**Topic:** Constraint Satisfaction Problem (CSP) over a fixed finite template
- a class of computational problems

**In this context**
- A problem is hard ⇔ it lacks symmetry
  - lacks symmetry ⇒ can simulate many problems ⇒ hard
    1 reason for hardness
  - symmetry can be exploited in algorithms (directly/indirectly)
    1 (?) algorithm scheme for all easy cases
- The most popular symmetries (eg. automorphisms) are useless

**In general**
- Goes beyond this particular class
- How far? **Still a big hole in the market**
Topic: CSP over a fixed finite template

In this context

- A problem is hard $\iff$ it lacks symmetry
  - lacks symmetry $\Rightarrow$ can simulate many problems $\Rightarrow$ hard
  - 1 reason for hardness
  - symmetry can be exploited in algorithms (directly/indirectly)
    - 1 (?) algorithm scheme for all easy cases
- The most popular symmetries (eg. automorphisms) are useless

In general

- Goes beyond this particular class
- How far? **Still a big hole in the market**
CSP over fixed finite template

Fix $\mathbb{A} = (A; R_1, R_2, \ldots, R_n)$: finite relational structure
each $R_i \subset A^k$ or $R_i : A^k \rightarrow \{\text{true, false}\}$

Definition

Instance of CSP($\mathbb{A}$): primitive positive sentence, eg.
$(\exists x)(\exists y)(\exists z)(\exists t) \ R_1(x, y, z) \land R_2(t, z) \land R_1(y, y, z)$
where each $R_i$ is in $\mathbb{A}$.
Question: Is it true?

- **Other variants**: infinite $A$; nothing is fixed; something else is fixed; different connectives
- **Other questions**
  - Count the number of solutions
  - Optimize the number of satisfied constraints
  - Approximately optimize the number of satisfied constraints
Examples and a conjecture

- 2-SAT: $\mathbb{A} = (\{0, 1\}; x \lor y, x \lor \neg y, \neg x \lor \neg y)$
- 3-SAT: $\mathbb{A} = (\{0, 1\}; x \lor y \lor z, x \lor y \lor \neg z, \ldots)$
- HORN-3-SAT: $\mathbb{A} = (\{0, 1\}; x = 0, x = 1, x \land y \rightarrow z)$
- Directed st–connectivity: $\mathbb{A} = (\{0, 1\}; x = 0, x = 1, x \leq y)$
- Undirected st–connectivity: $\mathbb{A} = (\{0, 1\}; x = 0, x = 1, x = y)$
- 3-COLOR: $\mathbb{A} = (\{0, 1, 2\}; x \neq y)$
- p-3-LIN: $\mathbb{A} = (GF(p); x + y + z = 0, x + 2y + 3z = 10, \ldots)$

Conjecture (The dichotomy conjecture [Feder and Vardi’93])

For every $\mathbb{A}$, $\text{CSP}(\mathbb{A})$ is either in $P$ or $\text{NP}$-complete.
Selected results

- **The dichotomy conjecture is true:**
  - if $|A| = 2$ [Schaefer’78]
  - if $\mathbb{A} = (A; R)$, $R$ is binary and symmetric [Hell and Nešetřil’90]
  - if $|A| = 3$ [Bulatov’06]
  - if $\mathbb{A}$ contains all unary relations [Bulatov’03 ’16] [Barto’11]
  - if $\mathbb{A} = (A; R)$ where $R$ is binary, without sources or sinks [Barto, Kozik, Niven’09]
  - in general? [Zhuk?]

- **Applicability of known algorithmic principles understood:**
  - Describing all solutions [Idziak, Markovic, McKenzie, Valeriote, Willard’07]
  - Local consistency (constraint propagation) [Barto, Kozik’09], [Bulatov]
  - All known tractable cases solvable by a combination of these two

- **Work on finer complexity classification**
Topic: CSP over a fixed finite template

In this context
- A problem is hard $\iff$ it lacks symmetry
  - lacks symmetry $\implies$ can simulate many problems $\implies$ hard
    - 1 reason for hardness
  - symmetry can be exploited in algorithms (directly/indirectly)
    - 1 (?) algorithm scheme for all easy cases
- The most popular symmetries (eg. automorphisms) are useless

In general
- Goes beyond this particular class
- How far? **Still a big hole in the market**
Example of simulation (gadget reduction, pp-definition)

- **A** = \((A; R)\), where \(R\) is ternary
- **B** = \((A; S, T)\), where \(S\) is binary and \(T\) is unary
  - \(S(x, y)\) iff \((\exists z) R(x, y, z) \land R(y, y, x)\)
  - \(T(x)\) iff \(R(x, x, x)\)

- Each instance of \(\text{CSP}(B)\), eg.

\[(\exists x)(\exists y)(\exists z) T(z) \land S(x, y)\]

- can be rewritten to an equivalent instance of \(\text{CSP}(A)\)

\[(\exists x)(\exists y)(\exists z)(\exists z') R(z, z, z) \land R(x, y, z') \land R(y, y, x)\]

- Thus \(\text{CSP}(B)\) is easier than \(\text{CSP}(B)\)
1 reason for hardness

- **Fact:** If
  - A pp-defines B
    - definition like in the previous slide
  - or more generally, A pp-interprets B
    - powers allowed ↔ variables encoded by tuples of variables
  - or more generally, A pp-constructs B
    - homomorphic equivalence allowed

  then CSP(B) is easier than CSP(A)

- **Corollary:** If A pp-constructs some structure with NP-hard CSP (like 3–SAT), then CSP(A) is NP-hard

- **Remark:** A pp-constructs 3–SAT ⇒ A pp-constructs every finite structure

- **Tractability conjecture:** If A does not pp-construct 3–SAT then CSP(A) is in P

  [Feder, Vardi’93] [Bulatov, Jeavons, Krokhin’00] [Bodirsky] [Willard]
Digression: Group theory vs. Universal algebra
Group theory, Semigroup theory

- **group**: algebraic structure \( G = (G; \cdot, -1, 1) \) satisfying \ldots
- **permutation group**: when \( G \) happens to be a set of bijections, \( \cdot \) is composition, \ldots

- **monoid**: algebraic structure \( M = (M; \cdot, 1) \) satisfying \ldots
- **transformation monoid**: \ldots

Universal algebra

- **algebra**: any algebraic structure \( Z = (Z; \text{some operations}) \)

Rants

- Model theorist: models of purely algebraic signature, why do you avoid relations?
- Algebraist: groups are complicated enough, nothing interesting can be said about general algebras
- All: have you ever seen a 37-ary operation? You shouldn’t study such a nonsense
Alternative viewpoint

<table>
<thead>
<tr>
<th><strong>unary invert. symmetries</strong></th>
<th>concrete</th>
<th>abstract</th>
</tr>
</thead>
<tbody>
<tr>
<td>unary symmetries</td>
<td>permutation group</td>
<td>group</td>
</tr>
<tr>
<td>higher arity symmetries</td>
<td>transformation monoid</td>
<td>monoid</td>
</tr>
<tr>
<td></td>
<td>function clone</td>
<td>abstract clone</td>
</tr>
</tbody>
</table>

- **permutation group**: Subset of \( \{ f : A \to A \} \) closed under composition and \( \text{id}_A \) and inverses. . .
  
  can be given by a generating unary algebra

- **group**: Forget concrete mappings, remember composition

- **function clone**: Subset of \( \{ f : A^n \to A : n \in \mathbb{N} \} \) closed under composition and projections
  
  can be given by a generating algebra

- **abstract clone**: Forget concrete mappings, remember composition
  
  aka variety, finitary monad over SET, Lawvere theory
End of digression
Topic: CSP over a fixed finite template

In this context

- A problem is hard ⇔ it lacks symmetry
  - lacks symmetry ⇒ can simulate many problems ⇒ hard
    - 1 reason for hardness
  - symmetry can be exploited in algorithms (directly/indirectly)
    - 1 (?) algorithm scheme for all easy cases
- The most popular symmetries (e.g., automorphisms) are useless

In general

- Goes beyond this particular class
- How far? Still a big hole in the market
Polymorphisms

Objects capturing symmetry of $\text{CSP}(\mathbb{A})$

- $\text{Aut}(\mathbb{A}) = \{ f : \mathbb{A} \to \mathbb{A} \text{ automorphism} \}$ automorphism group
- $\text{End}(\mathbb{A}) = \{ f : \mathbb{A} \to \mathbb{A} \text{ homomorph.} \}$ endomorphism monoid
- $\text{Pol}(\mathbb{A}) = \{ f : \mathbb{A}^n \to \mathbb{A} \text{ homomorphism} \}$ polymorphism clone

Trivial clone $\mathcal{T}$ – contains only projections

- aka 0,1,2

- Example: $\text{Pol}(3\text{-SAT})$
In this context

- A problem is hard ⇔ it lacks symmetry
  - lacks symmetry ⇒ can simulate many problems ⇒ hard
  - 1 reason for hardness
- symmetry can be exploited in algorithms (directly/indirectly)
  - 1 (?) algorithm scheme for all easy cases
- The most popular symmetries (eg. automorphisms) are useless

In general

- Goes beyond this particular class
- How far? Still a big hole in the market
**Expressive power and polymorphisms**

**Theorem** ([Birkhoff'35] [Geiger'68] [Bodnarčuk et al.'69] [Bodirsky] [Willard] [Barto, Opršal, Pinsker])

- **A pp-defines B** iff $\text{Pol}(A) \subseteq \text{Pol}(B)$
- **A pp-interprets B** iff $\text{Pol}(A) \rightarrow \text{Pol}(B)$ (homo)
- **A pp-constructs B** iff $\text{Pol}(A) \rightarrowrightarrow \text{Pol}(B)$ (h1 homo)

**Example:** 3–SAT pp-interprets every structure

**Remarks**

- Proofs constructive $\Rightarrow$ generic reductions
- $f : \text{Pol}(A) \rightarrow \text{Pol}(B)$ is a homo iff it preserves equations
  (eg. associative operation $\mapsto$ associative operation)
- $f : \text{Pol}(A) \rightarrowrightarrow \text{Pol}(B)$ is a h1 homo iff it preserves equations
  of height 1 (eg. commutative op. $\mapsto$ commutative op.)
Tractability conjecture again

Tractability conjecture

\[ \text{If } \not\exists \, \text{Pol}(\mathbb{A}) \rightarrow T, \text{ then } \text{CSP}(\mathbb{A}) \text{ in } P. \]

Recall: Otherwise \( \text{CSP}(\mathbb{A}) \) is NP-complete.

Theorem

TFAE

- A does not pp-construct all finite ie. \( \not\exists \) homo Pol(\( \mathbb{A} \) \( \rightarrow \) P ie. polymorphisms satisfy nontrivial equations
- Pol(\( \mathbb{A} \)) contains an operation \( s \) of arity 4 such that 
  \[ s(a, r, e, a) = s(r, a, r, e) \]
  \[ [\text{Siggers'10}, [\text{Kearnes, Marković, McKenzie'14}] \]
- Pol(\( \mathbb{A} \)) contains an operation \( c \) of arity \( > 1 \) such that 
  \[ c(a_1, a_2, \ldots, a_n) = c(a_2, \ldots, a_n, a_1) \]
  \[ [\text{Barto, Kozik'12}] \]

3rd and 4th items: concrete and positive alternatives
Tractability conjecture vs. reality

Conjecture

\[ TFAE \ (if \ P \neq NP) \]

- CSP(\(A\)) is in P
- \(A\) has a polymorphism \(s\) such that \(s(a, r, e, a) = s(r, a, r, e)\)

Even if the conjecture is wrong, we know that CSP(\(A\)) depends only on height 1 equations
Topic: CSP over a fixed finite template

In this context

- A problem is hard ⇔ it lacks symmetry
  - lacks symmetry ⇒ can simulate many problems ⇒ hard
    1 reason for hardness
  - symmetry can be exploited in algorithms (directly/indirectly)
    1 (?) algorithm scheme for all easy cases
- The most popular symmetries (eg. automorphisms) are useless

In general

- Goes beyond this particular class
- How far? Still a big hole in the market
Endomorphism monoids are useless

- $\forall A \exists B$ such that
  - $A$ pp-constructs $B$ pp-constructs $A$ (ie. the same complexity)
  - $\text{Aut}(B) = \text{End}(B) = \{\text{id}_B\}$

- $\forall A, B$ there is $\text{End}(A) \rightarrow \text{End}(B)$
In this context

- A problem is hard ⇔ it lacks symmetry
  - lacks symmetry ⇒ can simulate many problems ⇒ hard
    - 1 reason for hardness
  - symmetry can be exploited in algorithms (directly/indirectly)
    - 1 (?) algorithm scheme for all easy cases
- The most popular symmetries (eg. automorphisms) are useless

In general

- Goes beyond this particular class
- How far? Still a big hole in the market
How universal algebra helps in CSP
  ▶ tools
  ▶ identifying intermediate cases

How polymorphisms are used
  ▶ Direct: A way to combine solutions to get another solution
  ▶ Indirect: Proving correctness

Two algorithmic ideas:
  ▶ Describe all solutions (direct)
  ▶ Refute unsolvable instances by enforcing consistency (indirect)
Describing all solutions

- Consider an instance of $\text{CSP}(\mathbb{A})$ with $n$ variables
- The set of solutions is $S \subseteq A^n$ invariant under $\text{Pol}(\mathbb{A})$
- Can happen $|S| = |A|^n \Rightarrow$ cannot list all solutions
- **Idea:** find a generating set of $S$, needs to be small
- Example: $\text{CSP}(\mathbb{A}) = p\text{-LIN}$
  - $\text{Pol}(\mathbb{A}) =$ affine combinations
  - $S$ is affine subspace of $GF(p)^n$
  - $S$ has generating set of size $\leq (n + 1)$
  - eg. $A^2$ generated by $(0, 0), (0, 1), (1, 0)$
- $\text{UA} \Rightarrow$ obvious more general polymorphisms to look at
  Malcev [Bulatov’02], [Bulatov, Dalmau’06]
- $\text{UA} \Rightarrow$ another class where small generating sets exist
  Near unanimity [Baker, Pixley’75]
- $\text{UA} \Rightarrow$ class covering these two
Theorem (Berman, Idziak, Markovic, McKenzie, Valeriote, Willard’10)

TFAE

- All invariant $n$-ary relations have small generating sets ($\leq$ polynomial in $n$)
- The number of $n$-ary invariant relations is small ($\leq$ exponential in $n$)

In this case, $\text{CSP}(A)$ is in $P$. Moreover, a generating set of all solutions can be found in $P$–time.
Local consistency

**Roughly:** \( \Delta \) has **bounded width** iff \( \text{CSP}(\Delta) \) can be solved by checking local consistency

**More precisely:**

- Fix \( k \leq l \) (integers)
- \((k, l)\)-algorithm: Derive the strongest constraints on \( k \) variables which can be deduced by “considering” \( l \) variables at a time.
- If a contradiction is found, answer “no” otherwise answer “yes”
- “no” answers are always correct
- if “yes” answers are correct for every instance of \( \text{CSP}(\Delta) \) we say that \( \Delta \) has **width** \((k, l)\).
- if \( \Delta \) has width \((k, l)\) for some \( k, l \) then \( \Delta \) has **bounded width**

Various equivalent formulations (bounded tree width duality, definability in Datalog, least fix point logic)
Local consistency 2

- A has a semilattice polymorphism $\Rightarrow$ CSP($A$) has width 1 [Feder, Vardi'93]
- A has a near unanimity polymorphism of arity $(n + 1)$ $\Rightarrow$ CSP($A$) has width $n$ [Feder, Vardi'93]
- $p$-LIN does not have bounded width [Feder, Vardi'93]
- Conjecture: A has bounded width iff A does not pp-construct $p$–LIN [Larose, Zádori'07]
- UA suggests what to do next
  - 2-semilattices [Bulatov'06]
  - CD(3) [Kiss, Valeriote'07]
  - CD(4) [Carvalho, Dalmau, Marković, Maróti'09]
  - CD [Barto, Kozik'09]
Theorem

TFAE

1. $\mathcal{A}$ does not pp-construct $p$-LIN
2. $\mathcal{A}$ has bounded width [Barto, Kozik’09]
3. $\mathcal{A}$ has width $(2, 3)$ [Barto’16] [Bulatov]
4. $\text{CSP}(\mathcal{A})$ is decided by singleton arc consistency [Kozik]
5. the canonical semidefinite programming relaxation correctly decides $\text{CSP}(\mathcal{A})$ [Barto, Kozik’16]
Topic: CSP over a fixed finite template

In this context

- A problem is hard ⇔ it lacks symmetry
  - lacks symmetry ⇒ can simulate many problems ⇒ hard
  1 reason for hardness
- symmetry can be exploited in algorithms (directly/indirectly)
  1 (?) algorithm scheme for all easy cases

- The most popular symmetries (eg. automorphisms) are useless

In general

- Goes beyond this particular class
- How far? Still a big hole in the market
Optimisation
- Complexity captured by weighted polymorphisms
  [Cohen, Cooper, Creed, Jeavons, Živný’13]
- Even for valued CSP
- Tractability conjecture $\Rightarrow$ dichotomy for optimisation
  [Kolmogorov, Krokhin, Rolínek’15]

Exact counting $\#$ solutions
- Complexity captured by polymorphisms
  [Bulatov, Dalmau’03] [Bulatov, Grohe’05]
- Dichotomy [Bulatov’08] [Dyer, Richerby’10]

Robust satisfiability – almost solutions on almost satisfiable instances
- Complexity captured by polymorphisms [Dalmau, Krokhin’11]
- Dichotomy: in P if doesn’t pp-construct $p$-LIN, otherwise NP-c
  [Hastad’01] [Barto, Kozik’12]
Infinite domains

- All decision problems up to P–time reductions [Bodirsky, Grohe’08]
- Restrict to ω-categorical (aka oligomorphic)
  - Complexity captured by polymorphisms [Bodirsky, Nešetřil’06]
  - Actually abstract polymorphism clone + topology [Bodirsky, Pinsker’15]
  - Still “almost” covers all decision problems [Bodirsky, Grohe’08]
- Restrict even more
  - back to NP
  - P/NP-c dichotomy conjecture
    [Bodirsky, Pinsker’11], [Barto, Pinsker’16] [Barto, Opršíal, Pinsker]
    [Olšák]
Topic: CSP over a fixed finite template

In this context
- A problem is hard ⇔ it lacks symmetry
  - lacks symmetry ⇒ can simulate many problems ⇒ hard
    1 reason for hardness
  - symmetry can be exploited in algorithms (directly/indirectly)
    1 (?) algorithm scheme for all easy cases
- The most popular symmetries (eg. automorphisms) are useless

In general
- Goes beyond this particular class
- How far? Still a big hole in the market
How far?

**Optimistic**: everywhere

**Realistic – ish**

- Approximation
  - Complexity captured by approximate polymorphisms if UGC [Raghavendra’08]
  - Challenge: hardness part
- Hybrid CSPs (edge CSP, planar CSP, . . .)
  - eg. Perfect matching problem in graphs
  - What is the right notion of symmetry?
- Approximate hybrid, approximate counting, hybrid counting
  - eg. Holant problems
  - What is the right notion of symmetry?
- Infinite domain CSP
  - Explore the theory for larger classes (eg. to include linear programming)
  - Criterion for undecidability?
Big hole in the market

Do you see gadgets? Find symmetry!
Topic: CSP over a fixed finite template

In this context

- A problem is hard $\iff$ it lacks symmetry
  - lacks symmetry $\Rightarrow$ can simulate many problems $\Rightarrow$ hard
    1 reason for hardness
  - symmetry can be exploited in algorithms (directly/indirectly)
    1 (?) algorithm scheme for all easy cases

- The most popular symmetries (e.g. automorphisms) are useless

In general

- Goes beyond this particular class
- How far? **Still a big hole in the market**

Thank you!