Algebraic Theory of Promise Constraint Satisfaction Problems, First Steps

Libor Barto

Department of Algebra, Charles University, Prague

FCT 2019, Copenhagen, 14 August □:□

European Research Council
Established by the European Commission

CoCoSym: Symmetry in Computational Complexity

This project has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No 771005)

Outline

Constraint Satisfaction Problems (CSPs) over finite templates

- class of computational problems
- goal: determine the computational complexity
- 3 step development of algebraic theory
- goal scored (two complexity classes: P, NP-complete)

Promise Constraint Satisfaction Problems (PCSPs)

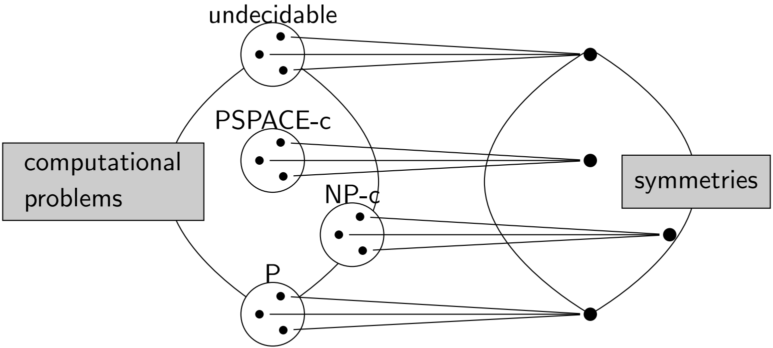
- larger class of computational problems, goal not scored
- richer on both algorithmic and hardness side
  - algorithms need to be infinitary
  - hardness requires heavy tools
- algebraic theory for CSP generalizes
- 4th step: drastic simplification of the basics

Barto, Bulín, Krokhin, Opršal: Algebraic approach to promise constraint satisfaction

Ideal world vs reality

$$\mathrm{CoolFunc}\colon \text{ computational problems} \longrightarrow \text{objects capturing symmetry}$$

$$\text{kernel of } \mathrm{CoolFunc} \;=\; \text{polynomial time reducibility}$$



(P)CSPs over fixed finite templates

- tiny portion of problems on the left
- kernel $\subsetneq$ polynomial time reducibility

CSP

Definition

Fix $\mathbb{A} = (A; R, S, \dots)$ relational structure

Definition $(\mathrm{CSP}(\mathbb{A}))$

Input: pp-sentence ϕ, eg. $(\exists x_1 \exists x_2 \dots) R(x_1, x_3) \wedge S(x_5, x_2) \wedge \dots$
Answer Yes: ϕ satisfied in $\mathbb{A}$
Answer No: ϕ not satisfied in $\mathbb{A}$

Search version: Find a satisfying assignment.
Search looks harder, but it's not [Bulatov, Jeavons, Krokhin'05]

Fact: Always in NP.

Example 1: 3-coloring

$\mathbb{K}_3 = (A; R)$ where

- $A = \{\mathit{lilac}, \mathit{mauve}, \mathit{cyclamen}\}$
- $R =$ (binary) inequality relation on A

Input of $\mathrm{CSP}(\mathbb{K}_3)$ is, e.g.

$(\exists x_1 \exists x_2 \dots \exists x_4) R(x_1, x_2) \wedge R(x_1, x_3) \wedge R(x_1, x_4) \wedge R(x_2, x_3) \wedge R(x_2, x_4)$

Viewpoint

- variables = vertices
- clauses (constraints) = edges

$\mathrm{CSP}(\mathbb{K}_3)$ is the 3-coloring problem for graphs

Fact: It is NP-hard (7-coloring NP-hard, 2-coloring in P)

Examples 2: hypergraph coloring problems

- $3\mathbb{NAE}_2 = (\{0,1\}; 3\mathit{NAE}_2)$ where
  $3\mathit{NAE}_2 =$ all but $\{(0,0,0),(1,1,1)\}$
  $\mathrm{CSP}(3\mathbb{NAE}_2) =$ positive not-all-equal 3-SAT
  $=$ 2-coloring problem for 3-uniform hypergraphs
- $3\mathbb{NAE}_4 = (\{0,1,2,3\}; 3\mathit{NAE}_4)$, where $3\mathit{NAE}_4$ still ternary
  $\mathrm{CSP}(3\mathbb{NAE}_4) =$ 4-coloring problem for 3-uniform hypergraphs
- $1\mathbb{IN}3 = (\{0,1\}; 1\mathit{in}3)$ where
  $1\mathit{in}3 = \{(0,0,1),(0,1,0),(1,0,0)\}$
  $\mathrm{CSP}(1\mathbb{IN}3) =$ positive 1-in-3 SAT

Fact: All NP-hard

Example 3: systems of linear equations

$3\mathbb{LIN}_5 = (\mathbb{Z}_5; L_{0000}, L_{0001}, \ldots, L_{4444})$ where e.g.

$L_{1234} = \{(x, y, z) \in \mathbb{Z}_5^3 : 1x + 2y + 3z = 4\}$

(note: relations are affine subspaces of $\mathbb{Z}_5^3$)

$\mathrm{CSP}(3\mathbb{LIN}_5) =$ solving systems of linear equations in $\mathbb{Z}_5$

Fact: In P

CSP and symmetry

Polymorphisms

polymorphism of $\mathbb{A}$: mapping $f : A^n \to A$
compatible with every relation

compatible with R: f applied component-wise to tuples in R
is a tuple in R

Example: $f(x_1, \ldots, x_4) = 2x_1 + 3x_2 + 3x_3 + 3x_4$ $\quad f : \mathbb{Z}_5^4 \to \mathbb{Z}_5$
is compatible with each L_{abcd}
because $f(\mathbf{v}_1, \ldots, \mathbf{v}_4)$ is an affine combination of these
vectors (as $2 + 3 + 3 + 3 = 1$)
and L_{abcd} is an affine subspace

$\mathrm{Pol}(\mathbb{A})$: the set of all polymorphisms (it is a "clone")
$=$ set of (multivariable) symmetries of $\mathbb{A}$

Algebraic theory, 1st step

Jeavons'98: On the algebraic structure of combinatorial problems

Theorem

Complexity of $\mathrm{CSP}(\mathbb{A})$ *is determined by* $\mathrm{Pol}(\mathbb{A})$*:*

If $\mathrm{Pol}(\mathbb{A}) \subseteq \mathrm{Pol}(\mathbb{B})$ *then* $\mathrm{CSP}(\mathbb{B})$ *reduces to* $\mathrm{CSP}(\mathbb{A})$*.*

Proof.

If $\mathrm{Pol}(\mathbb{A}) \subseteq \mathrm{Pol}(\mathbb{B})$, then relations in $\mathbb{B}$ can be defined from relations in $\mathbb{A}$ by a pp-formula.

[Geiger'69, Bondarčuk, Kalužnin, Kotov, Romov'69]

This gives a computational reduction of $\mathrm{CSP}(\mathbb{B})$ to $\mathrm{CSP}(\mathbb{A})$. □

So: $\mathrm{CSP}(3\mathbb{LIN}_5)$ is in P because $3\mathbb{LIN}_5$ has a lot of polymorphs

$\mathrm{CSP}(1\mathbb{IN}3)$ is NP-complete because $1\mathbb{IN}3$ has few

Systems of functional equations

System of functional equations is, e.g.

$$\begin{aligned} f(g(x,y),z) &= g(x,h(y,z)) \\ m(y,x,x) &= m(y,y,y) \\ m(x,x,y) &= m(y,y,y) \end{aligned}$$

Satisfied in $\mathcal{M}$, where $\mathcal{M}$ is a set of functions:
symbols can be interpreted in $\mathcal{M}$ so that
each equality is (universally) satisfied

Example: The above system is satisfied in Pol($3\mathbb{LIN}_5$):

- take $f(x,y) = g(x,y) = h(x,y) = x$
  (note: projections are always polymoprhisms)
- take $m(x,y,z) = x - y + z$

Algebraic theory, 2nd step

Bulatov, Jeavons, Krokhin'05: Classifying the complexity of constraints using finite algebras + Bodirsky'08: PhD thesis

Theorem

Complexity of $\mathrm{CSP}(\mathbb{A})$ *is determined by systems of functional equations satisfied in* $\mathrm{Pol}(\mathbb{A})$*:*

If each system satisfied in $\mathrm{Pol}(\mathbb{A})$ *is satisfied in* $\mathrm{Pol}(\mathbb{B})$*, then* $\mathrm{CSP}(\mathbb{B})$ *reduces to* $\mathrm{CSP}(\mathbb{A})$*.*

Proof.

Previous theorem, pp-definitions → pp-interpretations, the HSP theorem [Birkhoff'35] □

So: $\mathrm{CSP}(3\mathbb{LIN}_5)$ is in P because
$\mathrm{Pol}(3\mathbb{LIN}_5)$ satisfies strong systems of functional equations.

Algebraic theory, 3rd step

Barto, Opršal, Pinsker'18: The wonderland of reflections

minor condition = system of functional equations, each of the form

$$\mathit{symbol}(\mathit{variables}) = \mathit{symbol}(\mathit{variables}),$$

e.g. $m(y,x,x) = m(y,y,y)$, $m(x,x,y) = m(y,y,y)$

Theorem

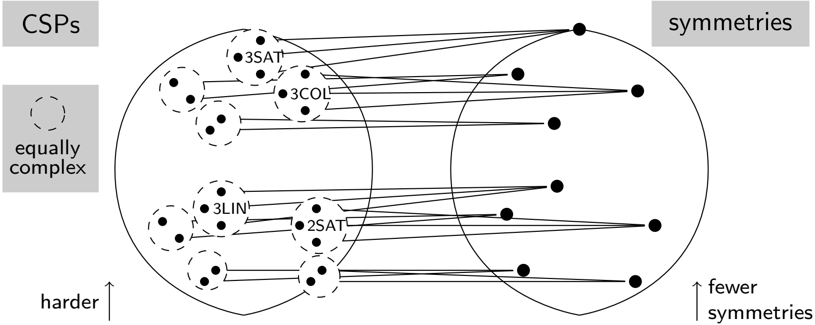
Complexity of $\mathrm{CSP}(\mathbb{A})$ *determined by*
minor conditions satisfied in $\mathrm{Pol}(\mathbb{A})$*:*

If each minor condition satisfied in $\mathrm{Pol}(\mathbb{A})$ *is satisfied in* $\mathrm{Pol}(\mathbb{B})$*,*
then $\mathrm{CSP}(\mathbb{B})$ *reduces to* $\mathrm{CSP}(\mathbb{A})$*.*

Proof.

pp-interpretation → pp-construction,
version of the HSP theorem. □

The Three Steps (movie)



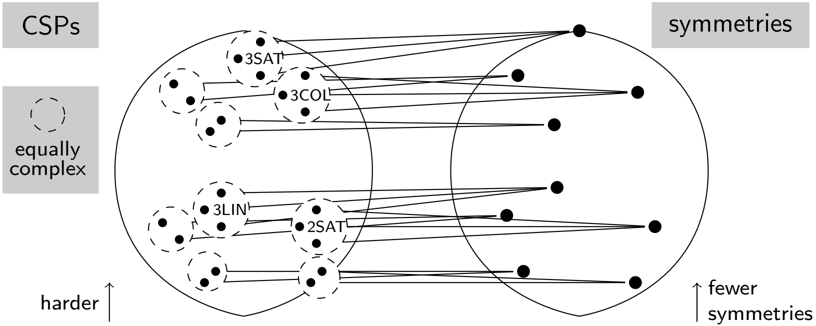
(1) polymorphisms

(2) systems of functional equations satisfied by polymorphisms

(3) minor conditions satisfied by polymorphisms

Where are the borderlines between complexity classes?

The Three Steps (movie)



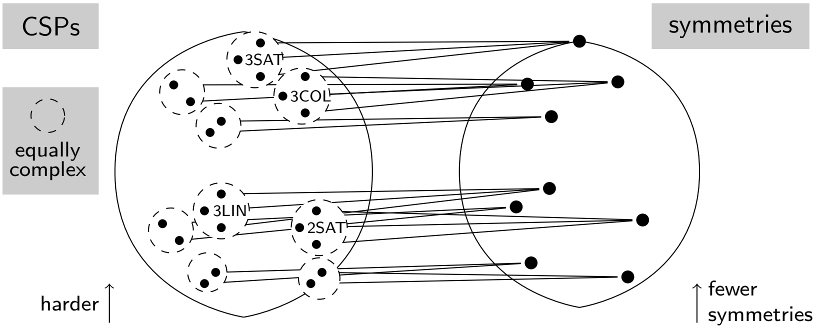
(1) polymorphisms

(2) systems of functional equations satisfied by polymorphisms

(3) minor conditions satisfied by polymorphisms

Where are the borderlines between complexity classes?

The Three Steps (movie)



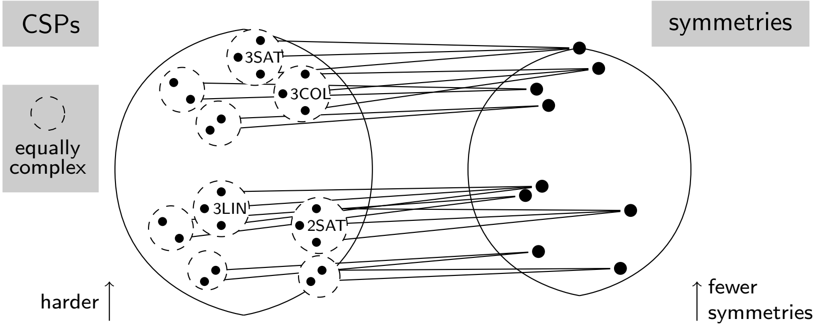
(1) polymorphisms

(2) systems of functional equations satisfied by polymorphisms

(3) minor conditions satisfied by polymorphisms

Where are the borderlines between complexity classes?

The Three Steps (movie)



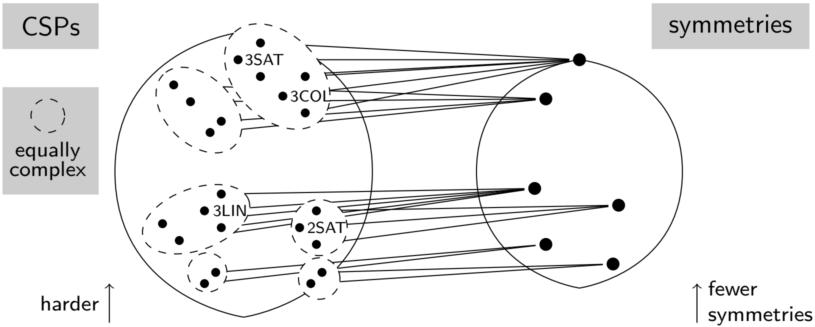
(1) polymorphisms

(2) systems of functional equations satisfied by polymorphisms

(3) minor conditions satisfied by polymorphisms

Where are the borderlines between complexity classes?

The Three Steps (movie)



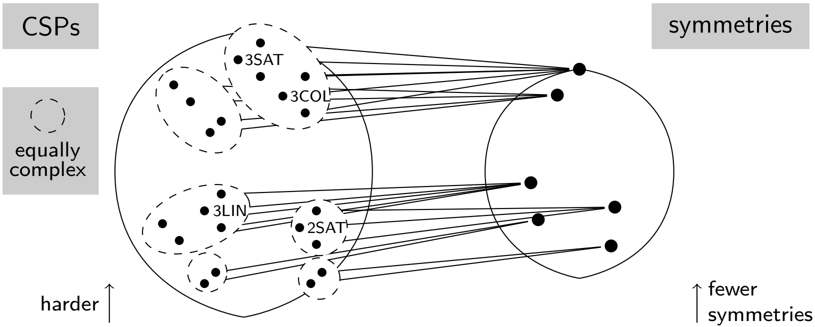
(1) polymorphisms

(2) systems of functional equations satisfied by polymorphisms

(3) minor conditions satisfied by polymorphisms

Where are the borderlines between complexity classes?

The Three Steps (movie)



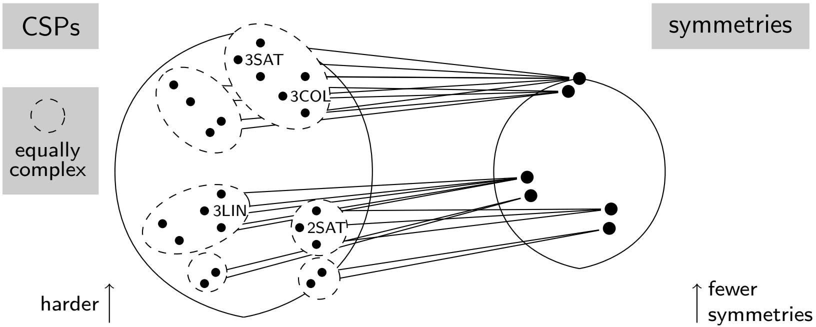
(1) polymorphisms

(2) systems of functional equations satisfied by polymorphisms

(3) minor conditions satisfied by polymorphisms

Where are the borderlines between complexity classes?

The Three Steps (movie)



CSPs

equally complex

harder

symmetries

fewer symmetries

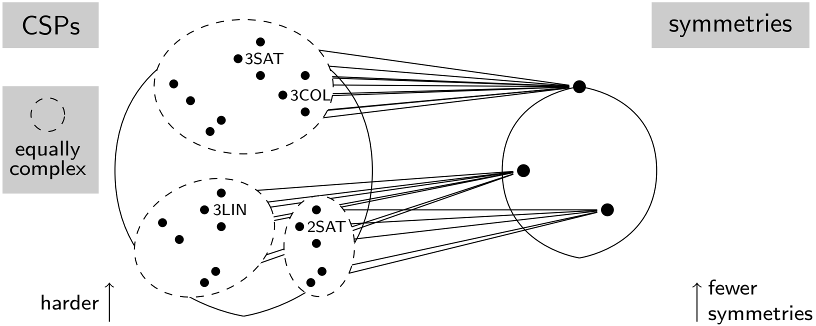
(1) polymorphisms

(2) systems of functional equations satisfied by polymorphisms

(3) minor conditions satisfied by polymorphisms

Where are the borderlines between complexity classes?

The Three Steps (movie)



(1) polymorphisms

(2) systems of functional equations satisfied by polymorphisms

(3) minor conditions satisfied by polymorphisms

Where are the borderlines between complexity classes?

The classification result

Minor condition is trivial:

satisfied in every Pol($\mathbb{A}$)

$=$ satisfied in $\mathcal{P}$, the set of projections on $\{0, 1\}$

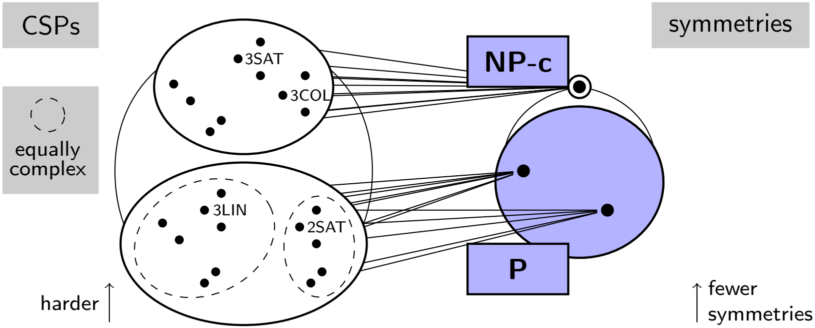
Corollary

If Pol($\mathbb{A}$) *satisfies only trivial minor conditions,*
then $\mathrm{CSP}(\mathbb{A})$ *is NP-hard.*

Theorem ([Bulatov'17], [Zhuk'17])

If Pol($\mathbb{A}$) *satisfies some non-trivial minor condition,*
then $\mathrm{CSP}(\mathbb{A})$ *is in P.*

Dichotomy



- only trivial minor conditions ⇒ NP-complete
- some nontrivial minor condition ⇒ P

Further steps?

PCSP

Relax!

$\mathrm{CSP}(\mathbb{A})$ is often NP-complete

What can we do?

1. **Approximation:** Try to satisfy only some fraction of the constraints, eg.

   for a satisfiable 3SAT instance,
   find an assignment satisfying at least 90% of the clauses
   Theorem: NP-hard [Håstad'01]

2. **PCSP:** Try to satisfy a relaxed version of all constraints, eg.

   for a 3-colorable graph,
   find a 37-coloring

Definition

Fix 2 relational structures in the same language

- $\mathbb{A} = (A; R^{\mathbb{A}}, S^{\mathbb{A}}, \dots)$
- $\mathbb{B} = (B; R^{\mathbb{B}}, S^{\mathbb{B}}, \dots)$
- there is a homomorphism $\mathbb{A} \to \mathbb{B}$ (eg. $A \subseteq B, R^{\mathbb{A}} \subseteq R^{\mathbb{B}}, \dots$)

Definition $(\mathrm{PCSP}(\mathbb{A}, \mathbb{B}))$

Input: pp-sentence ϕ, eg. $(\exists x_1 \exists x_2 \dots) R(x_1, x_3) \wedge S(x_5, x_2) \wedge \dots$
Answer Yes: ϕ satisfied in $\mathbb{A}$
Answer No: ϕ not satisfied in $\mathbb{B}$

Search version: Find a $\mathbb{B}$-satisfying assignment given an $\mathbb{A}$-satisfiable input.

(it may be a harder problem, we don't know)

Example 1: 4-coloring a 3-colorable graph

Recall: $\mathbb{K}_n = (\{1, 2, \ldots, n\}; \text{inequality})$

$\mathrm{PCSP}(\mathbb{K}_3, \mathbb{K}_4)$

Input: a graph
Answer Yes: it is 3-colorable
Answer No: it is not 4-colorable

Search version: Find a 4-coloring of a 3-colorable graph

Fun facts:

- **Theorem:** it is NP-hard [Brakensiek, Guruswami'16]
  (more generally $\mathrm{PCSP}(\mathbb{K}_n, \mathbb{K}_{2n-2})$ is NP-hard)
- $\mathrm{PCSP}(\mathbb{K}_n, \mathbb{K}_{2n-1})$ [Bulín, Krokhin, Opršal'19]
- $\mathrm{PCSP}(\mathbb{K}_n, \mathbb{K}_{\binom{n}{\lfloor n/2 \rfloor}-1})$, $n \geq 4$ [Wrochna, Živný]
- 6-coloring 3-colorable graph: complexity not known
- **Conjecture**: k-coloring l-colorable graph NP-hard ($k \geq l \geq 3$)

Example 2: hypergraph coloring

Recall: $3\mathbb{NAE}_k$ ternary not-all-equal relation on a k-element set

$\mathrm{PCSP}(3\mathbb{NAE}_2, 3\mathbb{NAE}_{137})$

Input: a 3-uniform hypergraph
Answer Yes: it is 2-colorable
Answer No: it is not 137-colorable

Theorem: It is NP-hard [Dinur,Regev,Smyth'05]
(more generally $\mathrm{PCSP}(3\mathbb{NAE}_l, 3\mathbb{NAE}_k)$ NP-hard
for every $k \geq l \geq 2$)

Proof uses

- the PCP theorem [Arora, Lund, Motwani, Sudan, Szegedy'98]
- $+$ the Parallel Repetition Thoerem [Raz'98]
- Lovász's theorem on Kneser's graphs [Lovász'78]

Example 3: 1-in-3 vs not-all-equal

Recall: $1\mathbb{IN}3 = (\{0,1\}; \{(1,0,0),(0,1,0),(0,0,1)\})$

$\mathrm{PCSP}(1\mathbb{IN}3, 3\mathbb{NAE}_2)$

Input: a 3-uniform hypergraph

Answer Yes: there is a 2-coloring such that
exactly one vertex in each hyperedge receives 1

Answer No: it is not 2-colorable

Fact: It is in P. Algorithm for finding a 2-coloring of a Yes input:

- for each hyperedge $\{x,y,z\}$ write $x+y+z=1$
- solve the system over $\mathbb{Q} \setminus \{\frac{1}{3}\}$ (it is solvable in $\{0,1\}$)
- assign $x \mapsto 1$ iff $x > 1/3$

Note: algorithm uses infinite domain CSP

Theorem: infinity is necessary [Barto'19]

PCSP and symmetry

Polymorphisms

polymorphism of $(\mathbb{A}, \mathbb{B})$: mapping $f : A^n \to B$
compatible with every relation-pair

compatible with $(R^{\mathbb{A}}, R^{\mathbb{B}})$: f applied to tuples in $R^{\mathbb{A}}$
is a tuple in $R^{\mathbb{B}}$

Example: $f(x_1, \ldots, x_{97}) = 1$ iff $\frac{\sum x_i}{97} > \frac{1}{3}$ $\quad f : \{0,1\}^{97} \to \{0,1\}$
is compatible with $(1in3, 3NAE_2)$

$\mathrm{Pol}(\mathbb{A}, \mathbb{B})$: the set of all polymorphisms (it is a "minion")
$=$ set of (multivariable) symmetries of $(\mathbb{A}, \mathbb{B})$

Algebraic theory, 3 steps

1st step (polymorphisms):
can be generalized [Brakensiek, Guruswami'18]
using [Pippenger'02]

2nd step (systems of functional equations):
makes no sense
since polymorphisms can no longer be composed

3rd step (minor conditions): the same as in CSP!

Algebraic theory, 4th step

Definition ($\mathrm{MinorCond}(N, \mathcal{M})$)

Input: minor condition **X** with symbols of arity N
Answer Yes: **X** is trivial (=satisfied in $\mathcal{P}$)
Answer No: **X** not satisfied in $\mathcal{M}$

Theorem ([Bulín, Krokhin, Opršal'19])

Let $\mathcal{M} = \mathrm{Pol}(\mathbb{A}, \mathbb{B})$. *The following computational problems are equivalent for a large enough* N.

(i) $\mathrm{CSP}(\mathbb{A})$

(ii) $\mathrm{MinorCond}(N, \mathcal{M})$

Consequence: 3rd step
Proof: direct, simple, known

Proof 1: Reduction from CSP

Given input of $\mathrm{CSP}(3\mathbb{NAE}_2)$, eg.

$$(\exists a, b, c, d)\ R(c, a, b) \wedge R(a, d, c)$$

transform it to a minor condition, eg.

$$\begin{aligned}
f_1(x_1, x_0, x_0, x_0, x_1, x_1) &= g_c(x_0, x_1)\\
f_1(x_0, x_1, x_0, x_1, x_0, x_1) &= g_a(x_0, x_1)\\
f_1(x_0, x_0, x_1, x_1, x_1, x_0) &= g_b(x_0, x_1)
\end{aligned}$$

$$\begin{aligned}
f_2(x_1, x_0, x_0, x_0, x_1, x_1) &= g_a(x_0, x_1)\\
f_2(x_0, x_1, x_0, x_1, x_0, x_1) &= g_d(x_0, x_1)\\
f_2(x_0, x_0, x_1, x_1, x_1, x_0) &= g_c(x_0, x_1)
\end{aligned}$$

"Yes input → Yes input": easy
"No input → No input": for contrapositive use $y \mapsto g_y(0, 1)$.

Proof 2: Reduction to CSP

Given a minor condition, e.g.

$$\begin{aligned} f(x_1, x_2, x_1, x_3) &= g(x_1, x_2, x_3) \\ h(x_3, x_1) &= g(x_1, x_2, x_3) \end{aligned}$$

- introduce variables f_{a_1,a_2,a_3,a_4} one for each $(a_1, \dots, a_4) \in A^4$, h_{a_1,a_2}, and g_{a_1,a_2,a_3}.
- so evaluation of f's $\leftrightarrow$ function $f : A^4 \to A$
- express that f, g, h are polymorphisms (by constraints)
- merge variables to enforce the equations

Remarks

Two classes of computational problems

General problem: Given a structure $\mathfrak{A}$ and 1st order sentence ϕ (the same language), decide whether $\mathfrak{A}$ satisfies ϕ.

CSP

- fix a finite relational structure
- restrict to primitive positive (pp-) sentences

Another problem: Given a structure $\mathfrak{A}$ and 1st order sentence ϕ (different language), decide whether symbols in ϕ can be interpreted in $\mathfrak{A}$ so that $\mathfrak{A}$ satisfies ϕ.

Our case: solving functional equations over an algebra

- fix a finite algebraic structure
- restrict to universally quantified conjunction of (special) equations
- take a promise version

Borderline for CSPs

Theorem

Let $\mathcal{M} = \mathrm{Pol}(\mathbb{A}, \mathbb{A})$. The following are equivalent.

- *$\mathcal{M}$ satisfies some nontrivial minor condition*
- *There is no mapping $\xi : \mathcal{M} \to \mathbb{N}$*
  - *if f is of arity n, then $\xi(f) \in \{1, 2, \dots, n\}$*
    *(**think**: an important coordinate of f)*
  - *ξ behaves nicely with minors*
- *$\mathcal{M}$ satisfies, for some $n \geq 2$, the minor condition*

$$c(x_1, x_2, \dots, x_n) = c(x_2, \dots, x_n, x_1)$$

  [Barto, Kozik'12]

- *. . .*
- *. . . zillion other characterizations . . .*
- *. . .*

A hardness criterion for PCSPs

Theorem

Let $\mathcal{M} = \mathrm{Pol}(\mathbb{A}, \mathbb{B})$*. If there exists* $C \in \mathbb{N}$ *and a mapping* $\xi : \mathcal{M} \to P(\mathbb{N})$ *such that*

- *if* f *is of arity* n*, then* $\xi(f) \subseteq \{1, 2, \ldots, n\}$, $|\xi(f)| \leq C$ *(**think**: a small set of important coordinates of* f*)*
- ξ *behaves nicely with minors*

Then $\mathrm{PCSP}(\mathbb{A}, \mathbb{B})$ *is NP-complete.*

Summary

Summary

CSP

- = problem about minor conditions
- Complexity captured by a piece of information about polymorphisms

PCSP is cool and fun

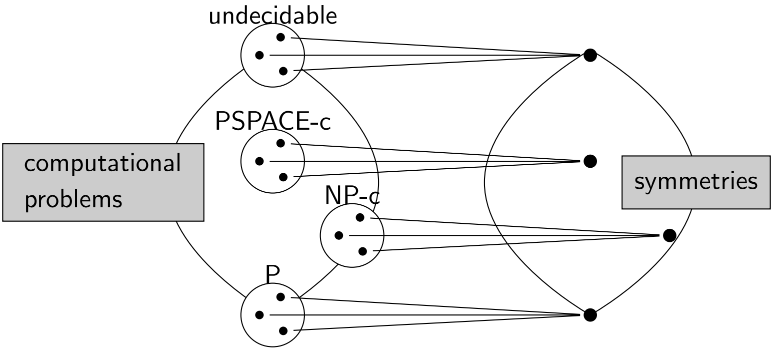
- Basics work but a lot is open: eg. borderlines, special cases
- More algorithms needed
- More interesting hardness proofs (PCP, topology)
- **Q:** What else can we forget about polymorphisms?

Reading

- Barto, Krokhin, Willard: Polymorphisms, and How to Use Them
- other surveys in this Dagstuhl Follow-Up volume
- Barto, Bulín, Krokhin, Opršal: Algebraic Approach to Promise Constraint Satisfaction

Ideal world

$$\begin{aligned}\mathrm{CoolFunc}\colon \text{ computational problems} &\longrightarrow \text{objects capturing symmetry}\\ \text{kernel of } \mathrm{CoolFunc} &= \text{polynomial time reducibility}\end{aligned}$$



Thank you!