcal{F}$ is:
Input: non-colored graph G,
Question: does there exist a vertex-coloring χ such that (G, χ) contains no element of $\mathcal{F}$.

Examples. $\mathcal{F} = \{$  $\}$ $\rightsquigarrow$ 3-coloring problem

$\mathcal{F} = \{$  $\}$ $\rightsquigarrow$ no-monochromatic-triangle problem

Observation. There is $\mathbb{A}$ such that CSP($\mathbb{A}$) is the no-mono-triangle problem, but no finite $\mathbb{A}$.

Theorem (Bodirsky-Madelaine-**M.**, *SIAM J. Comp.* '21)**.** Every vertex partitioning problem is in P or NP-complete. Given $\mathcal{F}$, it is decidable which case applies.

The proof uses techniques from infinite-domain constraint satisfaction.

Project (with D. Perinti)**.** Understand the complexity of edge partitioning problems.

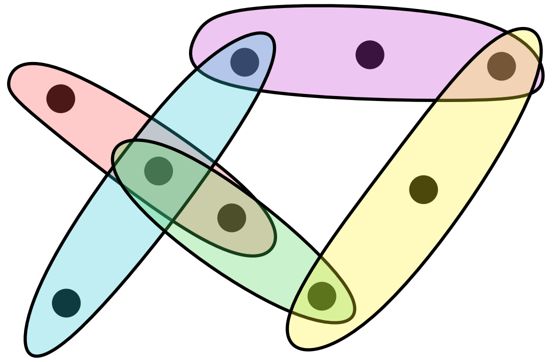
Algebraic approach works whenever $\mathrm{CSP}(\mathbb{A})$ has combinatorial certificates.

Example (1-in-3 Satisfiability). Given a set of constraints $(x, y, z), (y, z, t), \ldots,$ decide if there is a map $h\colon V \to \{0, 1\}$ such that for every constraint (a, b, c), exactly one of $h(a), h(b), h(c)$ is 1.

Algebraic approach works whenever CSP($\mathbb{A}$) has combinatorial certificates.

Example (1-in-3 Satisfiability). Given a set of constraints $(x, y, z), (y, z, t), \ldots,$ decide if there is a map $h \colon V \to \{0, 1\}$ such that for every constraint (a, b, c), exactly one of $h(a), h(b), h(c)$ is 1.

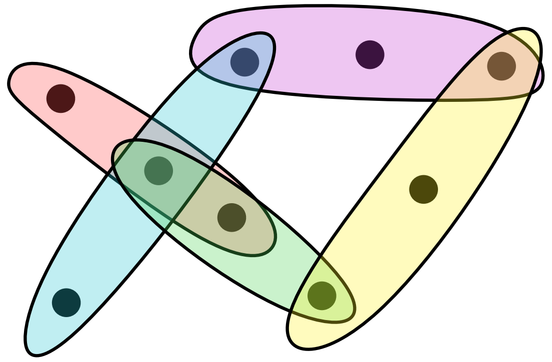
Instance



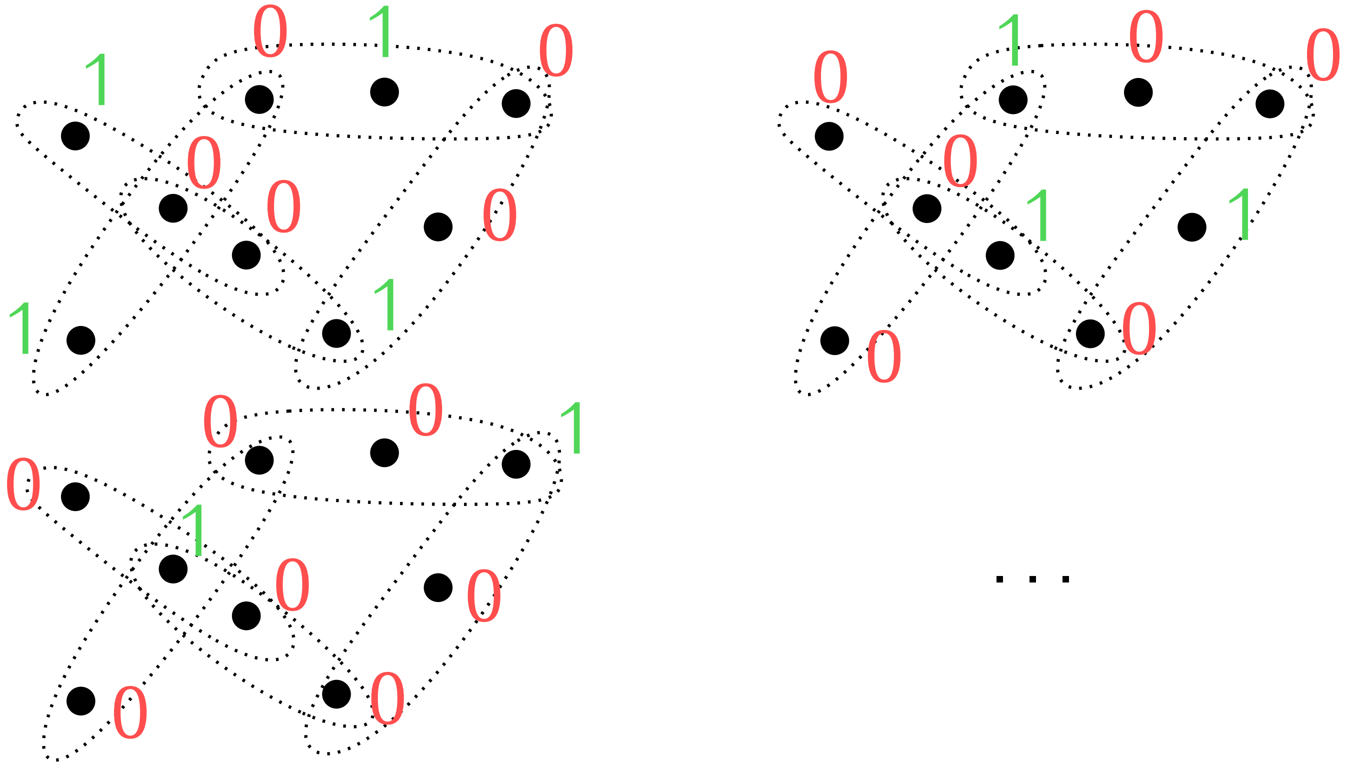
Algebraic approach works whenever CSP($\mathbb{A}$) has combinatorial certificates.

Example (1-in-3 Satisfiability). Given a set of constraints $(x, y, z), (y, z, t), \ldots$, decide if there is a map $h\colon V \to \{0, 1\}$ such that for every constraint (a, b, c), exactly one of $h(a), h(b), h(c)$ is 1.

Instance



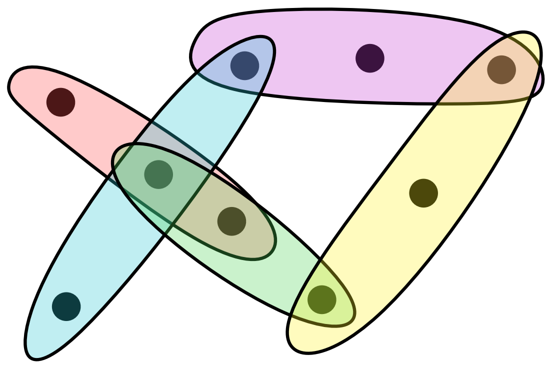
Possible certificates



Algebraic approach works whenever CSP($\mathbb{A}$) has combinatorial certificates.

Example (Betweenness). Given a set of constraints $(x, y, z), (y, z, t), \ldots,$ decide if there is a linear order on the variables such that for every constraint (a, b, c), either $a < b < c$ or $c < b < a$.

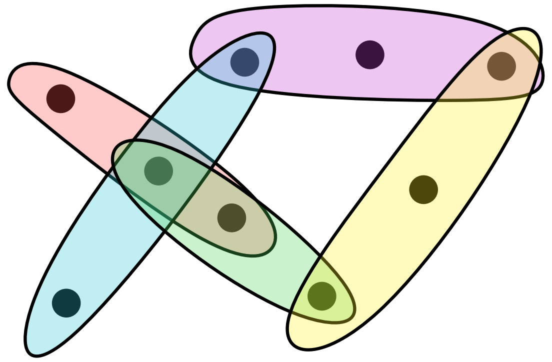
Instance



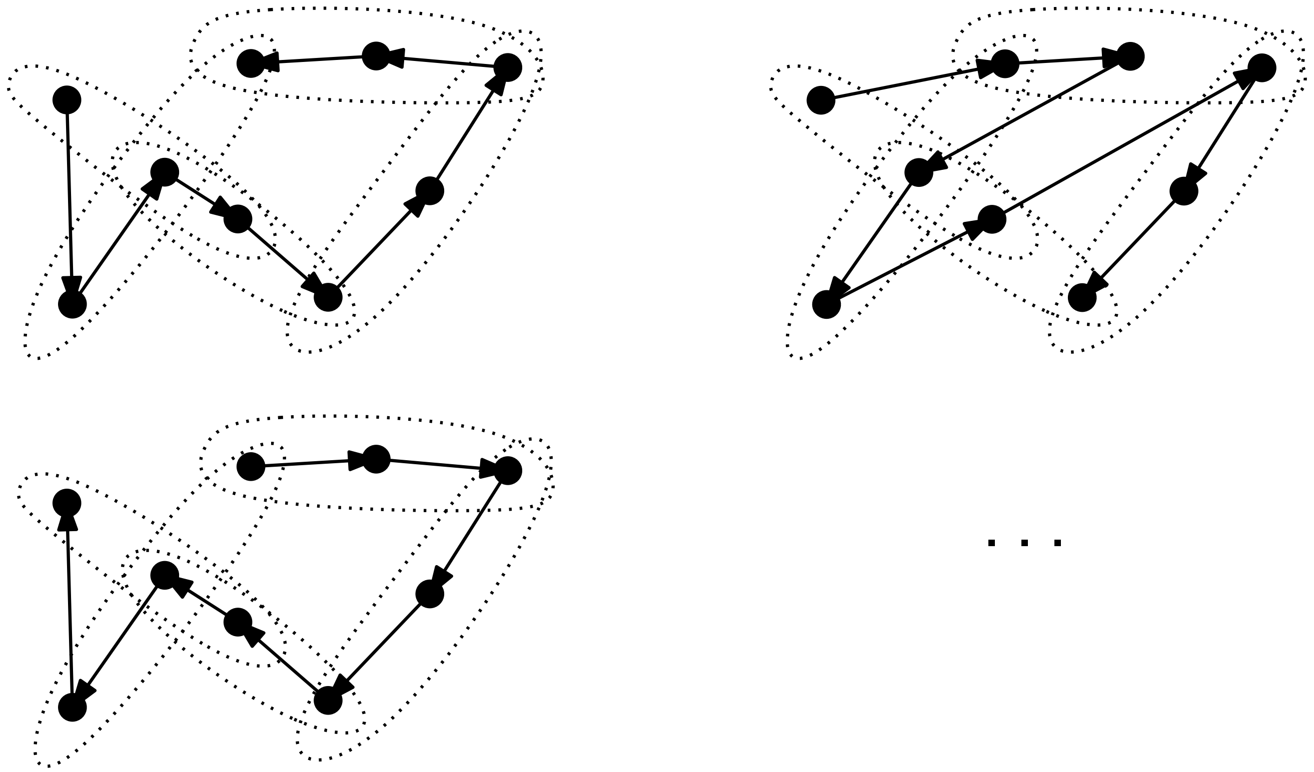
Algebraic approach works whenever CSP($\mathbb{A}$) has combinatorial certificates.

Example (Betweenness)**.** Given a set of constraints $(x, y, z), (y, z, t), \ldots$, decide if there is a linear order on the variables such that for every constraint (a, b, c), either $a < b < c$ or $c < b < a$.

Instance



Possible certificates



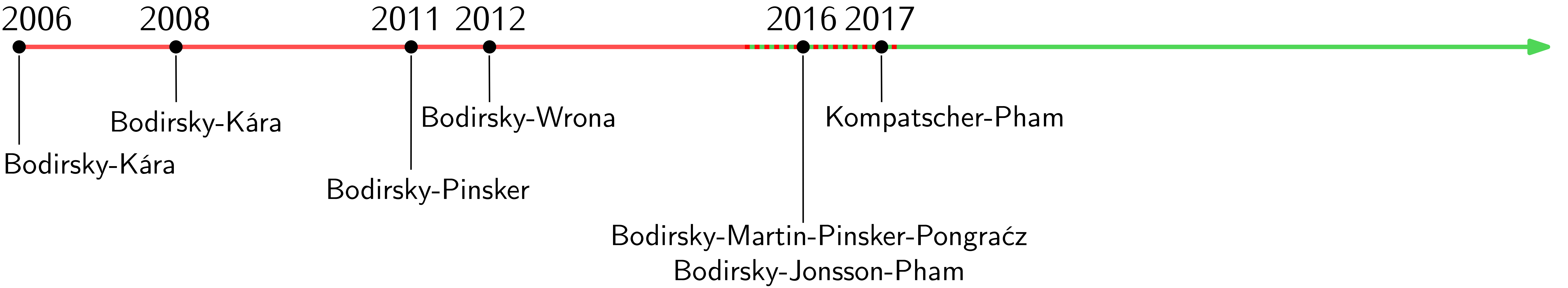
$\ldots$

Algebraic approach works whenever $\mathrm{CSP}(\mathbb{A})$ has combinatorial certificates.

Conjecture (2011). Let $\mathbb{A}$ be s.t. $\mathrm{CSP}(\mathbb{A})$ has combinatorial certificates and $\mathrm{Pol}(\mathbb{A}) \not\to \mathrm{Pol}(\mathrm{S}\small{\mathrm{AT}})$. Then $\mathrm{CSP}(\mathbb{A})$ is solvable in polynomial time.

Algebraic approach works whenever CSP($\mathbb{A}$) has combinatorial certificates.

Conjecture (2011). Let $\mathbb{A}$ be s.t. CSP($\mathbb{A}$) has combinatorial certificates and $\mathrm{Pol}(\mathbb{A}) \not\to \mathrm{Pol}(\textsc{Sat})$. Then CSP($\mathbb{A}$) is solvable in polynomial time.

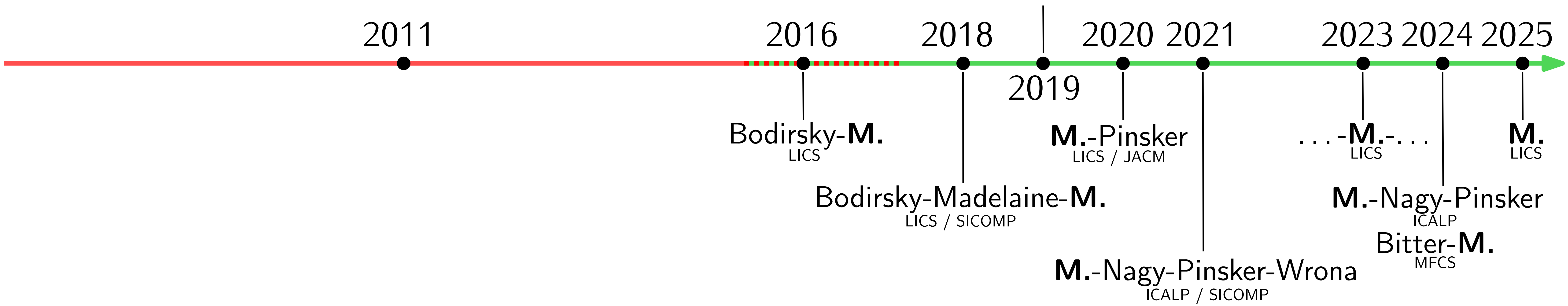


Early proofs:

- Proofs by case distinctions
- Victim of combinatorial explosion
- *Ad hoc* arguments
- Limited theoretical understanding

Algebraic approach works whenever CSP($\mathbb{A}$) has combinatorial certificates.

Conjecture (2011). Let $\mathbb{A}$ be s.t. CSP($\mathbb{A}$) has combinatorial certificates and Pol($\mathbb{A}$) $\not\to$ Pol(Sat). Then CSP($\mathbb{A}$) is solvable in polynomial time.



Early proofs:

- Proofs by case distinctions
- Victim of combinatorial explosion
- *Ad hoc* arguments
- Limited theoretical understanding

Modern proofs:

- Unifying algebraic theory
- No combinatorial explosion
- Generic reduction to finite-domain CSPs
- Generic uniform algorithms
- Finer understanding than P vs. NP

Theorem (Bodirsky-**M.** LICS'16)**.** For every $\mathbb{A}$ with combinatorial certificates, there exists an equivalence relation $\equiv$ such that if $\mathrm{Pol}(\mathbb{A}, \equiv) \not\to \mathrm{Pol}(\textsc{Sat})$, then $\mathrm{CSP}(\mathbb{A})$ is solvable in polynomial time.

Theorem (Bodirsky-**M.** LICS'16)**.** For every $\mathbb{A}$ with combinatorial certificates, there exists an equivalence relation $\equiv$ such that if $\mathrm{Pol}(\mathbb{A}, \equiv) \not\to \mathrm{Pol}(\textsc{Sat})$, then $\mathrm{CSP}(\mathbb{A})$ is solvable in polynomial time.

- $\equiv$ only depends on the type of combinatorial certificates
- Uses a reduction to a finite-domain CSP combined with the Bulatov-Zhuk theorem
- $\mathrm{Pol}(\mathbb{A}, \equiv) \subseteq \mathrm{Pol}(\mathbb{A})$

Theorem (Bodirsky-**M.** LICS'16). For every $\mathbb{A}$ with combinatorial certificates, there exists an equivalence relation $\equiv$ such that if $\mathrm{Pol}(\mathbb{A}, \equiv) \not\to \mathrm{Pol}(\textsc{Sat})$, then $\mathrm{CSP}(\mathbb{A})$ is solvable in polynomial time.

- $\equiv$ only depends on the type of combinatorial certificates
- Uses a reduction to a finite-domain CSP combined with the Bulatov-Zhuk theorem
- $\mathrm{Pol}(\mathbb{A}, \equiv) \subseteq \mathrm{Pol}(\mathbb{A})$

"Only" remaining case to understand:

$$\begin{tikzcd}
\mathrm{Pol}(\mathbb{A}) \arrow[drr, bend left, "/" marking] & & \\
\cup\!\mid & & \\
\mathrm{Pol}(\mathbb{A}, \equiv) \arrow[rr] & & \mathrm{Pol}(\textsc{Sat})
\end{tikzcd}$$

Theorem (Bodirsky-**M.** LICS'16). For every $\mathbb{A}$ with combinatorial certificates, there exists an equivalence relation $\equiv$ such that if $\mathrm{Pol}(\mathbb{A}, \equiv) \not\to \mathrm{Pol}(\textsc{Sat})$, then $\mathrm{CSP}(\mathbb{A})$ is solvable in polynomial time.

- $\equiv$ only depends on the type of combinatorial certificates
- Uses a reduction to a finite-domain CSP combined with the Bulatov-Zhuk theorem
- $\mathrm{Pol}(\mathbb{A}, \equiv) \subseteq \mathrm{Pol}(\mathbb{A})$

"Only" remaining case to understand:

$$\begin{tikzcd}
\mathrm{Pol}(\mathbb{A}) \arrow[d, phantom, "\cup\mid"] \arrow[dr, bend left, "/" marking] & \\
\mathrm{Pol}(\mathbb{A}, \equiv) \arrow[r] & \mathrm{Pol}(\textsc{Sat})
\end{tikzcd}$$

2020–2025: algebraic theory to handle this situation in generality

Theorem (Bodirsky-**M.** LICS'16). For every $\mathbb{A}$ with combinatorial certificates, there exists an equivalence relation $\equiv$ such that if $\mathrm{Pol}(\mathbb{A}, \equiv) \not\to \mathrm{Pol}(\textsc{Sat})$, then $\mathrm{CSP}(\mathbb{A})$ is solvable in polynomial time.

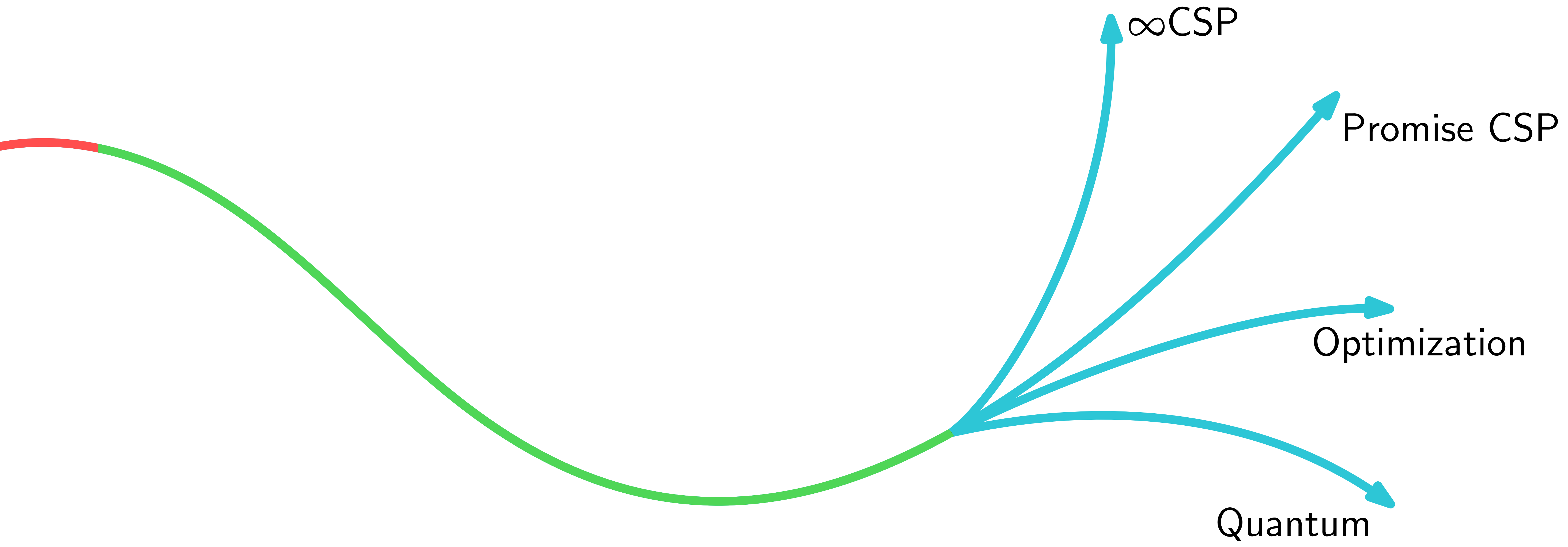
- $\equiv$ only depends on the type of combinatorial certificates
- Uses a reduction to a finite-domain CSP combined with the Bulatov-Zhuk theorem
- $\mathrm{Pol}(\mathbb{A}, \equiv) \subseteq \mathrm{Pol}(\mathbb{A})$

"Only" remaining case to understand:

$$\begin{tikzcd}
\mathrm{Pol}(\mathbb{A}) \arrow[drr, bend left, "/" marking] \arrow[d, phantom, sloped, "\supseteq"] & & \\
\mathrm{Pol}(\mathbb{A}, \equiv) \arrow[rr] & & \mathrm{Pol}(\textsc{Sat})
\end{tikzcd}$$

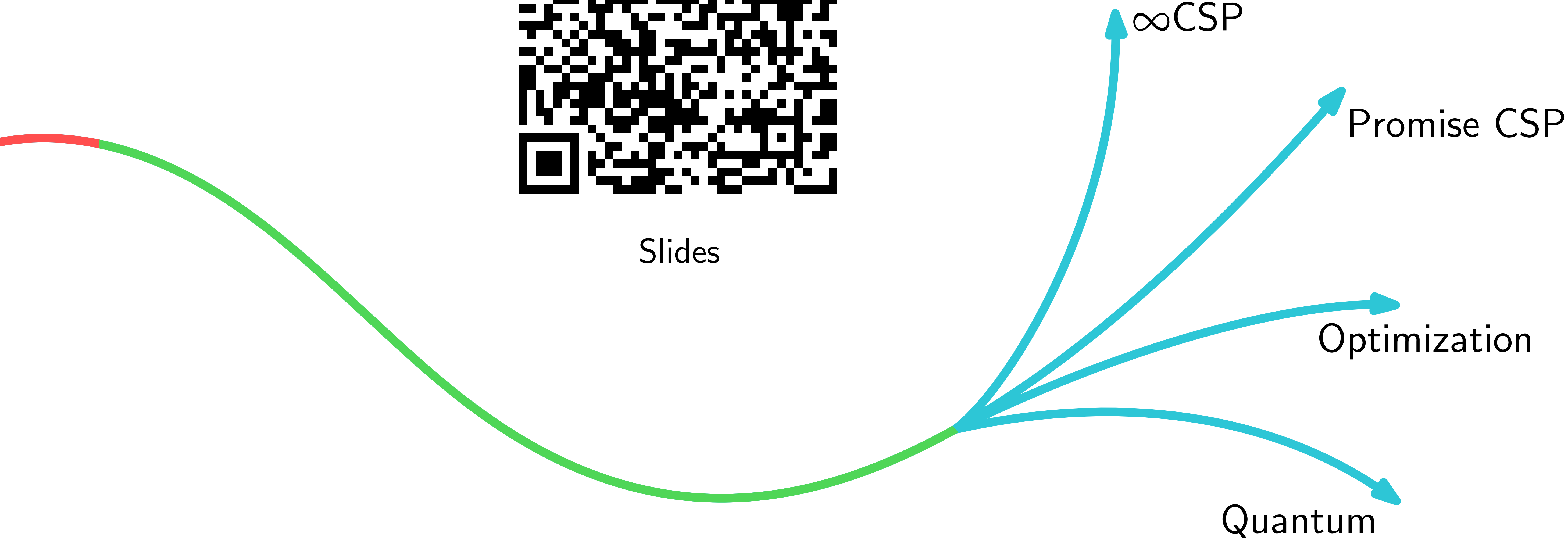
2020–2025: algebraic theory to handle this situation in generality

Invited tutorial from 2025 Dagstuhl seminar:

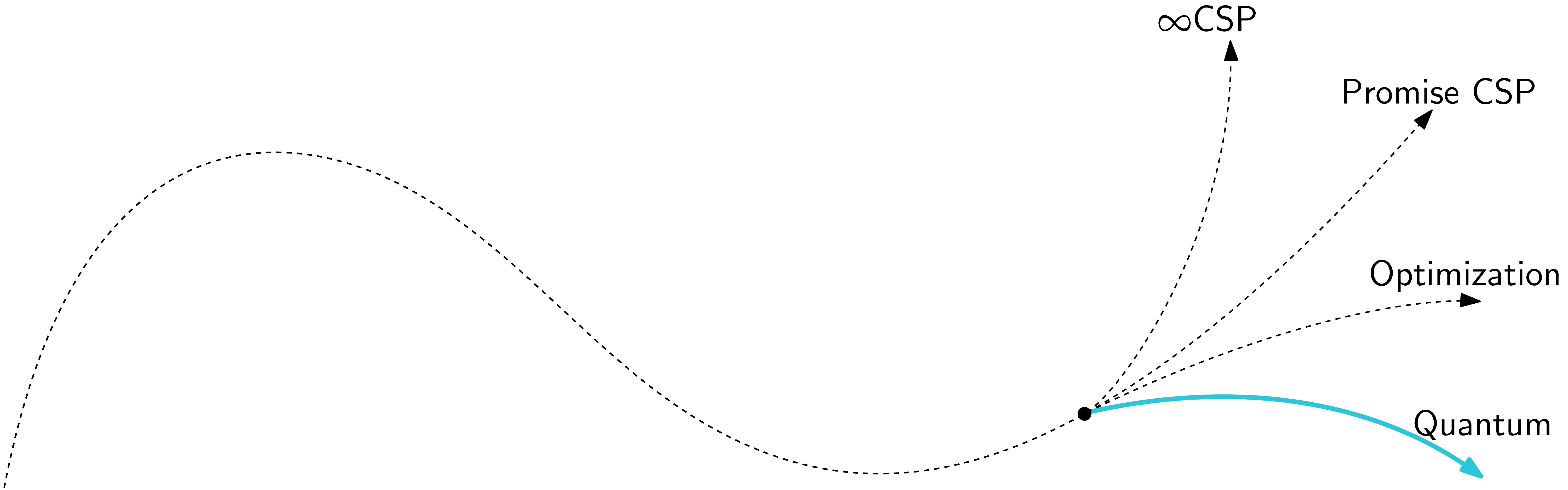

∞CSP
Promise CSP
Optimization
Quantum

Thank you for your attention!

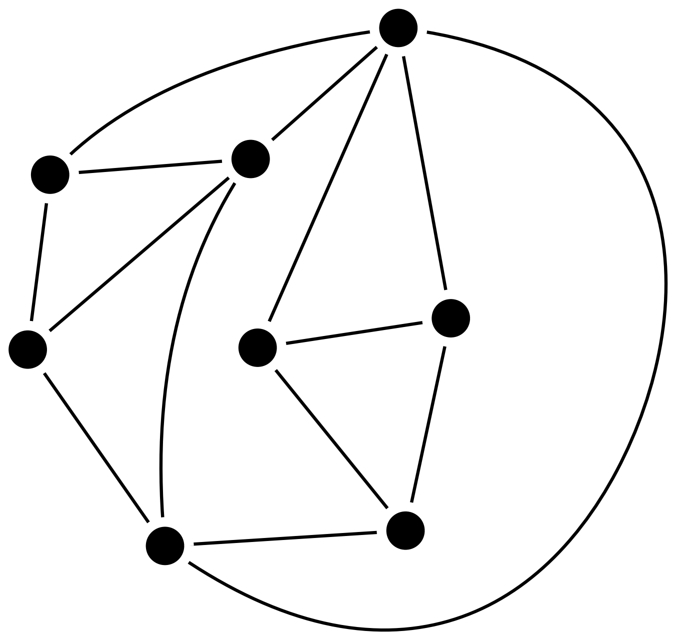
Slides



Research in Quantum Constraint Satisfaction



Verifier

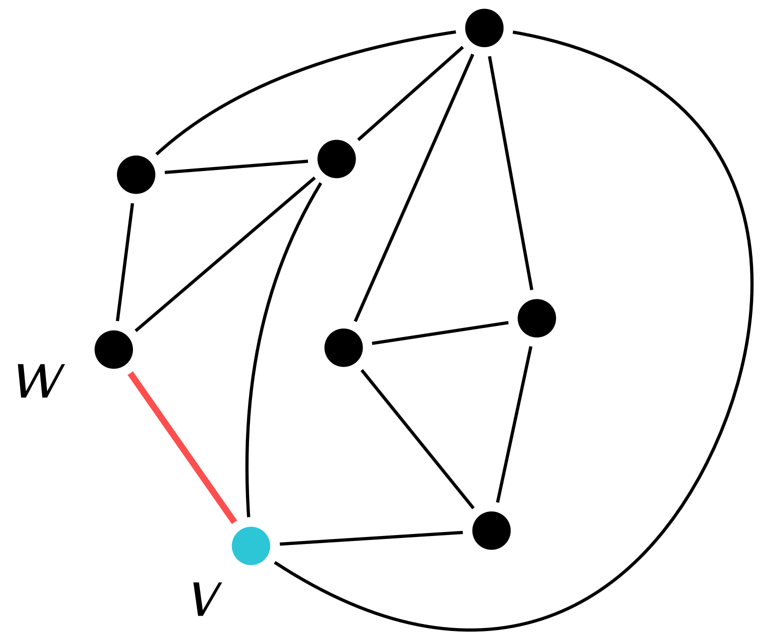


Alice

Bob

Verifier

1. selects $e = (v, w)$ following distribution π

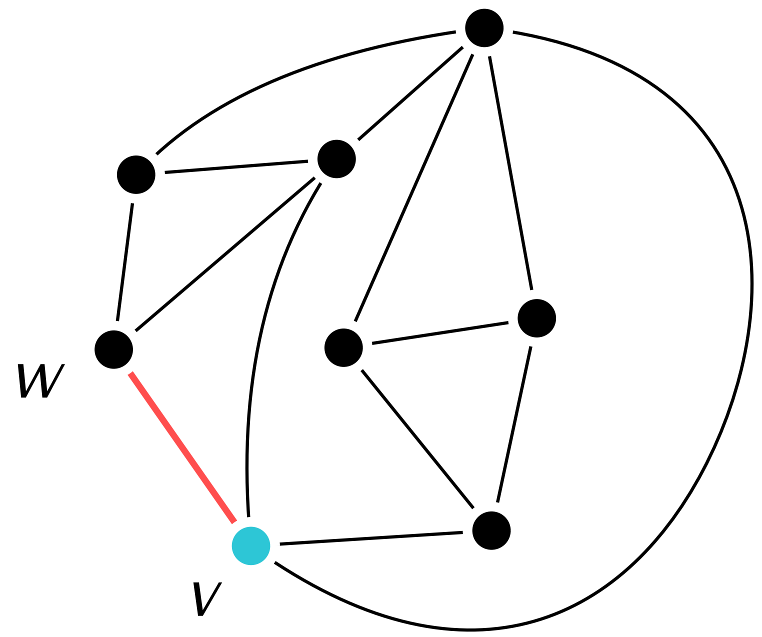


Alice

Bob

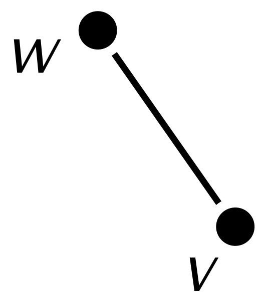
Verifier

1. selects $e = (v, w)$ following distribution π
2. sends e to Alice
3. sends v to Bob

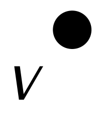


Alice

4. Answers $(a, b) \in [k]^2$ with $a \neq b$ following distribution p_e



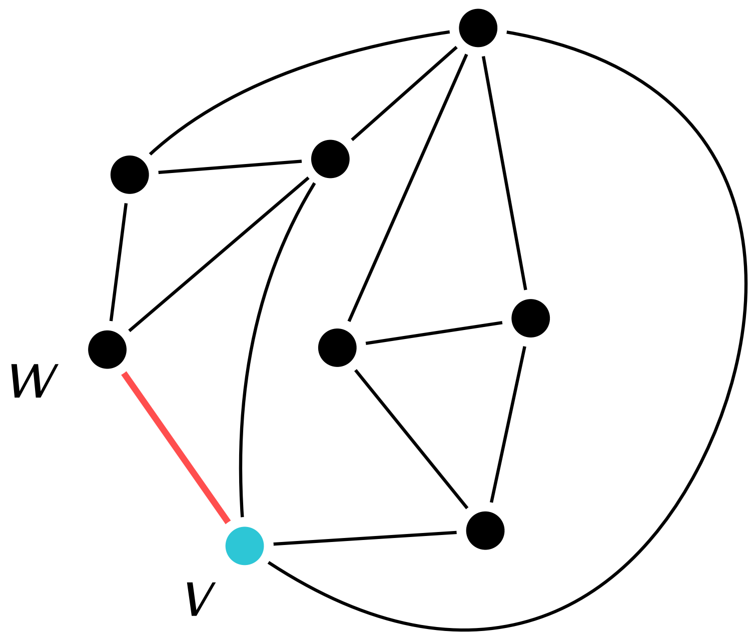
Bob



4. Answers $c \in [k]$ following distribution p_v

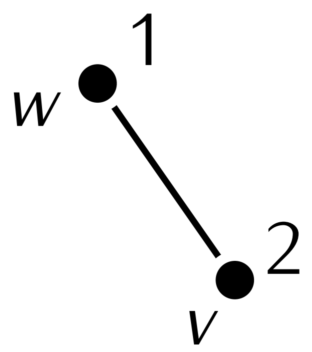
Verifier

1. selects $e = (v, w)$ following distribution π
2. sends e to Alice
3. sends v to Bob

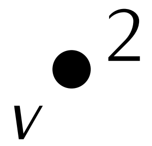


Alice

4. Answers $(a, b) \in [k]^2$ with $a \neq b$ following distribution p_e



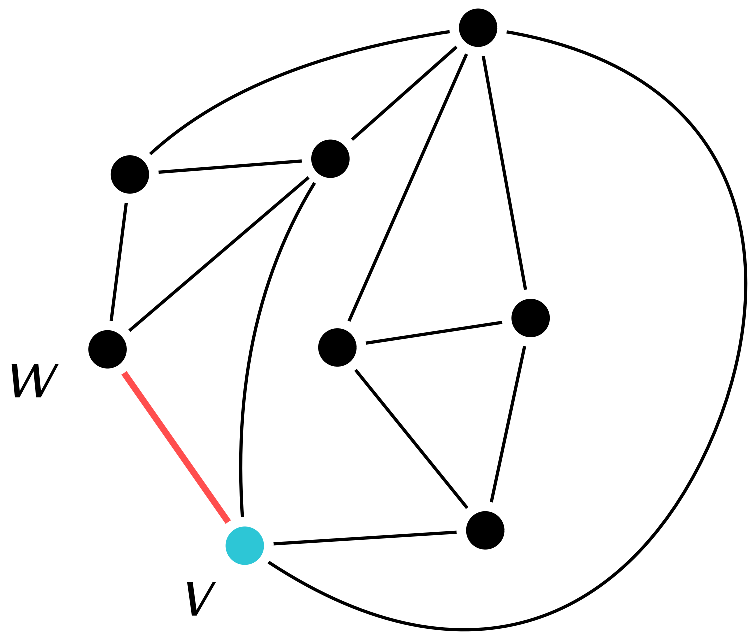
Bob



4. Answers $c \in [k]$ following distribution p_v

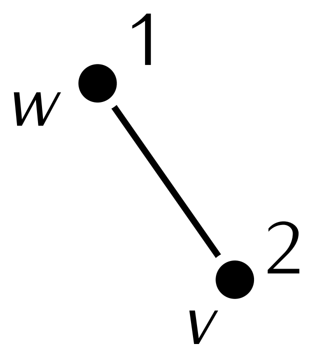
Verifier

1. selects $e = (v, w)$ following distribution π
2. sends e to Alice
3. sends v to Bob

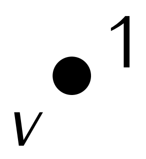


Alice

4. Answers $(a, b) \in [k]^2$ with $a \neq b$ following distribution p_e



Bob



4. Answers $c \in [k]$ following distribution p_v

- Strategy for Alice: for each $e \in E$, a probability distribution p_e on $\{(a, b) \in [k]^2 \mid a \neq b\}$
- Strategy for Bob: for each $v \in V$, a probability distribution p_v on $[k]$
- Winning probability of a strategy: $\Pr_{e \sim \pi, (a,b) \sim p_e, c \sim p_v}[a = c]$

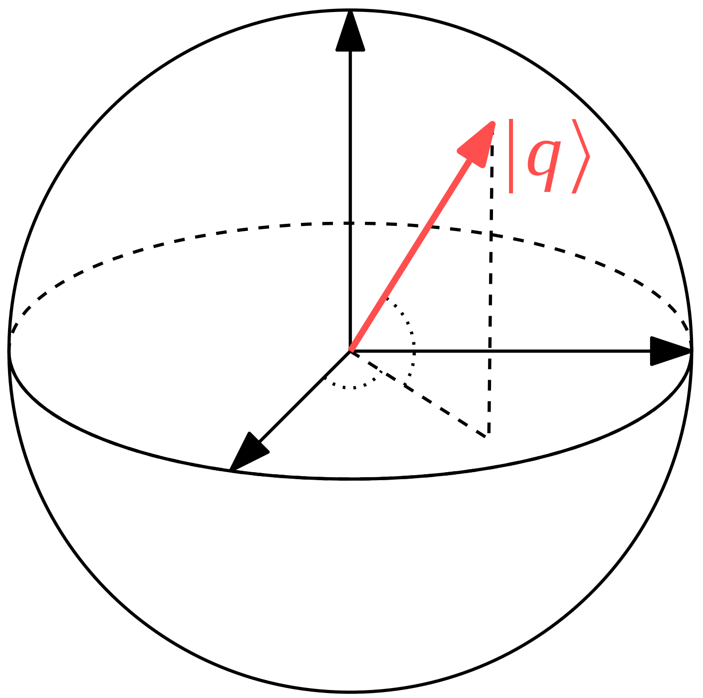
Observation. Alice and Bob win with probability 1 if, and only if, G is k-colorable.

- Strategy for Alice: for each $e \in E$, a probability distribution p_e on $\{(a, b) \in [k]^2 \mid a \neq b\}$
- Strategy for Bob: for each $v \in V$, a probability distribution p_v on $[k]$
- Winning probability of a strategy: $\Pr_{e \sim \pi, (a,b) \sim p_e, c \sim p_v}[a = c]$

Observation. Alice and Bob win with probability 1 if, and only if, G is k-colorable.
What if Alice and Bob share an entangled state $|q\rangle$?

- Strategy for Alice: for each $e \in E$, a probability distribution p_e on $\{(a, b) \in [k]^2 \mid a \neq b\}$
- Strategy for Bob: for each $v \in V$, a probability distribution p_v on $[k]$
- Winning probability of a strategy: $\Pr_{e \sim \pi, (a,b) \sim p_e, c \sim p_v}[a = c]$

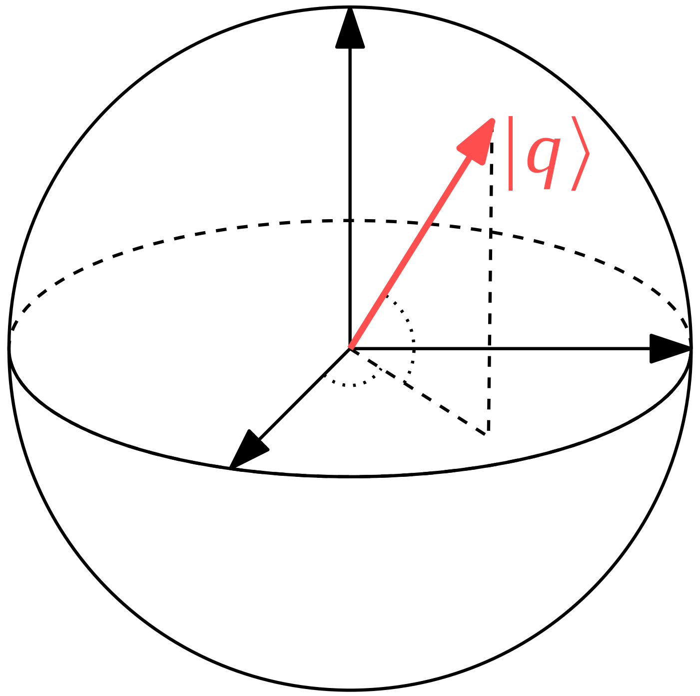
Observation. Alice and Bob win with probability 1 if, and only if, G is k-colorable.
What if Alice and Bob share an entangled state $|q\rangle$?



- Strategy for Alice: for each $e \in E$, a probability distribution p_e on $\{(a,b) \in [k]^2 \mid a \neq b\}$
- Strategy for Bob: for each $v \in V$, a probability distribution p_v on $[k]$
- Winning probability of a strategy: $\Pr_{e \sim \pi, (a,b) \sim p_e, c \sim p_v}[a = c]$

Observation. Alice and Bob win with probability 1 if, and only if, G is k-colorable.

What if Alice and Bob share an entangled state $|q\rangle$?



How Alice performs a measurement:

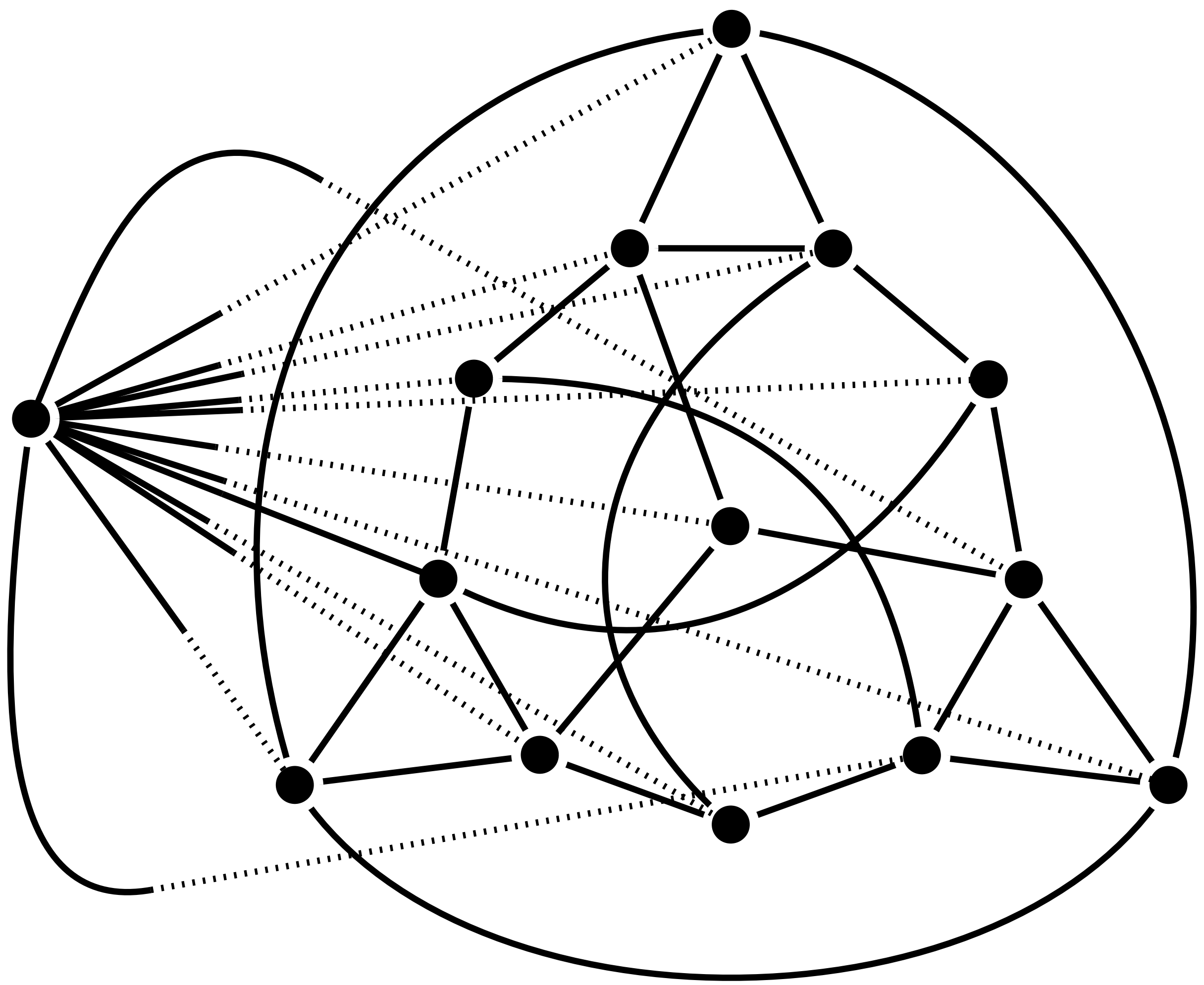
- Choose projectors $Q_{a,b}$ summing to id
- Output of measurement is (a,b) with probability $\langle q|Q_{a,b}|q\rangle$

- Strategy for Alice: for each $e \in E$, a measurement Q_e on $\{(a,b) \in [k]^2 \mid a \neq b\}$
- Strategy for Bob: for each $v \in V$, a measurement Q_v on $[k]$
- Winning probability of a strategy: $\sum_e \sum_{(a,b)} \pi(e)\langle q|Q_{e,ab} \otimes Q_{v,a}|q\rangle$

Definition. Say G is quantum k-colorable if there is a quantum strategy that wins with probability 1.

- Strategy for Alice: for each $e \in E$, a measurement Q_e on $\{(a,b) \in [k]^2 \mid a \neq b\}$
- Strategy for Bob: for each $v \in V$, a measurement Q_v on $[k]$
- Winning probability of a strategy: $\sum_e \sum_{(a,b)} \pi(e)\langle q|Q_{e,ab} \otimes Q_{v,a}|q\rangle$

Definition. Say G is quantum k-colorable if there is a quantum strategy that wins with probability 1.



Quantum 4-colorable graph with chromatic number 5
[Mančinska-Roberson '18]

- Strategy for Alice: for each $e \in E$, a measurement Q_e on $\{(a,b) \in [k]^2 \mid a \neq b\}$
- Strategy for Bob: for each $v \in V$, a measurement Q_v on $[k]$
- Winning probability of a strategy: $\sum_e \sum_{(a,b)} \pi(e) \langle q | Q_{e,ab} \otimes Q_{v,a} | q \rangle$

Definition. Say G is quantum k-colorable if there is a quantum strategy that wins with probability 1.

Theorem (Culf-Mastel, FOCS'25)**.** There exists $s \in (0,1)$ such that it is undecidable to distinguish graphs that are quantum 3-colorable from those where every strategy wins with probability $< s$.

- Strategy for Alice: for each $e \in E$, a measurement Q_e on $\{(a,b) \in [k]^2 \mid a \neq b\}$
- Strategy for Bob: for each $v \in V$, a measurement Q_v on $[k]$
- Winning probability of a strategy: $\sum_e \sum_{(a,b)} \pi(e)\langle q | Q_{e,ab} \otimes Q_{v,a} | q \rangle$

Definition. Say G is quantum k-colorable if there is a quantum strategy that wins with probability 1.

Theorem (Culf-Mastel, FOCS'25)**.** There exists $s \in (0,1)$ such that it is undecidable to distinguish graphs that are quantum 3-colorable from those where every strategy wins with probability $< s$.

- Similar flavor to the PCP theorem in classical complexity
- Hardness of approximation result in the quantum setting
- Unlike in classical complexity, many reductions still unknown
- Relies on commutativity gadgets for 3-coloring (Ji, 2013)

- Strategy for Alice: for each $e \in E$, a measurement Q_e on $\{(a,b) \in [k]^2 \mid a \neq b\}$
- Strategy for Bob: for each $v \in V$, a measurement Q_v on $[k]$
- Winning probability of a strategy: $\sum_e \sum_{(a,b)} \pi(e)\langle q|Q_{e,ab} \otimes Q_{v,a}|q\rangle$

Definition. Say G is quantum k-colorable if there is a quantum strategy that wins with probability 1.

Theorem (Culf-Mastel, FOCS'25)**.** There exists $s \in (0,1)$ such that it is undecidable to distinguish graphs that are quantum 3-colorable from those where every strategy wins with probability $< s$.

- Similar flavor to the PCP theorem in classical complexity
- Hardness of approximation result in the quantum setting
- Unlike in classical complexity, many reductions still unknown
- Relies on commutativity gadgets for 3-coloring (Ji, 2013)

Theorem (Ciardo-**M.**, 2025+)**.** k-coloring has commutativity gadgets for all $k \geq 3$.
Algebraic characterization of the existence of commutativity gadgets using polymorphisms.

- Strategy for Alice: for each $e \in E$, a measurement Q_e on $\{(a,b) \in [k]^2 \mid a \neq b\}$
- Strategy for Bob: for each $v \in V$, a measurement Q_v on $[k]$
- Winning probability of a strategy: $\sum_e \sum_{(a,b)} \pi(e)\langle q|Q_{e,ab} \otimes Q_{v,a}|q\rangle$

Definition. Say G is quantum k-colorable if there is a quantum strategy that wins with probability 1.

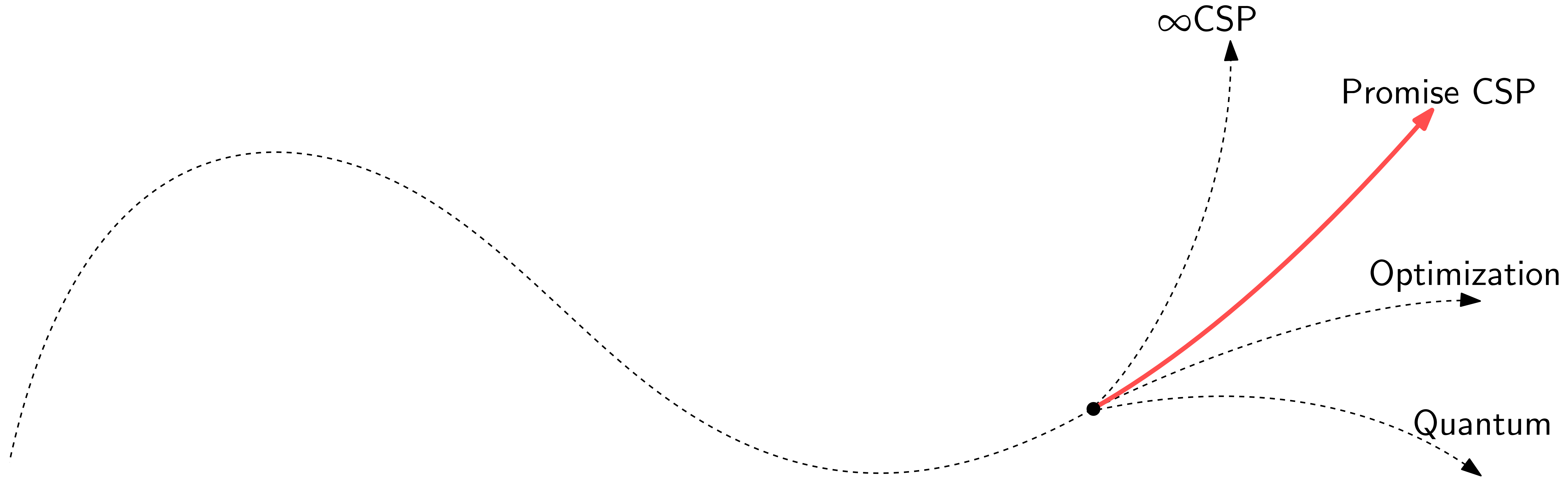
Theorem (Culf-Mastel, FOCS'25)**.** There exists $s \in (0,1)$ such that it is undecidable to distinguish graphs that are quantum 3-colorable from those where every strategy wins with probability $< s$.

- Similar flavor to the PCP theorem in classical complexity
- Hardness of approximation result in the quantum setting
- Unlike in classical complexity, many reductions still unknown
- Relies on commutativity gadgets for 3-coloring (Ji, 2013)

Theorem (Ciardo-**M.**, 2025+)**.** k-coloring has commutativity gadgets for all $k \geq 3$.
Algebraic characterization of the existence of commutativity gadgets using polymorphisms.

Project (with L. Ciardo and G. Joubert)**.** Understand the complexity landscape for entangled CSPs.

Research in Promise Constraint Satisfaction



Solving CSPs under structural assumptions on the instances:

Definition. $\mathrm{PCSP}(\mathbb{A}, \mathbb{B})$ is the problem to find a satisfying assignment in $\mathbb{B}$, under the promise that there exists one in $\mathbb{A}$.

Solving CSPs under structural assumptions on the instances:

Definition. $\mathrm{PCSP}(\mathbb{A}, \mathbb{B})$ is the problem to find a satisfying assignment in $\mathbb{B}$, under the promise that there exists one in $\mathbb{A}$.

Example. $\mathrm{PCSP}(\mathbb{K}_3, \mathbb{K}_6)$
Given an undirected 3-colorable graph G, find a 6-coloring.
Unknown complexity, related to the existence of a 2-approximation algorithm for the chromatic number.

Solving CSPs under structural assumptions on the instances:

Definition. $\mathrm{PCSP}(\mathbb{A}, \mathbb{B})$ is the problem to find a satisfying assignment in $\mathbb{B}$, under the promise that there exists one in $\mathbb{A}$.

Example. $\mathrm{PCSP}(\mathbb{K}_3, \mathbb{K}_6)$
Given an undirected 3-colorable graph G, find a 6-coloring.
Unknown complexity, related to the existence of a 2-approximation algorithm for the chromatic number.

- Massive framework for the study of approximation problems
- Existence of a suitable generalization to the algebraic approach

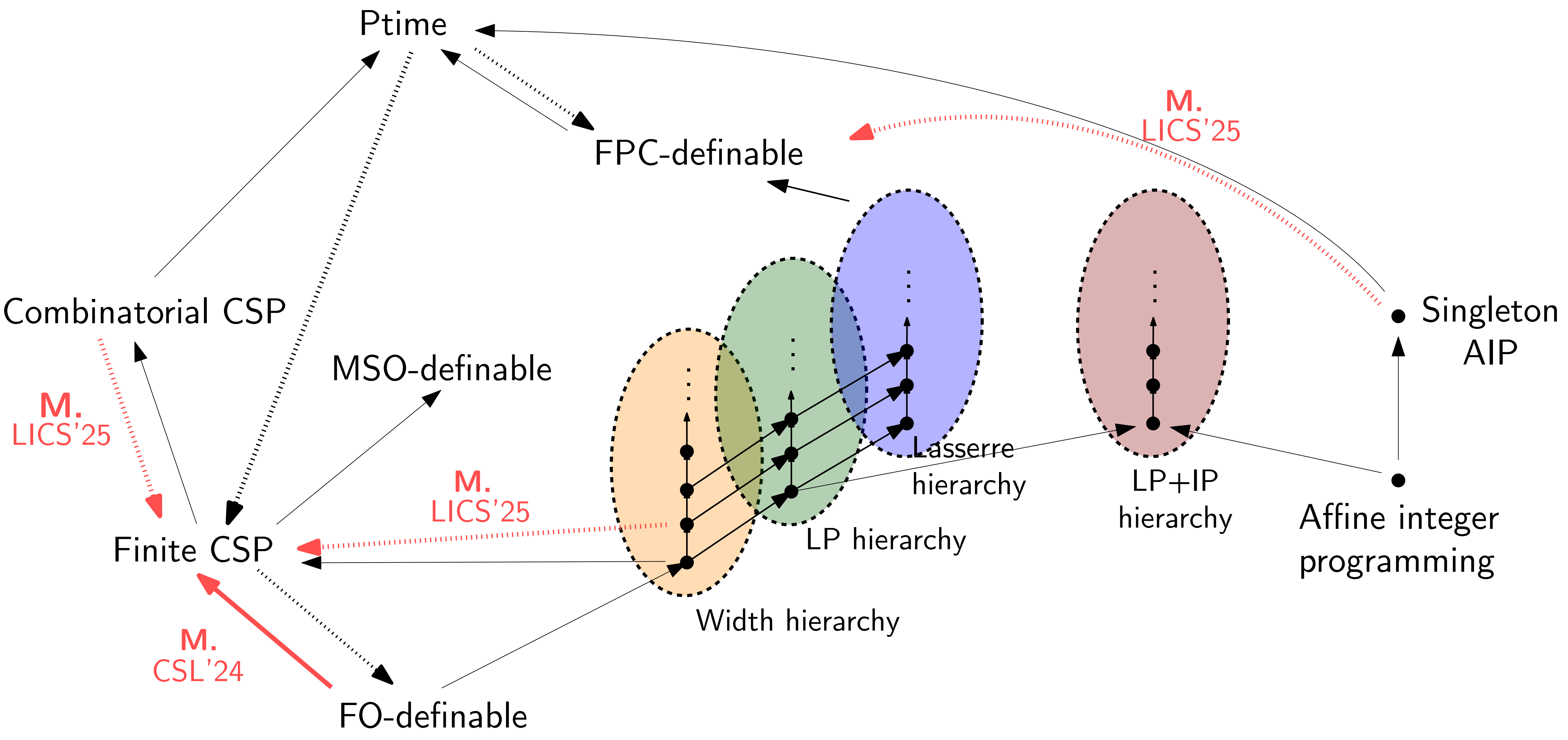
Solving CSPs under structural assumptions on the instances:

Definition. $\mathrm{PCSP}(\mathbb{A}, \mathbb{B})$ is the problem to find a satisfying assignment in $\mathbb{B}$, under the promise that there exists one in $\mathbb{A}$.

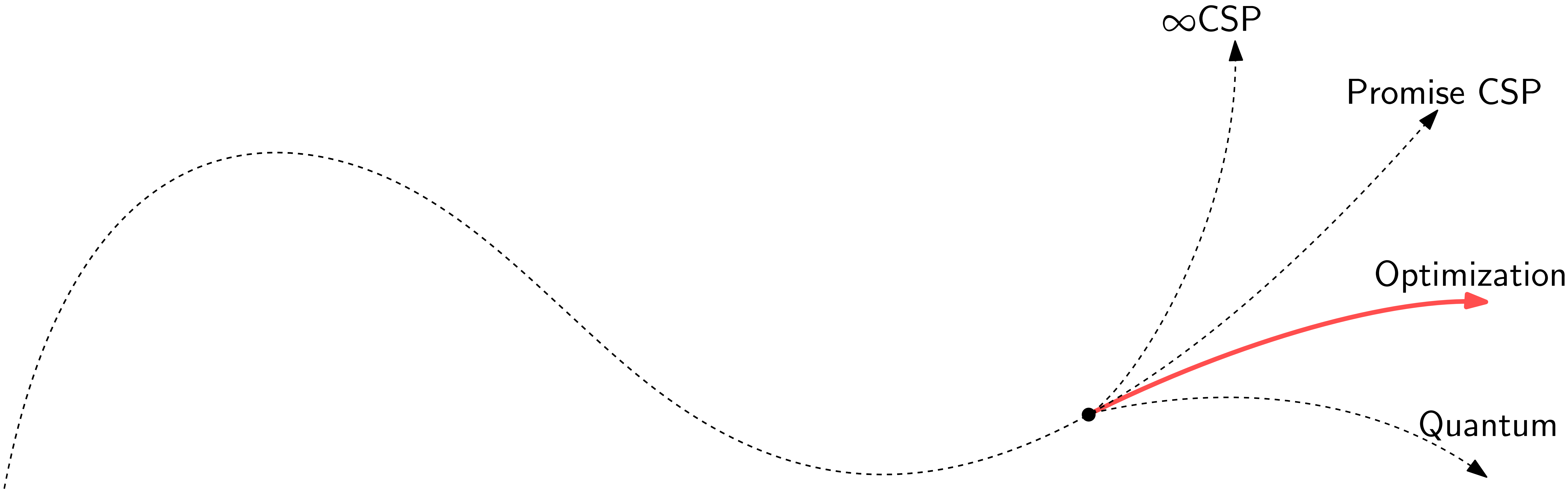
Example. $\mathrm{PCSP}(\mathbb{K}_3, \mathbb{K}_6)$
Given an undirected 3-colorable graph G, find a 6-coloring.
Unknown complexity, related to the existence of a 2-approximation algorithm for the chromatic number.

- Massive framework for the study of approximation problems
- Existence of a suitable generalization to the algebraic approach
- Surprising connection betweeen finite-domain PCSPs and combinatorial CSPs:

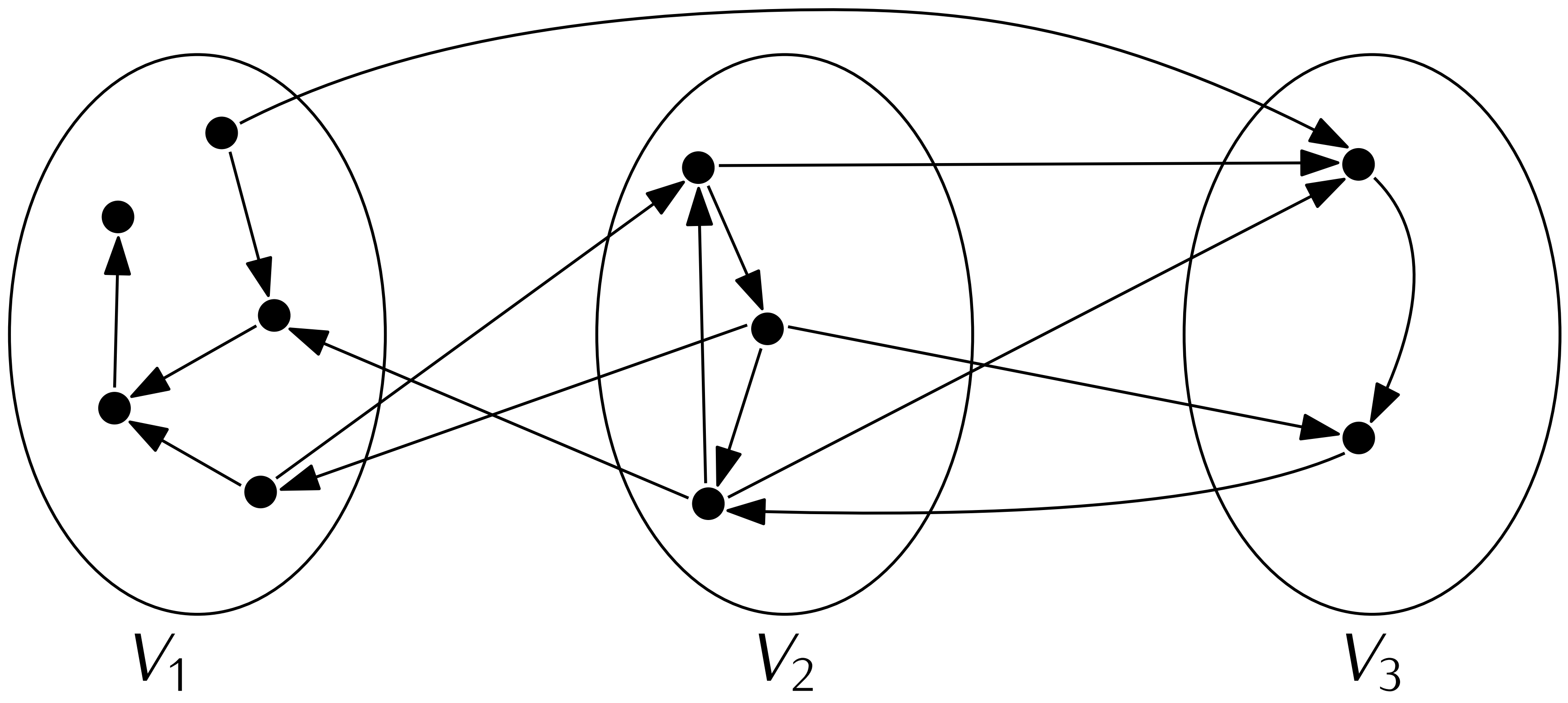
Theorem (**M.** LICS'25)**.** The following hold:
- Combinatorial CSP $\simeq$ $\mathrm{PCSP}(\text{combinatorial}, \text{finite})$
- There are finite PCSPs whose tractability can be shown by combinatorial CSPs and not by finite CSPs.
- There exists a uniform algorithm for temporal CSPs obtained by a reduction to finite PCSPs.


Ptime
FPC-definable
M.
LICS'25
Combinatorial CSP
MSO-definable
M.
LICS'25
M.
LICS'25
Singleton
AIP
Lasserre
hierarchy
LP+IP
hierarchy
Finite CSP
LP hierarchy
Affine integer
programming
Width hierarchy
M.
CSL'24
FO-definable

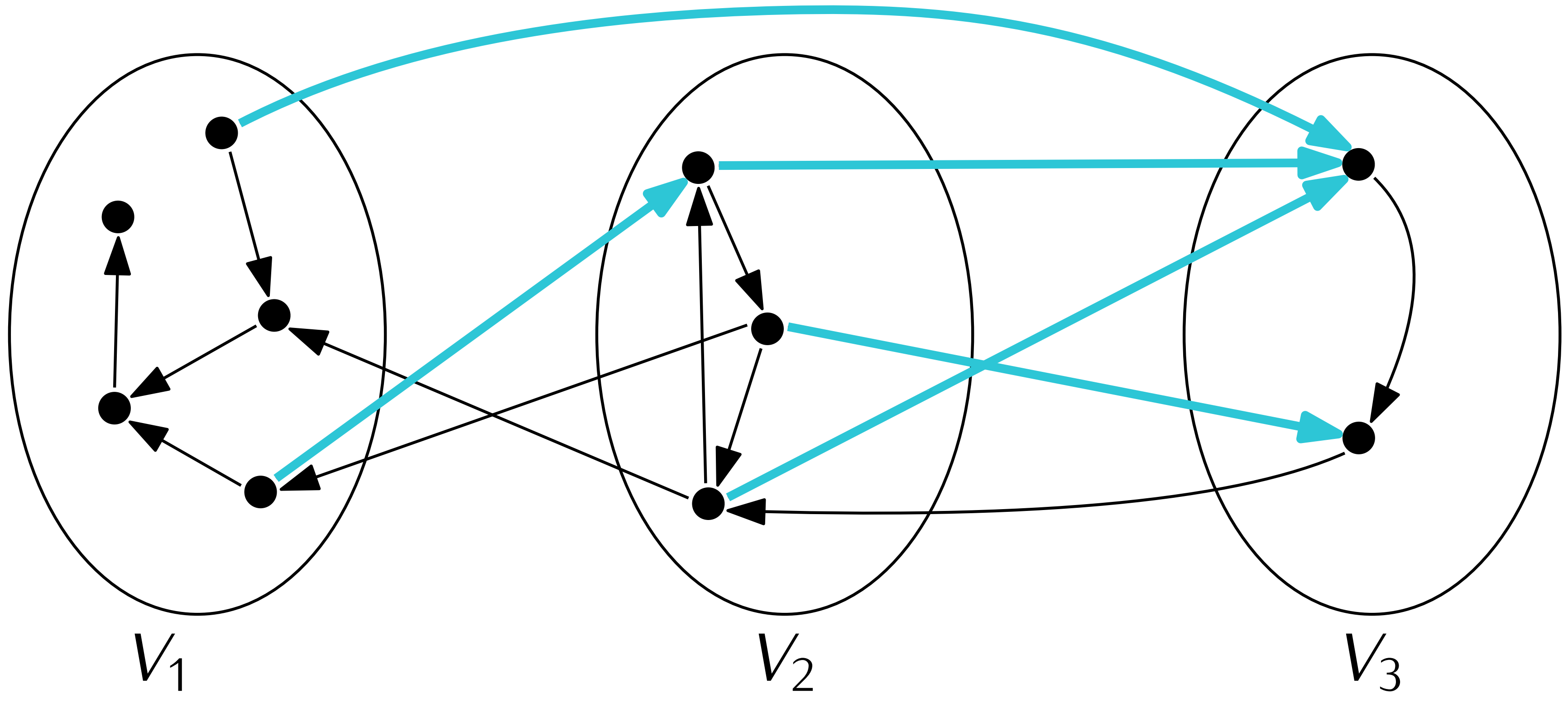
Research in Optimization



k-layering of a digraph: map $\ell\colon V \to \{1, \ldots, k\}$

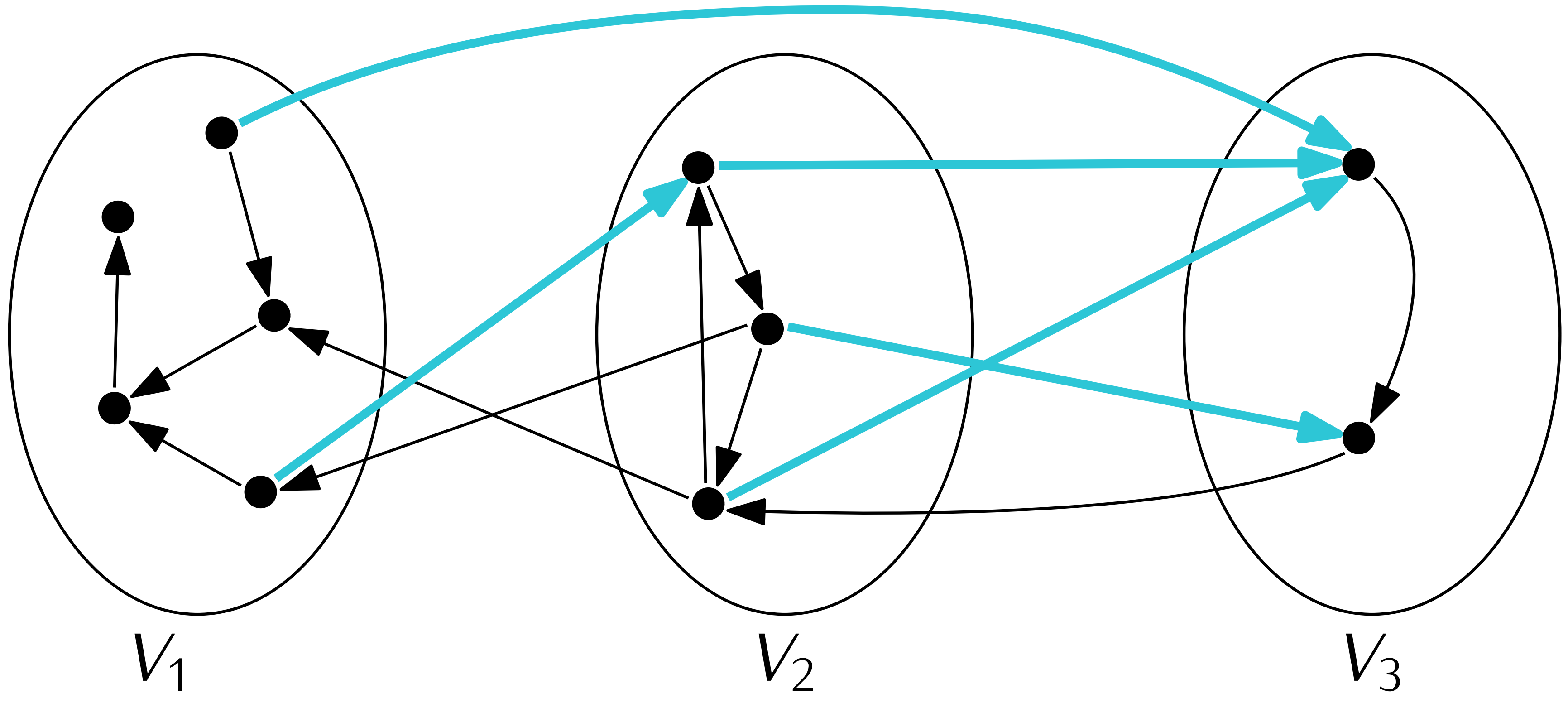


k-layering of a digraph: map $\ell \colon V \to \{1, \ldots, k\}$
Value of ℓ: $\#\{(u, v) \in E \mid \ell(u) < \ell(v)\}$



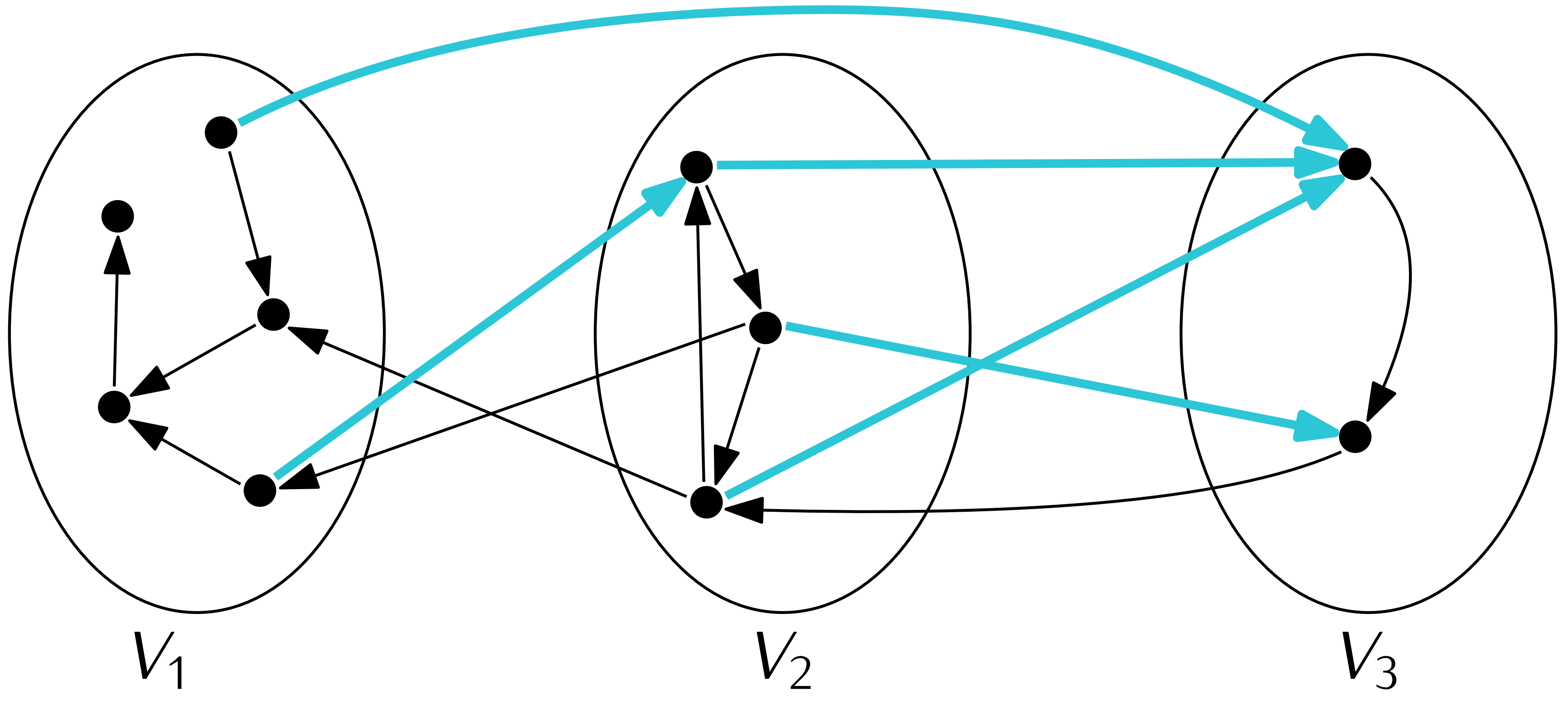
k-layering of a digraph: map $\ell\colon V \to \{1, \ldots, k\}$
Value of ℓ: $\#\{(u, v) \in E \mid \ell(u) < \ell(v)\}$

Note: every layering with value ρ is an acyclic subgraph with ρ edges

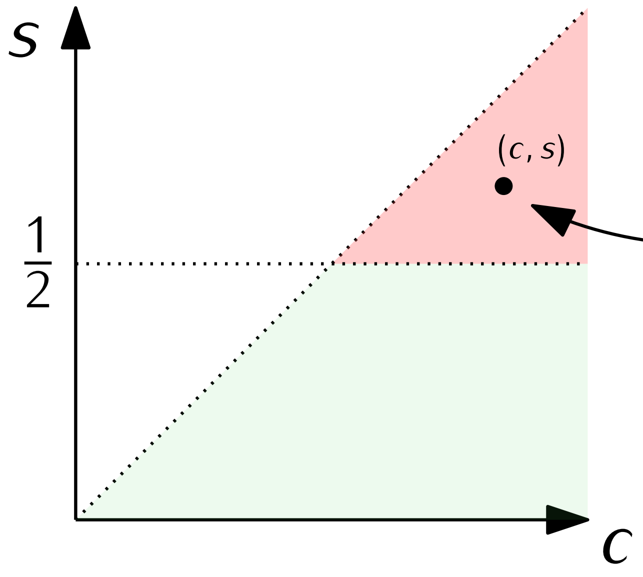


k-layering of a digraph: map $\ell\colon V \to \{1, \dots, k\}$
Value of ℓ: $\#\{(u, v) \in E \mid \ell(u) < \ell(v)\}$

Note: every layering with value ρ is an acyclic subgraph with ρ edges



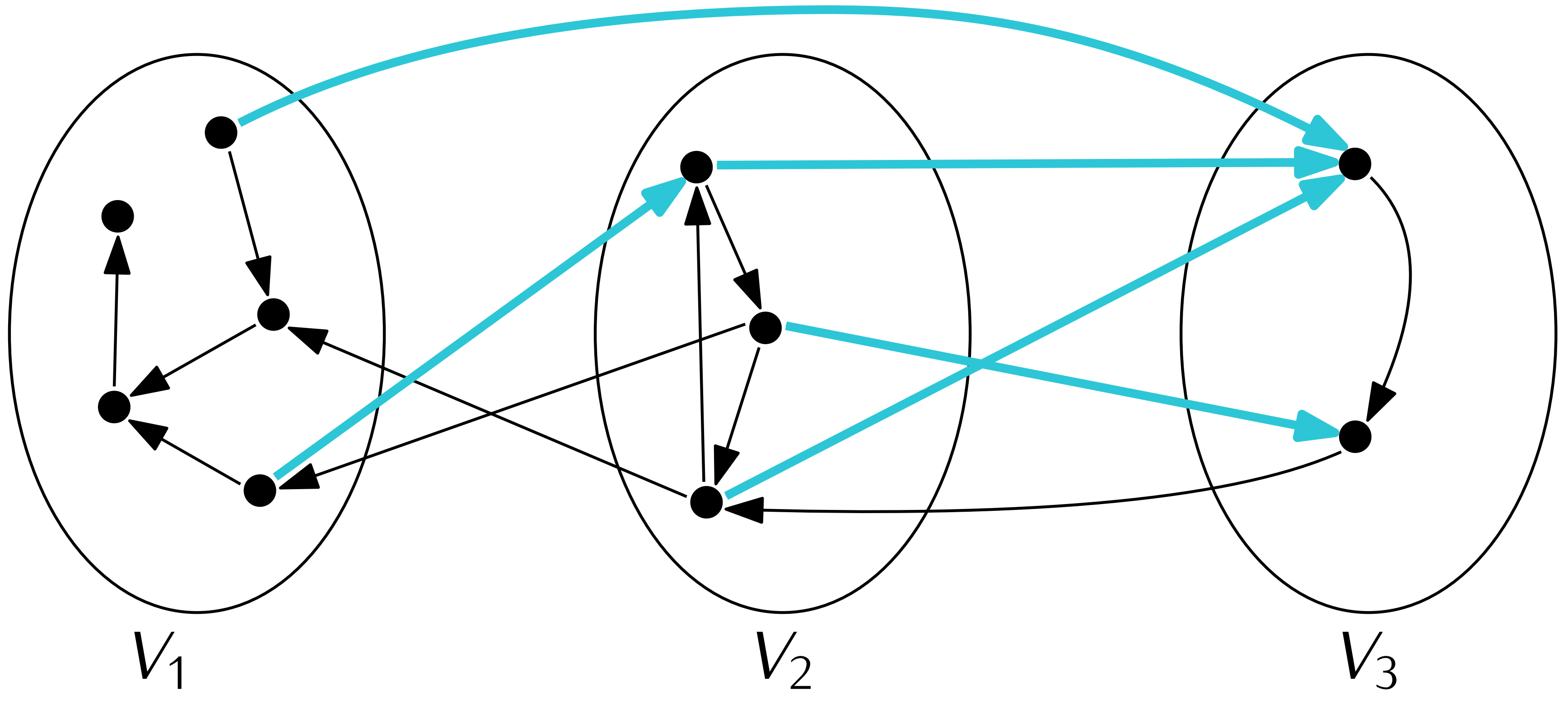
Approximation diagram for Maximum Acyclic Subgraph



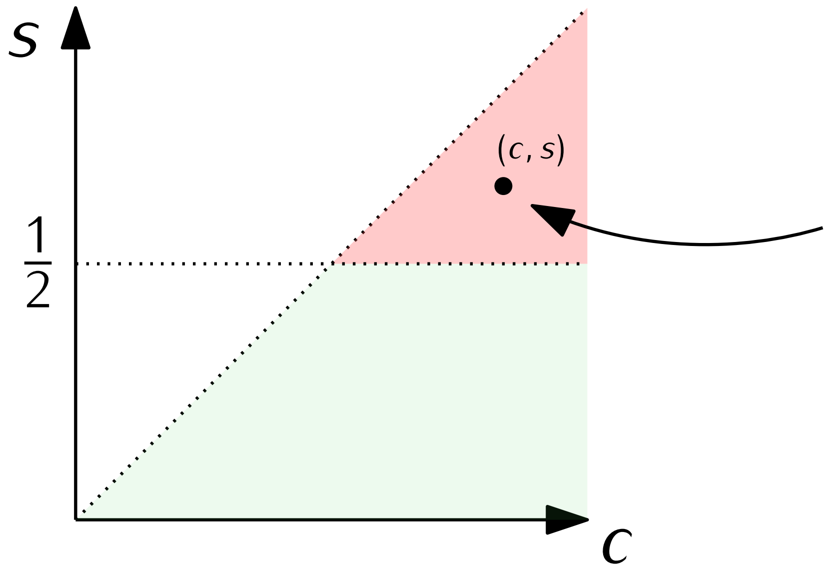
Given a digraph with an acyclic subgraph of weight $\geq c|E|$, find an acyclic subgraph with weight $\geq s|E|$.

k-layering of a digraph: map $\ell \colon V \to \{1, \dots, k\}$
Value of ℓ: $\#\{(u, v) \in E \mid \ell(u) < \ell(v)\}$

Note: every layering with value ρ is an acyclic subgraph with ρ edges



Approximation diagram for Maximum Acyclic Subgraph

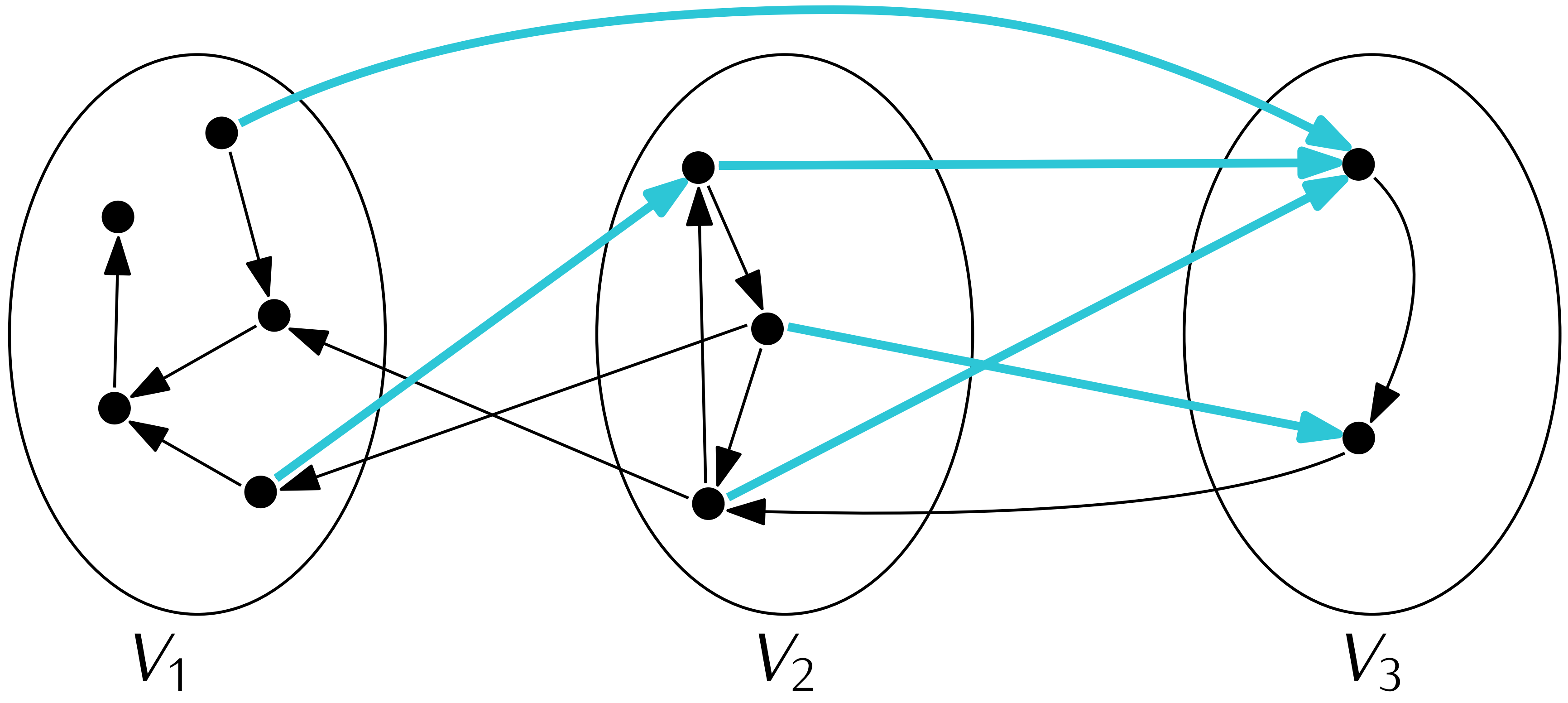


Given a digraph with an acyclic subgraph of weight $\geq c|E|$, find an acyclic subgraph with weight $\geq s|E|$.

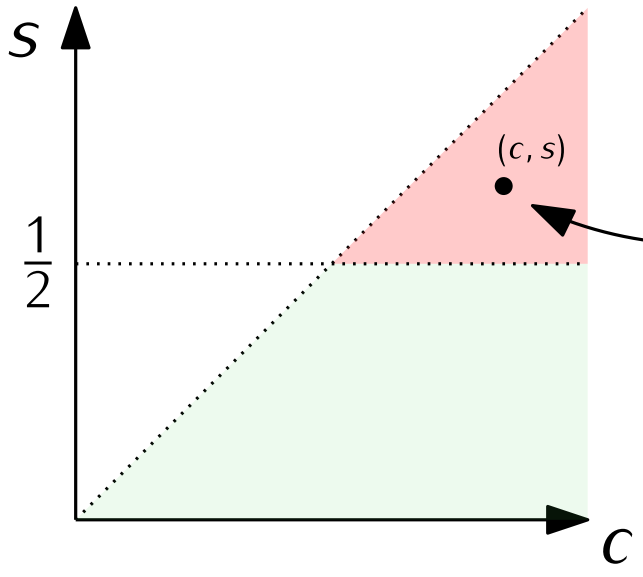
$\mathrm{PCSP}(\mathrm{LO}_k, \mathrm{LO}_\infty)$ (maximization version): given a directed graph G admitting a k-layering of value ρ, find an acyclic subgraph of G containing at least ρ edges.

k-layering of a digraph: map $\ell \colon V \to \{1, \ldots, k\}$
Value of ℓ: $\#\{(u, v) \in E \mid \ell(u) < \ell(v)\}$

Note: every layering with value ρ is an acyclic subgraph with ρ edges



Approximation diagram for Maximum Acyclic Subgraph



Given a digraph with an acyclic subgraph of weight $\geq c|E|$, find an acyclic subgraph with weight $\geq s|E|$.

$\mathrm{PCSP}(\mathrm{LO}_k, \mathrm{LO}_\infty)$ (maximization version): given a directed graph G admitting a k-layering of value ρ, find an acyclic subgraph of G containing at least ρ edges.

- Solvable in polynomial time for $k = 2$ (Nakajima-Živný, **M.** '25)
- NP-hard for $k \geq 4$, but unknown tractability boundary in approximation diagram
- Open complexity for $k = 3$