Constraint Satisfaction Problem over a Fixed Template

Libor Barto

Charles University in Prague

Highlights of Logic, Games and Automata
Prague, 15 September 2015

Constraint satisfaction problem (CSP)

- Common framework for many real-life problems
- Not the topic of this tutorial

- We will restrict to a tiny subclass – CSPs over a finite template
- We will study computational complexity of these problems (mainly NP versus P)

CSP over a finite template

- Common framework for some computational problems
  - Broad enough to include interesting examples
  - Narrow enough to make significant progress (on all problems within a class, rather than just a single computational problem)
  - Generalizations to broader classes of problems
- Main achievement: better understanding why problems are easy or hard:
  - Hardness comes from lack of symmetry
  - Symmetries of higher arity are important (not just automorphisms or endomorphisms) $\rightarrow$ **universal algebra** (not just group or semigroup theory)
- Long term goal: go beyond CSP

Instance of the CSP

Definition

Instance of the CSP is a list of constraints – expression of the form

$$R_1(x, y, z),\ R_2(t, z),\ R_1(y, y, z),\ \dots$$

where R_i are relations on a common domain A

(subsets of A^k or mappings $A^k \to \{true, false\}$).

Assignment = mapping *variables* → *domain*

- **Satisfiability problem:** Is there an assignment satisfying all constraints (a solution)
- **Search problem:** Find a solution
- **Counting CSP:** How many solutions are there?
- **Max-CSP:** Find a map satisfying maximum number of constraints
- **Approx. Max-CSP:** Find a map satisfying $0.7 \times$ *Optimum* constraints
- **Robust CSP:** Find an almost satifying assignment given an almost satisfiable instance

CSP over a fixed template (aka constraint language)

Definition

$\mathcal{A} = (A; R_1, R_2, \ldots, R_k)$: relational structure with A finite
Instance of $CSP(\mathcal{A})$: Expression of the form

$$R_1(x, y, z),\ R_2(t, z),\ R_1(y, y, z),\ \ldots$$

where each R_i is in $\mathcal{A}$.

- What is the computational (or descriptive) complexity for fixed $\mathcal{A}$?
- **This tutorial:** Satisfiability problem for $CSP(\mathcal{A})$
- **Other interesting problems:**
  - restrict something else than the set of allowed relations
  - allow infinite A
  - allow weighted relations: mappigs $A^k \to Q \cup \{\infty\}$
  - (approximate) counting, Max-CSP, Approx Max-CSP

3 formulations of CSP($\mathcal{A}$)

- Basic form
  Instance: List of constraints over $\mathcal{A}$
  Question: Is there a satisfying assignment variables $\rightarrow$ domain?

- Logical version
  Instance: Sentence ϕ in the language of $\mathcal{A}$ with $\exists$ and $\wedge$
  Question: Is ϕ true in $\mathcal{A}$?

- Homomorphism version
  Instance: Relational structure $\mathcal{B}$ of the same type as $\mathcal{A}$
  Question: Is there a homomorphism $\mathcal{B} \rightarrow \mathcal{A}$?

Example 3-SAT (NP-complete)

$\mathcal{A} = (\{0,1\};\ R_{000}, R_{001}, R_{011}, R_{111})$

$R_{000}(x,y,z)$ iff $x \vee y \vee z$ $\quad R_{000} =$ all triples but $(0,0,0)$
$R_{001}(x,y,z)$ iff $x \vee y \vee \neg z$ $\quad R_{001} =$ all triples but $(0,0,1)$
$R_{011}(x,y,z)$ iff $x \vee \neg y \vee \neg z$ $\quad R_{011} =$ all triples but $(0,1,1)$
$R_{111}(x,y,z)$ iff $\neg x \vee \neg y \vee \neg z$ $\quad R_{111} =$ all triples but $(1,1,1)$

Instance: $R_{001}(x_1,x_4,x_7),\ R_{001}(x_2,x_2,x_6),\ R_{111}(x_2,x_1,x_5)$
Meaning: $x_1 \vee x_4 \vee \neg x_7,\ x_2 \vee x_2 \vee \neg x_6,\ \neg x_2 \vee \neg x_1 \vee \neg x_5$
Question: Is there a satisfying assignment $\{x_1,x_2,\dots\} \to \{0,1\}$?

Inst: $\exists x_1,x_2,\dots R_{001}(x_1,x_4,x_7) \wedge R_{001}(x_2,x_2,x_6) \wedge R_{111}(x_2,x_1,x_5)$
Quest: Is it true?

Inst: $\mathcal{B} = (B; S_{000}, S_{001}, S_{011}, S_{111})$, where $B = \{x_1,x_2,\dots\}$,
$S_{000} = \emptyset,\ S_{001} = \{(x_1,x_4,x_6),(x_2,x_2,x_6)\}$,
$S_{011} = \emptyset,\ S_{111} = \{(x_2,x_1,x_5)\}$
Quest: Is there a homomorphism $\mathcal{B} \to \mathcal{A}$?

Some other Boolean templates

- **1-in-3-SAT** (NP-complete): $\mathcal{A} = (\{0,1\}; R)$, $R = \{(0,0,1),(0,1,0),(1,0,0)\}$
- **NAE-3-SAT** (NP-complete): $\mathcal{A} = (\{0,1\}; R)$, $R =$ all triples but $\{(0,0,0),(1,1,1)\}$
- **2-SAT** (in P, NL-complete): $\mathcal{A} = (\{0,1\}; R_{00}, R_{01}, R_{11})$
- **HORN-3-SAT** (in P, P-complete): $\mathcal{A} = (\{0,1\}; C_0, C_1, R_{011}, R_{111})$, $C_0 = \{0\}$, $C_1 = \{1\}$, $R_{011}(x,y,z)$ iff $y \wedge z \to x$, $R_{111}(x,y,z)$ iff $y \wedge z \to \neg x$
- **Digraph unreachability** (in P, NL-complete): $\mathcal{A} = (\{0,1\}; C_0, C_1, \leq)$
- **Graph unreachability** (in P, L-complete): $\mathcal{A} = (\{0,1\}; C_0, C_1, =)$

Examples on larger domains

- **k-COLOR** (L-complete for $k \leq 2$, NP-complete for $k > 3$): $\mathcal{A} = (\{1, \ldots, k\};\ \neq)$

- $\mathbb{Z}_p$-**3-LIN** (in P): $\mathcal{A} = (\mathbb{Z}_p;$ affine subspaces of $\mathbb{Z}_p^3)$

The dichotomy conjecture

A largest natural class of problems with a dichotomy?

Conjecture (The dichotomy conjecture Feder and Vardi'93)

For every $\mathcal{A}$, $\mathrm{CSP}(\mathcal{A})$ is either in P or NP-complete.

- Evidence (in 93):
  - True for $|A| = 2$ Schaefer'78
  - True if $\mathcal{A} = (A; R)$, R is binary and symmetric Hell and Nešetřil'90
- Feder and Vardi suggested that tractability is tied to "closure properties"
- $\rightarrow$ algebraic approach Bulatov, Jeavons, Krokhin'00

Reductions

Reductions and universal algebra

- Write $\mathrm{CSP}(\mathcal{A}) \leq \mathrm{CSP}(\mathcal{B})$ if $\mathrm{CSP}(\mathcal{A})$ is "at most as hard as" $\mathrm{CSP}(\mathcal{B})$ (precise meaning: log-space reducible)
- **Crucial**: pp-interpretations give reductions
- pp-interpretations are (indirectly) the main subject of universal algebra

Plan for the rest:

- reductions in relational language
- algebra
- results

pp-definitions give reductions

Definition

Let $\mathcal{A}, \mathcal{B}$ be relational structures with common domain $A = B$.
We say that $\mathcal{A}$ pp-defines $\mathcal{B}$ if each relation in $\mathcal{B}$ can be defined by a first order formula which uses relations in $\mathcal{A}$, $=$, $\wedge$ and $\exists$.

Will also use "$\mathcal{A}$ pp-defines a relation R",
"R is pp-definable from $\mathcal{A}$", etc

Theorem

If $\mathcal{A}$ pp-defines $\mathcal{B}$, then $\mathrm{CSP}(\mathcal{B}) \leq \mathrm{CSP}(\mathcal{A})$.

Proof in a moment

Examples and exercises

- the template of 3-SAT $\mathcal{A} = (\{0,1\}; R_{000}, R_{001}, R_{011}, R_{111})$ pp-defines
  - each ternary relation
  - each unary and binary relation
  - the 4-ary relation $R_{0000} =$ all tuples but $(0,0,0,0)$
  - all relations
  - for each $\mathcal{B} = (\{0,1\}, \dots)$, $\mathcal{A}$ pp-defines $\mathcal{B}$.
    Thus $\mathrm{CSP}(\mathcal{B}) \leq$ 3-SAT.
- (the template of) 1-in-3-SAT
  $\mathcal{A} = (\{0,1\};\ \{(0,0,1),(0,1,0),(1,0,0)\})$ pp-defines
  (the template) of 3-SAT
- NAE-SAT $\mathcal{A} = (\{0,1\};$ all triples but $(0,0,0),(1,1,1))$
  does not pp-define 3-SAT
  - it even does not define $C_0 = \{0\}$ – why?
- HORN-SAT $\mathcal{A} = (\{0,1\};\ C_0, C_1, R_{011}, R_{111})$
  does not pp-define 3-SAT – why?

pp-definitions give reductions - proof

Theorem

If $\mathcal{A}$ pp-defines $\mathcal{B}$, then $\mathrm{CSP}(\mathcal{B}) \leq \mathrm{CSP}(\mathcal{A})$.

- Say $\mathcal{A} = (A; R)$, $\mathcal{B} = (A;\ S, T)$, where
  $S(x, y)$ iff $(\exists z)\ R(x, y, z) \wedge R(y, y, x)$
  $T(x, y)$ iff $R(x, x, x) \wedge (x = y)$
- Reduction of $\mathrm{CSP}(\mathcal{B})$ to $\mathrm{CSP}(\mathcal{A})$:
- Say, our instance is
  $(\exists x_1, x_2, x_3, x_4)\ S(x_3, x_2) \wedge T(x_1, x_4) \wedge S(x_2, x_4)$
- Rewrite using the definitions:
  $(\exists x_1, x_2, x_3, x_4,\ y_1, y_2)\ R(x_3, x_1, y_1) \wedge R(x_2, x_2, x_3) \wedge$
  $R(x_1, x_1, x_1) \wedge (x_1 = x_4)\ \wedge R(x_2, x_4, y_2) \wedge R(x_4, x_4, x_2)$
- Get rid of $=$
  $(\exists x_1, x_2, x_3,\ y_1, y_2)\ \ R(x_3, x_1, y_1) \wedge R(x_2, x_2, x_3) \wedge$
  $R(x_1, x_1, x_1)\ \wedge R(x_2, x_1, y_2) \wedge R(x_1, x_1, x_2)$
- The new instance has a solution iff the original one does

pp-definitions are not satisfactory

- 3-COLOR does not pp-define 3-SAT: different domains
- 3-SAT does not pp-define 3-COLOR: even worse, the domain is larger
- solution:
  - each variable of a 3-COLOR instance is encoded as a pair of variables in a Boolean instance
  - a (binary) constraint is encoded as a 4-ary constraint

Informal definition: $\mathcal{A}$ pp-interprets $\mathcal{B}$ if

- the domain of B is a pp-definable relation (from $\mathcal{A}$) modulo a pp-definable equivalence
- the relations in $\mathcal{B}$ (regarded as relations on A) are also pp-definable

pp-interpretations

Definition

We say that $\mathcal{A}$ pp-interprets $\mathcal{B}$ if
$\exists n \in \mathbb{N}$, $\exists C \subseteq A^n$, $\exists f : C \to B$ onto, such that $\mathcal{A}$ pp-defines

- C, the kernel of f (regarded as a $2n$-ary relation on A), and
- the f-preimage of every relation in $\mathcal{B}$ (f-preimage of a k-ary relation is regarded as a nk-ary relation on A)

Example: $\mathcal{A} = (\{0,1\}; \dots)$ 3-SAT, $\mathcal{B} = (\{1,2,3\}, \neq)$ 3-COLOR
$n = 2$, $C = \{(0,1),(1,0),(1,1)\}$,
$f : (0,1) \mapsto 1, (1,0) \mapsto 2, (1,1) \mapsto 3$

- $\mathcal{A}$ pp-defines C and the kernel of f
- f-preimage of $\neq$ is
  $\{f^{-1}(1,2), f^{-1}(1,3), \dots\} =$
  $\{((0,1),(1,0)), ((0,1),(1,1)), \dots$
  regarded as a 4-ary relation: $\{(0,1,1,0),(0,1,1,1),\dots\}$
  is pp-definable from $\mathcal{A}$.

pp-interpretations give reductions

Theorem

If $\mathcal{A}$ pp-interprets $\mathcal{B}$, then $\mathrm{CSP}(\mathcal{B}) \leq \mathrm{CSP}(\mathcal{A})$.

Remarks

- Proof is easy – idea was mentioned
- It seems that finding pp-definitions requires creativity (we will see that it doesn't)
- Does not easily show that 3-SAT $\leq$ NAE-SAT (further reductions will show this easily)

Homomorphic equivalence, reduction to cores

Definition

$\mathcal{A}$ and $\mathcal{B}$ of the same signature are homomorphically equivalent if there exist homorphisms $\mathcal{A} \to \mathcal{B}$ and $\mathcal{B} \to \mathcal{A}$.

Theorem

If $\mathcal{A}$ and $\mathcal{B}$ are homomorphically equivalent, then $\mathrm{CSP}(\mathcal{A}) = \mathrm{CSP}(\mathcal{B})$

Theorem

Each $\mathcal{A}$ is homomorphically equivalent to a unique core, ie. a structure whose each endomorphism is a bijection

Example: If $\exists c \in A$ such that each relation contains a constant tuple $(c, \ldots, c)$, then the core of $\mathcal{A}$ is a singleton structure, and $\mathrm{CSP}(\mathcal{A})$ is VERY easy

Reduction to idempotent cores

Theorem

Let $\mathcal{A} = \{(a_1, \ldots, a_n); \ldots\}$ *be a core.*
Let $\mathcal{B}$ *be the structure obtained from* $\mathcal{A}$ *by adding* $C_{a_1}, \ldots, C_{a_n}$.
Then $\mathrm{CSP}(\mathcal{B}) \leq \mathrm{CSP}(\mathcal{A})$.

- **Crucial!** The set of endomorphisms of $\mathcal{A}$ regarded as an n-ary relation, ie.
  $S = \{(f(a_1), f(a_2), \ldots, f(a_n)) : f \in \mathrm{End}\,\mathcal{A} = \mathrm{Aut}\,\mathcal{A}\}$
  is pp-definable from $\mathcal{A}$ (without $\exists$):
  $S(x_1, \ldots, x_n)$ iff $\bigwedge_{R \text{ in } \mathcal{A}} \bigwedge_{(b_1,\ldots,b_k) \in R} R(x_{b_1}, \ldots, x_{b_k})$
- Consider an instance of $\mathrm{CSP}(\mathcal{B})$
- Introduce new variables $x_{a_1}, \ldots, x_{a_n}$
- Add the constraint $S(x_{a_1}, \ldots, x_{a_n})$
- Replace each $C_a(x)$ by $x = x_a$
- The new instance has a solution iff the orginal does:
  - $\Rightarrow$ use inverse of the automorphism determined by values of $x_{a_1}, \ldots, x_{a_n}$

Remarks

- **Exercise**: 3-SAT $\leq$ 3-COLOR: pp-construct 3-SAT from 3-COLOR + singletons

- **Def**: idempotent core ...contains all singleton unary relations
- we can WLOG concentrate on idempotent cores

- **Corollary**: If $\mathrm{CSP}(\mathcal{A})$ in P, then finding a solution is in P.

Reductions - recap

In the following situations, $\mathrm{CSP}(\mathcal{B}) \leq \mathrm{CSP}(\mathcal{A})$:

- $\mathcal{A}$ pp-interprets $\mathcal{B}$
- $\mathcal{A}$ is homomorphically equivalent to $\mathcal{B}$
- $\mathcal{A}$ is a core and $\mathcal{B}$ is obtained by adding singletons

Definition

We say that $\mathcal{A}$ pp-constructs $\mathcal{B}$ if $\mathcal{B}$ can be obtained from $\mathcal{A}$ by (repeated) application of the three constructions above.

So: $\mathcal{A}$ pp-constructs $\mathcal{B} \Rightarrow \mathrm{CSP}(\mathcal{B}) \leq \mathrm{CSP}(\mathcal{A})$

The tractability conjecture

Fun fact: Each known (template of an) NP-complete CSP pp-constructs all structures!

Corollary

If $\mathcal{A}$ pp-constructs all structures (equivalently 3-SAT), then $\mathrm{CSP}(\mathcal{A})$ is NP-complete

Conjecture (The algebraic dichotomy conjecture The tractability conjecture)

Otherwise $\mathcal{A}$ is in P.

Similar conjectures for the complexity classes L, NL.

Algebra

Polymorphism

n-ary operation on A = mapping $A^n \to A$

Definition

An operation $f : A^n \to A$ is compatible with relation $R \subseteq A^k$ if whenever (a_{ij}) is a $n \times k$ matrix whose all rows are in R then f applied to the columns gives a k-tuple from R

Polymorphism of $\mathcal{A}$ = operation compatible with all relations in $\mathcal{A}$

$\operatorname{Pol} \mathcal{A}$ = the set of all polymorphisms of $\mathcal{A}$

- Polymorphism of $\mathcal{A}$ = homomorphism $\mathcal{A}^n \to \mathcal{A}$
- Note: unary polymorphism = endomorphism
- Think: symmetry of higher arity

Polymorphisms – examples, exercises

- $min : \{0,1\}^2 \to \{0,1\}$ is a polymorphism of HORN-3-SAT, max is not
- the majority operation $major : \{0,1\}^3 \to \{0,1\}$, ie $major(x,x,y) \approx major(x,y,x) \approx major(y,x,x) \approx x$ is a polymorphism of 2-SAT
- the minority operation $minor : \{0,1\}^3 \to \{0,1\}$, ie $minor(x,y,z) = x - y + z \pmod 2$ is a polymorpism of $\mathbb{Z}_2$-LIN
- A constant operation $all \mapsto c$ (of any arity) is in $\operatorname{Pol} \mathcal{A}$ iff each relation in $\mathcal{A}$ contains a constant tuple $(c, c, \dots, c)$
- f is compatible with all singleton unary relations iff f is idempotent (i.e. $f(x, \dots, x) \approx x$)
- Each projection π_i^n is a polymorphism of every structure
- Pol 3-SAT $=$ projections

Pol($\mathcal{A}$) is a clone

Pol($\mathcal{A}$):

- contains all projections
- is closed under composition, for instance,
  if $f, g \in$ Pol($\mathcal{A}$) (arity 2, 3),
  then h (arity 4) defined by
  $h(x_1, x_2, x_3, x_4) = g(x_1, f(x_3, g(x_2, x_2, x_4)), x_3)$
  is in Pol($\mathcal{A}$)

Definition

A *(function) clone* on A is a set of operations on A which contains all projections and is closed under composition.

compare: transformation monoid

Polymoprhism clones control pp-definitions

Theorem

Let $\mathcal{A}, \mathcal{B}$ have the same domain. Then $\mathcal{A}$ pp-defines $\mathcal{B}$ iff $\mathrm{Pol}(\mathcal{A}) \subseteq \mathrm{Pol}(\mathcal{B})$.

For $\Leftarrow$ enough to show: if a relation $R \subseteq A^k$ is compatible with each $f \in \mathrm{Pol}(\mathcal{A})$, then $\mathcal{A}$ pp-defines R.

- $A = \{a_1, \dots, a_n\}$, $R = \{(c_{11}, \dots, c_{1k}), \dots, (c_{m1}, \dots, c_{mk})\}$.
- **Crucial:** The set of m-ary polymorphisms of $\mathcal{A}$ regarded as an $|A^m|$-ary relation, ie. $S = \{(f(a_1, a_1, \dots, a_1), \dots, f(a_n, a_n, \dots, a_n)) : f \in \mathrm{Pol}\,\mathcal{A}\}$ is pp-definable from $\mathcal{A}$ (without $\exists$).
- existentially quantify over all coordinates but those corresponding to $(c_{11}, \dots, c_{m1})$, $\dots$, $(c_{1k}, \dots, c_{mk})$
- the obtained relation contains R (because of projections) and is contained in R (because of compatibility)

Example

- Proof gives pp-definitions whenever they exist
- **Example**: 3-SAT $\leq$ 1-in-3-SAT now requires no creativity
- pp-definition of R_{ijk} from $\{(0,0,1),(0,1,0),(1,0,0)\}$ according to the proof:
  - R_{ijk} has 7-triples
  - 7-ary polymorphisms (of 1-in-3-SAT) form a 2^7-ary relation
  - its pp-definition will have $2^7 = 128$ variables and $3^7 = 2187$ clauses
  - we existentially quantify 121 variables

Boolean CSPs – Schaefer's theorem

- Take $\mathcal{A} = (\{0,1\};\ \dots)$
- If a constant is in $\operatorname{Pol}\mathcal{A}$, then $\mathrm{CSP}(\mathcal{A})$ is in P (answer YES)
- otherwise $\mathcal{A}$ is a core – we can add singletons without changing the complexity
- So, assume $\mathcal{A}$ is an idempotent core (contains C_0, C_1)
- Thus $\operatorname{Pol}\mathcal{A}$ is idempotent
- If $\operatorname{Pol}\mathcal{A}$ contains only projections, then $\mathcal{A}$ pp-interprets everything, therefore $\mathrm{CSP}(\mathcal{A})$ is NP-complete
- Now assume $\operatorname{Pol}\mathcal{A}$ is nontrivial (contains a non-projection).
- We will show that $\mathrm{CSP}(\mathcal{A})$ is in P.

Boolean CSPs II

Fact

Each nontrivial idempotent clone on $\{0, 1\}$ *contains* *max*, *min*, *major*, *or minor*.

Possible proofs:

- All clones on $\{0, 1\}$ are described – look at the list
- Direct elementary proof

Boolean CSPs III

Cases:

- *minor* $\in \mathrm{Pol}(\mathcal{A})$
  - **Exercise:** each relation compatible with *minor* is an affine subspace of $\mathbb{Z}_2^n$
  - Thus $\mathrm{CSP}(\mathcal{A})$ can be solved by Gaussian elimination
- *major* $\in \mathrm{Pol}(\mathcal{A})$
  - **Exercise:** each relation compatible with *major* is determined by its binary projections, therefore is pp-definable from binary relations
  - Thus $\mathrm{CSP}(\mathcal{A}) \leq \mathrm{CSP}(\{0,1\};$ all binaries$) \leq$ 2-SAT
- *min* $\in \mathrm{Pol}(\mathcal{A})$
  - **Exercise (hardest):** each relation compatible with *min* is pp-definable from HORN-3-SAT
  - Thus $\mathrm{CSP}(\mathcal{A}) \leq$ HORN-3-SAT
- *max* $\in \mathrm{Pol}(\mathcal{A})$ is dual

Boolean CSPs IV - remarks

- The polynomial solvability of *minor*, *major*, *min*, *max* follows from general results (later)
- We only used algebraic counterpart to pp-definitions (no pp-interpretations), because the domain is small

Now we continue with algebra

pp-interpretations and algebraic constructions

Basic constructions with algebras: forming subalgebras, finite powers, quotients, expansions

can be performed with clones: restricting to invariant subsets, forming finite powers, quotients, expansions

Theorem

TFAE

(i) $\mathcal{A}$ *pp-interprets* $\mathcal{B}$

(ii) $\operatorname{Pol}\mathcal{B}$ *can be obtained from* $\operatorname{Pol}\mathcal{A}$ *using these basic constructions*

Basic constructions and clone homomorphisms

Definition

A mapping $\mathrm{Pol}\,\mathcal{A} \to \mathrm{Pol}\,\mathcal{B}$ is a clone homomorphism if it preserves

- arities
- projections
- composition

- Does not depend on the concrete operations in the clones, depends only on the way how they compose
- An arity preserving mapping is a clone homomorphism iff it preserves identities

  eg. associative binary operation is mapped to an associative operation

  a majority operation is mapped to a majority operation

Birkhoff's theorem

Theorem

TFAE

(i) $\operatorname{Pol}\mathcal{B}$ *can be obtained from* $\operatorname{Pol}\mathcal{A}$ *using the basic constructions*

(ii) *There exists a clone homomorphism* $\operatorname{Pol}\mathcal{A} \to \operatorname{Pol}\mathcal{B}$

Proof: the crucial object is the same as before!

Identities!

Corollary

TFAE

(i) $\mathcal{A}$ *pp-interprets* $\mathcal{B}$

(ii) $\mathrm{Pol}\,\mathcal{B}$ *can be obtained from* $\mathrm{Pol}\,\mathcal{A}$ *using the basic constructions*

(iii) *There exists a clone homomorphism* $\mathrm{Pol}\,\mathcal{A} \to \mathrm{Pol}\,\mathcal{B}$

If this is the case, then $\mathrm{CSP}(\mathcal{B}) \leq \mathrm{CSP}(\mathcal{A})$.

The complexity of $\mathrm{CSP}(\mathcal{A})$ depends only on identities satisfied by polymorphisms of $\mathcal{A}$.

Universal algebra serves in 2 ways:

- toolbox containing heavy hammers
- catalog of important identities $\to$ guideline to identifying interesting intermediate cases and tools to attack them

The algebraic dichotomy conjecture again

Conjecture

Let $\mathcal{A}$ be a core. Then $\mathrm{CSP}(\mathcal{A})$ is in P if (equivalently):

(i) $\mathcal{A}$ does not pp-interpret everything

(ii) the trivial clone cannot be obtained from $\operatorname{Pol}\mathcal{A}$ by the basic constructions

(iii) there does not exist a clone homomorphism from $\operatorname{Pol}\mathcal{A}$ to the trivial clone
ie. operations $\operatorname{Pol}\mathcal{A}$ satisfy some nontrivial identities (=not satisfiable by projections)

...

(mdxii) Siggers $\operatorname{Pol}\mathcal{A}$ contains a 4-ary operation t satisfying
$t(r, a, r, e) \approx t(a, r, e, a)$

(hchkr) Barto, Kozik $\operatorname{Pol}\mathcal{A}$ contains a p-ary operation t ($\forall p > |A|$ a prime) satisfying
$t(x_1, \ldots, x_p) \approx t(x_2, \ldots, x_p, x_1)$

Hell and Nešetřil theorem

Theorem

Let $\mathcal{A} = (A; R)$, where R is binary, symmetric.
If R has no loops and is non-bipartite, then $\mathrm{CSP}(\mathcal{A})$ is NP-complete.
Otherwise $\mathrm{CSP}(\mathcal{A})$ is in P.

Proof:

- Assume $\mathcal{A}$ is non-bipartite, and a core
- If $\mathrm{CSP}(\mathcal{A})$ does not pp-interpret everything, then $\mathcal{A}$ has a cyclic polymorphism t_p of each prime arity $p > |A|$
- Find a closed walk $a_1, \dots, a_p, a_1$ for some prime $p > |A|$
- Then $(t_p(a_1, \dots, a_p), t_p(a_2, \dots, a_p, a_1)) = (c, c) \in R$ since t_p is a polymorphism

Basics – reading

- Feder, Vardi: The computational structure of monotone monadic snp and constraint satisfaction: A study through datalog and group theory
- Bulatov, Jeavons, Krokhin: Classifying the complexity of constraints using finite algebras
- Bodirsky: Constraint satisfaction problems with infinite templates
- Barto: The constraint satisfaction problem and universal algebra
- Barto, Opršal, Pinsker: The wonderland of the double shrink ***title may change***

Results

Results

- Better understanding of pre-algebraic results
- Far broader special cases solved. The dichotomy conjecture is true:
  - if $|A| = 3$ Bulatov'06
  - if $|A| = 4$ Marković et al.
  - if $\mathcal{A}$ contains all unary relations Bulatov'03, Barto'11
  - if $\mathcal{A} = (A; R)$ where R is binary, without sources or sinks Barto, Kozik, Niven'09
- Applicability of known algorithmic principles understood
  - Describing all solutions – "few subpowers" 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
- Progress on finer complexity classification

Local consistency

Roughly: $\mathcal{A}$ has bounded width iff $\mathrm{CSP}(\mathcal{A})$ 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 $\mathrm{CSP}(\mathcal{A})$ we say that $\mathcal{A}$ has width (k, l).
- if $\mathcal{A}$ has width (k, l) for some k, l then $\mathcal{A}$ has bounded width

Various equivalent formulations (bounded tree width duality, Datalog, LFP logic, games)

Example of $(2,3)$-consistency

Let $\mathcal{A} = (\{0,1\}; \neq)$ (2-COLOR)

Consider the instance

$$x_1 \neq x_2, x_2 \neq x_3, x_3 \neq x_4, x_4 \neq x_5, x_5 \neq x_1$$

- By looking at $\{x_1, x_2, x_3\}$ we see (using $x_1 \neq x_2$ and $x_2 \neq x_3$) that $x_1 = x_3$.
- By looking at $\{x_1, x_3, x_4\}$ we see (using $x_1 = x_3$ and $x_3 \neq x_4$) that $x_1 \neq x_4$.
- By looking at $\{x_1, x_4, x_5\}$ we now see (using $x_1 \neq x_4$, $x_4 \neq x_5$, $x_5 \neq x_1$) a contradiction

In fact, $\mathcal{A}$ has width $(2,3)$, that is, such reasoning is always sufficient for an instance of $\mathrm{CSP}(\mathcal{A})$.

Bounded width

- The problems $\mathbb{Z}_p$-*LIN* do not have bounded width Feder, Vardi'93
- If $\mathcal{A}$ pp-constructs $\mathbb{Z}_p$-LIN then $\mathcal{A}$ does not have bounded width Larose, Zádori'07
- Thus the "obvious" necessary condition for bounded width is that $\mathcal{A}$ does not pp-construct $\mathbb{Z}_p$-LIN.
- It is sufficient:

Theorem

The following are equivalent.

- *$\mathcal{A}$ does not pp-construct $\mathbb{Z}_p$-LIN*
- $\text{Pol}(\mathcal{A})$ *contains operations satisfying ....*
- *$\mathcal{A}$ has bounded width B, Kozik'09*
- *$\mathcal{A}$ has width $(2,3)$ B; Bulatov*

Towards the bounded width theorem

- First universal algebraic steps Feder and Vardi:
  - If Pol $\mathcal{A}$ contains TSI polymorphisms of all arities, then $\mathrm{CSP}(\mathcal{A})$ has width 1

    covers HORN-SAT
  - If Pol $\mathcal{A}$ has a majority polymorphism, then $\mathrm{CSP}(\mathcal{A})$ has width $(2, 3)$

    covers 2-SAT
  - More generally: if Pol $\mathcal{A}$ has an NU polymorphism, then $\mathrm{CSP}(\mathcal{A})$ has bounded width
- "$\mathcal{A}$ does not pp-construct $\mathbb{Z}_p$-LIN" is a well-known algebraic condition on Pol $\mathcal{A}$
- UA suggested more general intermediate steps (and gave tools)
  - 2-semilattices Bulatov
  - CD(3) Kiss, Valeriote, CD(4) Carvalho, Dalmau, Marković, Maróti, CD Barto, Kozik

TSI

TSI = operation whose value depends only on the set of its arguments:
$\{a_1, \dots, a_n\} = \{b_1, \dots, b_n\} \Rightarrow t(a_1, \dots, a_n) = t(b_1, \dots, b_n)$

- Assume $\mathcal{A}$ has TSI polymorphisms of all arities
- We will show that $\mathrm{CSP}(\mathcal{A})$ has width 1.
- $(1,1)$-algorithm more precisely:
  - For each variable x, set $P_x := A$ (meaning: possible values)
  - If $a \in P_x$, $R(x, y, z, \dots)$ is a constraint, and no tuple of the form $(a, b \in P_y, c \in P_z, \dots)$ is in R,
    then remove a from P_x
  - Repeat until no removals are made
  - If $(\exists x)$ $P_x = \emptyset$ for some x, return NO SOLUTION
- Need to show: If $(\forall x)$ $P_x \neq \emptyset$, then there is a solution
- Choose TSI polymorphism of sufficiently big arity
- Apply it to P_x: we get $a_x \in P_x$
- $x \mapsto a_x$ is a solution!

Describing all solutions – few subpowers

- In $\mathbf{Z}_p$-LIN we can "describe" all solutions – we can find polynomially large (wrt # of variables) set of solutions (called generating set) so that the solution set is its affine hull
- **Def**: Let R be pp-definable from $\mathcal{A}$. $X \subseteq R$ is a generating set of R if R is equal to the closure of R under Pol($\mathcal{A}$).
- Sequence of papers generalizing the algorithm for $\mathbf{Z}_p$ Feder, Vardi; Bulatov; Bulatov, Dalmau; Dalmau culminated in

Theorem (Berman et al, Idziak et al.)

TFAE for an idempotent core $\mathcal{A}$

- *$\mathcal{A}$ has at most $2^{poly(n)}$ pp-definable relations of arity n*
- *Each n-ary pp-definable relation has a generating set of size $poly(n)$.*
- Pol($\mathcal{A}$) *contains operations satisfying ......*

In this case, $\mathrm{CSP}(\mathcal{A})$ is in P; moreover, generating set of solutions can be found in P.

Bonuses

Bonus I: Other connectives

$CSP(\mathcal{A})$:
Instance: Sentence ϕ in the language of $\mathcal{A}$ with $\exists$ and $\wedge$
Question: Is ϕ true in $\mathcal{A}$?

What about: Allow some other combination of $\{\exists, \forall, \wedge, \vee, \neg, =, \neq\}$.

From 2^7 cases only 3 interesting (others reduce to these or are boring)

- $\{\exists, \wedge, (=)\}$ (CSP)
  open
- $\{\exists, \forall, \wedge, (=)\}$ (qCSP)
  open
- $\{\exists, \forall, \wedge, \vee\}$ (Positive equality free)
  solved - tetrachotomy P, NP-c, co-NP-c, PSPACE-c
  B.Martin, F.Madelaine 11

Bonus II: Robust approximation

- **Task:** Find an almost satisfying assignment given an almost satisfiable instance
- More precisely: Find an assignment satisfying at least $(1 - g(\varepsilon))$ fraction of the constraints given an instance which is $(1 - \varepsilon)$ satisfiable, where $g(\varepsilon) \to 0$ as $\varepsilon \to 0$ (g should only depend on $\mathcal{A}$).
- Algorithms for 2-SAT and HORN-SAT based on linear programming and semidefinite programming Zwick'98
- $\mathbb{Z}_p$-LIN has no robust polynomial algorithm (assuming P $\neq$ NP) Hastad'01
- If $\mathcal{A}$ pp-constructs $\mathbb{Z}_p$-LIN then $\mathrm{CSP}(\mathcal{A})$ has no robust algorithm Dalmau, Krokhin'11

Bonus II: Robust approximation 2

- If $\mathcal{A}$ pp-constructs $\mathbb{Z}_p$-LIN then $\mathrm{CSP}(\mathcal{A})$ has no robust algorithm Dalmau, Krokhin'11
- Conjecture of Guruswami and Zhou: this is the only obstacle

Theorem (B, Kozik'12)

The following are equivalent (assuming P $\neq$ NP)

- $\mathcal{A}$ *does not pp-construct* $\mathbb{Z}_p$*-LIN*
- $\mathrm{CSP}(\mathcal{A})$ *has a robust polynomial algorithm*
- *canonical semidefinite programming relaxation correctly decides* $\mathrm{CSP}(\mathcal{A})$

Bonus III: Counting CSP

- The complexity is also controlled by Pol($\mathcal{A}$)
- A necessary condition for tractability found
  Bulatov, Dalmau'03
  (inspiration: the other algorithm for decison CSPs)
- A stronger necessary condition for tractability found
  Bulatov, Grohe'05
- The stronger condition is sufficient
  Bulatov'08, Dyer and Richerby'10

Bonus IV: Valued CSP

- Weighted relation: mapping $A^n \to \mathbb{Q} \cup \{\infty\}$
- **Instance**: sum, eg $R(x_1, x_2) + S(x_3, x_1, x_2)$
- **Task**: Minimize the sum
- **Includes:** Satisfiability, optimization
- Algebraic theory Cohen, Cooper, Creed, Jeavons, Živný
- **Classification modulo the algebraic dichotomy conjecture!** Kolmogorov, Krokhin, Rolinek
- Algorithm: alg. for satisfiability + linear programming

Wrap up

Final remarks

Satisfiability problem

- Easy criterion for hardness
- Complexity depends on indentities
- Theory gives generic reduction between any two NP-complete CSPs (instead of ad hoc reductions)
- Applicability of known algorithms understood
- The dichotomy conjecture still open in general

For other variants (Approx-CSP, Valued CSP, infinite)

- Universal algebra also relevant Cohen, Cooper, Creed, Jeavons, Živný; Raghavendra; Bodirsky, Pinsker
- More or less the same criterion for easiness/hardness
- Easiness comes from "symmetry"
- One needs symmetry of higher arity (e.g. polymorphisms) rather than just automorphisms or endomorphisms

Beyond CSPs

- ???
- There is ≥ 1 examples Raghavendra

We need lunch!

Thank you!