A theoretical look at the CSP

Jakub Bulín

KTIML MFF UK

Jan 7, 2020

A brief history

- Operations research
- Database theory
- Artificial intelligence
- Computational logic
- Complexity theory
- Combinatorics
- Universal algebra, multivalued logic, category theory

Constraint Satisfaction Problem (CSP)

Input:

- X – a finite set of variables,
- A – a finite set of values,
- $\mathcal{C} = \{C_1, \ldots, C_m\}$ – finitely many constraints $C_i = (\bar{x}_i, R_i)$,
  - $\bar{x}_i$ is a k_i-tuple of variables ("constraint scope")
  - $R_i \subseteq A^{k_i}$ ("constraint relation")

Decide: Is there a solution, i.e. an evaluation $\varphi : X \to A$ satisfying $\varphi(\bar{x}_i) \in R_i$ for all $1 \leq i \leq m$?

Example

- $X = \{x, y, z\}$, $A = \{0, 1\}$, constraints $\mathcal{C} = \{C_1, C_2, C_3\}$
- $C_1 = ((x, y), R)$, $C_2 = ((y, z), R)$, $C_3 = ((z, x), R)$, where $R = \{(0, 1), (1, 0)\}$

Logical viewpoint

A (finite) relational structure: $\mathbf{A} = \langle A; R_1^{\mathbf{A}}, \dots, R_n^{\mathbf{A}} \rangle$ where $R_i^{\mathbf{A}} \subseteq A^{k_i}$ is a k_i-ary relation on the set A

"Primitive-positive" fragment of FO model checking

- **Input:** a $\{\exists, \wedge, =\}$-sentence Φ and a finite relational structure $\mathbf{A}$ (in the same language)
- **Decide:** Does $\mathbf{A} \models \Phi$, i.e., is Φ true in $\mathbf{A}$?

Construction: constraint $C = (\bar{x}, R)$ becomes a predicate $R(\bar{x})$, make a conjunction, quantify everything existentially

Example

- $\Phi = (\exists x)(\exists y)(\exists z)(\, R(x,y) \wedge R(y,z) \wedge R(z,x)\,)$
- $\mathbf{A} = \langle \{0,1\}; R^{\mathbf{A}} \rangle$ where $R^{\mathbf{A}} = \{(0,1),(1,0)\}$

[*k*-]SAT

- **Input:** a propositional formula ψ in [*k*-]CNF
- **Decide:** Is ψ satisfiable?

Fact: SAT is equivalent to 3-SAT

e.g. $x_1 \vee x_2 \vee x_3 \vee \neg x_4 \quad \rightsquigarrow \quad (x_1 \vee x_2 \vee \neg u) \wedge (u \vee x_3 \vee \neg x_4) \quad (u \text{ new})$

3-SAT as a CSP

$\mathbf{A} = \langle \{0,1\}; \{R^{\mathbf{A}}_{ijk} \mid i,j,k \in \{0,1\}\}\rangle \qquad R^{\mathbf{A}}_{ijk} = \{0,1\}^3 \setminus \{(i,j,k)\}$

Example

3-SAT input

$$\psi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_4 \vee x_5 \vee \neg x_1) \wedge (\neg x_1 \vee x_4 \vee \neg x_3)$$

becomes

$$\Phi = (\exists x_1 \ldots x_5)(R_{010}(x_1, x_2, x_3) \wedge R_{101}(x_4, x_5, x_1) \wedge R_{101}(x_1, x_4, x_3))$$

2-SAT

- **Input:** a propositional formula ψ in 2-CNF
  e.g. $\psi = (x \vee \neg y) \wedge (y \vee \neg z) \wedge (\neg x \vee z)$
- **Decide:** Is ψ satisfiable?
- CSP: $\mathbf{A} = \langle \{0,1\}; R_{11}^{\mathbf{A}}, R_{10}^{\mathbf{A}}, R_{01}^{\mathbf{A}}, R_{00}^{\mathbf{A}} \rangle \quad R_{ij}^{\mathbf{A}} = \{0,1\}^2 \setminus \{(i,j)\}$
  $\Phi = (\exists x, y, z)(\, R_{01}(x,y), R_{01}(y,z), R_{10}(x,z)\,)$

Horn-[3-]SAT

- **Input:** a conjunction of Horn clauses [of width 3]
- enough to encode "$x \wedge y \to z$", "$x \wedge y \to \neg z$", "$\neg x$", "x"
- CSP: $\mathbf{A} = \langle \{0,1\}; R_{110}^{\mathbf{A}}, R_{111}^{\mathbf{A}}, C_0^{\mathbf{A}}, C_1^{\mathbf{A}} \rangle \quad C_0^{\mathbf{A}} = \{0\}, C_1^{\mathbf{A}} = \{1\}$
  $\psi = (x \wedge y \to z) \wedge (y \wedge z \to \neg x) \wedge x \wedge \neg z$
  $\Phi = (\exists x, y, z)(\, R_{110}(x,y,z) \wedge R_{111}(y,z,x) \wedge C_1(x) \wedge C_0(z)\,)$

- A digraph (directed graph): $\mathbf{G} = \langle G; \to^{\mathbf{G}} \rangle$ where $\to^{\mathbf{G}} \subseteq G \times G$
- A (simple) graph: $\to^{\mathbf{G}}$ is symmetric and loopless
- Graph homomorphism: $\varphi : \mathbf{G} \to \mathbf{H}$ such that for every edge $u \to v$ in $\mathbf{G}$ we have $\varphi(u) \to \varphi(v)$ in $\mathbf{H}$, i.e.

$$(u, v) \in \to^{\mathbf{G}} \quad \Longrightarrow \quad (\varphi(u), \varphi(v)) \in \to^{\mathbf{H}}$$

- Relational homomorphism: $\varphi : \mathbf{A} \to \mathbf{B}$ preserving relations, i.e. for every R (say k-ary) in the language we have

$$(a_1, \ldots, a_k) \in R^{\mathbf{A}} \quad \Longrightarrow \quad (\varphi(a_1), \ldots, \varphi(a_k)) \in R^{\mathbf{B}}$$

CSP as a homomorphism problem

Homomorphism problem

- **Input:** a pair of finite relational structures $\mathbf{X}, \mathbf{A}$
- **Decide:** Is there a homomorphism $\varphi : \mathbf{X} \to \mathbf{A}$?

Example (from slide 3)

- $X = \{x, y, z\}$, $A = \{0, 1\}$, $\mathcal{C} = \{C_1, C_2, C_3\}$, $C_1 = ((x, y), R)$, $C_2 = ((y, z), R)$, $C_3 = ((z, x), R)$, where $R = \{(0, 1), (1, 0)\}$
- construction of $\mathbf{A}$ and $\mathbf{X}$:
  - $R^{\mathbf{A}}$'s are all distinct relations on A appearing as constraint relations in the CSP instance
  - collect to $R^{\mathbf{X}}$ all tuples of variables that are constraint scopes with constraint relation $R^{\mathbf{A}}$
- $\mathbf{X} = \langle\{x, y, z\}; R^{\mathbf{X}}\rangle$ where $R^{\mathbf{X}} = \{(x, y), (y, z), (z, x)\}$,
  $\mathbf{A} = \langle\{0, 1\}; R^{\mathbf{A}}\rangle$ where $R^{\mathbf{A}} = \{(0, 1), (1, 0)\}$
  ("Is the oriented 3-cycle 2-colorable?")

Graph homomorphism & coloring problems

Graph homomorphism

- **Input:** a pair of (simple) graphs $\mathbf{G}, \mathbf{H}$
- **Decide:** Is there a graph homomorphism $\varphi : \mathbf{G} \to \mathbf{H}$?

Note that every CSP can be encoded as a digraph homomorphism problem, but not (simple) graph homomorphism.

Graph coloring

- **Input:** a graph $\mathbf{G}$ and $c > 0$
- **Decide:** Is $\mathbf{G}$ colorable with c colors?

(A special case of graph homomorphism where $\mathbf{H} = \mathbf{K_c}$.)

Recap

- Every CSP instance can be equivalently viewed as
  - validity of a primitive positive $(\exists, \wedge, =)$ sentence in a finite relational structure,
  - the homomorphism problem for a pair of structures.
- Different viewpoints sometimes bring better insight and tools.
- Many classical computational problems are CSPs.

Computational complexity: P vs. NP

- Decision problem: for every instance answer YES or NO
- A problem is in P: "can be solved efficiently" — polynomial-time algorithm (linear, $n \log n$, quadratic,...)
- NP problem: "correctness of a given solution can be verified efficiently" — an oracle provides an answer with proof, we can verify by a polynomial-time algorithm
- Reduction [polynomial-time]: transform [in polynomial time] instances of one problem to instances of another problem, preserving the answer
- NP-complete problem: is in NP and every NP problem reduces to it in polynomial time
  - e.g. 3-SAT, graph 3-coloring
  - known algorithms are exponential-time (worst-case complexity)
  - The P vs. NP problem: P algorithm for NP-complete problems?

Complexity classification of CSPs?

Fact: CSP is $\texttt{NP}$-complete

- **In $\texttt{NP}$:** to verify if $\varphi : X \to A$ is a solution, check for every constraint $C = (\bar{x}, R)$ whether $\varphi(\bar{x}) \in R$
- $\texttt{NP}$-**complete:** contains (has a reduction from) 3-SAT

Easier subproblems? Restrict possible CSP inputs $(\mathbf{X}, \mathbf{A})$:

- if $\mathbf{X}$ is fixed, then $\mathrm{CSP}(\mathbf{X}, -)$ is solvable in polynomial time
- if $\mathbf{X}$'s are (relational) trees, then the CSP is in $\texttt{P}$
- also true if $\mathbf{X}$'s have treewidth k – dynamic programming ("looks like a tree from far away")

Theorem (Grohe 2007)

$\mathrm{CSP}(\mathcal{C}, -)$ *is in* $\texttt{P}$*, if and only if* $\mathcal{C}$ *is a class of structures with bounded treewidth.*[1]

---

[1] up to "hom. equivalence", under reasonable complexity theory assumptions

Fixing the template

- It is natural to restrict admissible *constraint relations*.
- Combinatorial view: fix the structure **A** ("template")
- Database theory: evaluate varying input queries **X** over a fixed database **A**.

$\mathrm{CSP}(\mathbf{A})$

- **Input:** a relational structure **X**
- **Decide:** Is there a homomorphism $\varphi : \mathbf{X} \to \mathbf{A}$?

Examples

- graph 3-coloring is $\mathrm{CSP}(\mathbf{K_3})$
- 3-SAT, 2-SAT, Horn-SAT are of this form too

"What properties of **A** *make* $\mathrm{CSP}(\mathbf{A})$ *easy vs. hard?"*

A polymorphism of $\mathbf{A}$:

- a function $f : A^n \to A$ preserving all the constraint relations, i.e. for each $R^{\mathbf{A}}$ and $\mathbf{a^i} \in R^{\mathbf{A}}$, $f(\mathbf{a^1}, \ldots, \mathbf{a^n}) \in R^{\mathbf{A}}$

$$\begin{array}{ccccccc} f(a_1 & a_2 & \ldots & a_n) & = & a \\ \downarrow & \downarrow & & \downarrow & \Longrightarrow & \downarrow \\ f(b_1 & b_2 & \ldots & b_n) & = & b \end{array}$$

- a multivariate homomorphism $f : \mathbf{A}^n \to \mathbf{A}$
- a "high-dimensional symmetry" of solution spaces of $\mathrm{CSP}(\mathbf{A})$ instances, can be used in algorithms to combine [partial] solutions to obtain "nicer" solutions
- Pol($\mathbf{A}$): the set of all polymorphisms of $\mathbf{A}$, closed under composition, contains projections $f(x_1, \ldots, x_n) = x_i$

"More symmetric problems are easier."

Hard Boolean CSPs

3-SAT: $\mathbf{A} = \langle \{0,1\}; \{R^{\mathbf{A}}_{ijk} \mid i,j,k \in \{0,1\}\} \rangle$

- Pol($\mathbf{A}$): only projections

1in3-SAT: $\mathbf{A} = \langle \{0,1\}; \{(1,0,0),(0,1,0),(0,0,1)\} \rangle$

- **Input:** a list of triples of Boolean variables
- **Goal:** evaluate so that in each triple exactly 1 variable is true
- Pol($\mathbf{A}$): only projections

NAE-SAT: $\mathbf{A} = \langle \{0,1\}; \{0,1\}^3 \setminus \{(0,0,0),(1,1,1)\} \rangle$

- **Input:** a list of triples of Boolean variables
- **Goal:** evaluate so that in each triple at least 1 variable is true and at least 1 is false
- Pol($\mathbf{A}$):projections and their negations

Easy Boolean CSPs

HORN-SAT: $\mathbf{A} = \langle \{0,1\}; R^{\mathbf{A}}_{110}, R^{\mathbf{A}}_{111}, \{0\}, \{1\} \rangle$

- unit propagation algorithm (essentially arc consistency)
- Pol($\mathbf{A}$): conjunctive functions, e.g. $\min(x, y)$

2-SAT: $\mathbf{A} = \langle \{0,1\}; R^{\mathbf{A}}_{11}, R^{\mathbf{A}}_{10}, R^{\mathbf{A}}_{01}, R^{\mathbf{A}}_{00} \rangle$

- propagate values via edges in search of a failure
- Pol($\mathbf{A}$): monotone functions, e.g. $\mathrm{majority}(x, y, z)$

PATH (digraph [un-]reachability): $\mathbf{A} = \langle \{0,1\}; x \leq y, \{0\}, \{1\} \rangle$

- given a digraph and vertices s, t, answer YES if there is no directed path from s to t
- Pol($\mathbf{A}$): same as 2-SAT, e.g. $\mathrm{majority}(x, y, z)$

UPATH (graph [un-]reachability): $\mathbf{A} = \langle \{0,1\}; x = y, \{0\}, \{1\} \rangle$

- given a (simple) graph and two vertices, YES if not connected
- Pol($\mathbf{A}$): $f(x, x, \ldots, x) = x$, e.g. $\min(x, y)$, $\mathrm{majority}(x, y, z)$

Arc Consistency (a very high-level view)

- For every variable $x \in X$ keep a list of possible values $P_x \subseteq A$
- Initialize: $P_x := A$
- Update: For every constraint $C = (\bar{x}, R)$ and every i,

$$P_{x_i} := P_{x_i} \cap \mathrm{proj}_{x_i} R$$
$$R := R \cap (P_{x_1} \times \cdots \times P_{x_n})$$

- Repeat until no change
- The instance is arc consistent, if all P_x are nonempty.
- A solution $\Rightarrow$ arc consistent. (“$\Leftarrow$” not true in general.)

Theorem

If $\mathrm{Pol}(\mathbf{A})$ *contains* $\min(x, y)$*, then every arc consistent instance of* $\mathrm{CSP}(\mathbf{A})$ *has a solution (*$\Rightarrow \mathrm{CSP}(\mathbf{A})$ *is in* $\mathtt{P}$*).*

Proof. Define $\varphi(x) := \min(\{a \in P_x\})$. [blackboard picture]

- For all $x, y \in X$ compute admissible $P_x \subseteq A$, $P_{xy} \subseteq A \times A$
- Initialize: $P_x := A$, $P_{xy} := A \times A$. Enforce the following:
  - for every $C = (\bar{x}, R)$ or $((x, y), P_{xy})$ and every $x, y \in \bar{x}$, $P_x = \mathrm{proj}_x R$, $P_{xy} = \mathrm{proj}_{xy} R$
  - for every $x, y, z \in X$ and $(a, b) \in P_{x,y}$ there is $c \in P_z$ such that $(a, c) \in P_{x,z}$ and $(b, c) \in P_{y,z}$

"Any partial solution on 2 var's extends to any 3rd variable."

- The instance is (2,3)-consistent, if all P_x are nonempty.
- A solution $\Rightarrow$ (2,3)-consistent. ("$\Leftarrow$" not true in general.)

Theorem

If Pol(**A**) *contains* $\mathrm{majority}(x, y, z)$, *then every (2,3)-consistent instance of* $\mathrm{CSP}(\mathbf{A})$ *has a solution (*$\Rightarrow$ $\mathrm{CSP}(\mathbf{A})$ *is in* P*).*

Proof? Every partial solution on 3 var's extends to any 4th var. [blackboard picture]

LINEQ($\mathbb{Z}_2$)

- **Input:** a system of linear equations Σ over $\mathbb{Z}_2$
- **Decide:** Is Σ consistent?

- **Fact:** Σ can be expressed using only $x + y = z$, $x = 0$, $x = 1$. For example, $x_1 + x_2 + x_3 = 1$ becomes

$$\begin{aligned} x_1 + x_2 &= u \\ u + x_3 &= v \\ v &= 1 \end{aligned}$$

- $\mathrm{CSP}(\mathbf{A})$ where $\mathbf{A} = \langle \{0, 1\}; x + y = z, \{0\}, \{1\} \rangle$
- Gaussian elimination (computing rank of a Boolean matrix)
- Pol($\mathbf{A}$): affine functions, e.g. $x + y + z \pmod 2$
- (Note: Local consistency is no guarantee of a solution.)

Theorem (Post 1941)

Let $\mathcal{F}$ be a set of Boolean functions closed under composition and containing all projections. Then either

0. *$\mathcal{F}$ only consists of projections or their negations,*

or $\mathcal{F}$ contains one of the following "nice" functions:

1. *a constant function (always output 0 or always output 1),*
2. $\min(x, y)$ *or* $\max(x, y)$,
3. $\mathrm{majority}(x, y, z)$,
4. $x + y + z \pmod 2$.

Corollary (Schaefer's dichotomy theorem 1978)

Every Boolean $\mathrm{CSP}(\mathbf{A})$ *is either in* P *or* NP*-complete.*

Proof of Schaefer's dichotomy theorem

0. If Pol(**A**) only consists of projections or their negations, then $\mathrm{CSP}(\mathbf{A})$ encodes NAE-SAT and thus is NP-complete.
   (see the Appendix for proof)

   Else, Pol(**A**) contains one of the "nice" functions:

1. $\mathrm{const}_0 \Rightarrow$ every (nonempty) $R^{\mathbf{A}}$ contains the tuple $(0, \dots, 0)$
   $\Rightarrow$ every instance is a YES instance

2. $\min(x, y) \Rightarrow \mathrm{CSP}(\mathbf{A})$ is solvable by arc consistency

3. $\mathrm{majority}(x, y, z) \Rightarrow \mathrm{CSP}(\mathbf{A})$ is solvable by (2,3)-consistency

4. $x + y + z \pmod 2 \Rightarrow$ every $R^{\mathbf{A}}$ is an affine subspace
   $\Rightarrow$ every CSP instance is a system of linear equations over $\mathbb{Z}_2$
   $\Rightarrow \mathrm{CSP}(\mathbf{A})$ is solvable by Gaussian elimination

Let **H** be a (simple) graph.

Graph homomorphism

- **Input:** a (simple) graph **G**
- **Decide:** Is there a graph homomorphism $\varphi : \mathbf{G} \to \mathbf{H}$?

Theorem (Hell, Nešetřil 1990)

If **H** *is bipartite, then* $\mathrm{CSP}(\mathbf{H})$ *is in* P*. Otherwise,* $\mathrm{CSP}(\mathbf{H})$ *is* NP*-complete.*

- **H** is bipartite with at least one edge $\Leftrightarrow$ homomorphically equivalent to $\mathbf{K_2}$, so $\mathrm{CSP}(\mathbf{H})$ has the same YES/NO instances as graph 2-coloring.
- non-bipartite $\Leftrightarrow$ contains a cycle of odd length
- graph 2-coloring is in P, c-coloring for $c \geq 3$ is NP-complete

The CSP dichotomy theorem

For every finite relational structure $\mathbf{A}$, $\mathrm{CSP}(\mathbf{A})$ is either in $\mathtt{P}$ or $\mathtt{NP}$-complete.

- Conjectured by Feder and Vardi in 1993
- Proved by Bulatov and Zhuk in 2017
- Classification via existence of a "nice" polymorphism
- In general, if $\mathtt{P} \neq \mathtt{NP}$, then there are infinitely many different complexity classes between (up to $\mathtt{P}$-reductions).
- CSPs are in some sense the "largest natural" class where a dichotomy is possible

Want to know more?

- A Matfyz course on basics of the theory
  - `NMAG563 Intro to complexity of the CSP`
- A (somewhat, partly) accessible survey article:
  - `Polymorphisms and how to use them`
    `(L. Barto, A. Krokhin, and R. Willard)`
- Talk to me!
  - `jakub.bulin@mff.cuni.cz`

Appendix: How polymorphisms work

- a relation $S \subseteq A^k$ is pp-definable from **A**, if it is definable with a $(\exists, \wedge, =)$-formula
- equivalently, S is the set of all solutions to some instance of $\mathrm{CSP}(\mathbf{A})$, with some "auxiliary" variables ignored
- adding S to **A** doesn't change the complexity of $\mathrm{CSP}(\mathbf{A})$
- **key lemma:** S is pp-definable, if and only if it is invariant under all polymorphisms of **A**

Corollary

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

Example

- Let $\mathrm{CSP}(\mathbf{B})$ be NAE-SAT, then $\mathrm{Pol}(\mathbf{B})$ is the set of all projections and negations of projections.
- If $\mathrm{Pol}(\mathbf{A})$ contains only projections or negations of projections, then by Corollary, $\mathrm{CSP}(\mathbf{B})$ reduces to $\mathrm{CSP}(\mathbf{A})$ which proves that $\mathrm{CSP}(\mathbf{A})$ is $\mathtt{NP}$-complete.