Proof complexity of a CSP dichotomy proof

Azza Gaysin

Department of Mathematical Logic, Faculty of Computer science and Mathematics,
University of Passau, Germany

Journées sur les Arithmétiques Faibles 2024, 09.09.2024

Content

1. Constraint satisfaction problem (CSP) over finite template and its algebraic characterization:
   - Definitions;
   - Exemples;
   - Complexity results;
   - CSP as tautologies.
2. Propositional proof systems and their correspondence with Theories of Bounded Arithmetic:
   - Definitions
   - Proof complexity of CSP
3. Proof complexity of a CSP dichotomy proof:
   - Results;
   - Open questions.

Constrain satisfaction problem (CSP)

Definition 1 (Constrain satisfaction problem).

Let Γ be a set of relations over a finite domain D, or a *constraint language*. The Constraint Satisfaction Problem (CSP) is the combinatorial decision problem such that an instance of CSP(Γ) is a list of constraints $C = \{C_1, ..., C_t\}$, where every constraint C_j is a pair $(\bar{x}_j, R_j)$ with

- $\bar{x}_j$ being a tuple of variables of length m_j, and
- R_j being an m_j-ary constraint relation from Γ.

The question is whether there exists an assignment to every variable x_i such that for each constraint C_j the image of the tuple $\bar{x}_j$ is a member of the constraint relation R_j. If such an assignment exists, we call the instance *satisfiable*.

Definition 2 (CSP as a Homomorphism problem).

Let $\mathcal{A}$ be a fixed relational structure over vocabulary R_1,..., R_n. An *instance of the constraint satisfaction problem* CSP($\mathcal{A}$) is any relational structure $\mathcal{X}$ over the same vocabulary. The question is whether there exists a homomorphism from $\mathcal{X}$ to $\mathcal{A}$. If such a homomorphism exists, we call the instance *satisfiable*. $\mathcal{A}$ is called a target structure and $\mathcal{X}$ an instance structure.

Constraint satisfaction problem: Examples

- 3-SAT, $D_{\mathsf{3SAT}} = \{0,1\}$, and $\Gamma_{\mathsf{3SAT}} = \{R_{ijk} : i,j,k \in \{0,1\}\}$ where $R_{ijk} = \{0,1\}^3 \backslash \{(i,j,k)\}$. For example, the formula,

$$\varphi = (x \vee y \vee z) \wedge (\neg x \vee \neg v \vee \neg w) \wedge (\neg x \vee v \vee \neg z) \wedge (\neg y \vee \neg z \vee w)$$

  corresponds to the following CSP instance of CSP(Γ_{3SAT}):

$$C_1 = ((x,y,z); R_{000}), C_2 = ((x,v,w); R_{111}),$$

$$C_3 = ((x,v,z); R_{101}), C_4 = ((y,z,w); R_{110}).$$

- For a fixed k, the problem k-COLORING, $D_{\mathsf{k\text{-}col}} = \{0,1,...,k-1\}$, $\Gamma_{\mathsf{k\text{-}col}}$ contains the only binary relation $\neq_k := \{(a,b) : a \neq b\}$;
- A system of linear equations over the p-element field $\mathbb{Z}_p$ where each equation contains 3 variables, $D_{\mathsf{3LIN(p)}} = \mathbb{Z}_p$, $\Gamma_{\mathsf{3LIN(p)}}$ consists of all

$$R_{abcd} = \{(x,y,z) \in \mathbb{Z}_p^3 : ax + bc + dz = d\};$$

- The problem of $\mathcal{H}$-COLORING, a homomorphism problem between two graphs. If $\mathcal{H}$ is a complete undirected graph $\mathcal{K}_n$, then the problem $\mathcal{K}_n$-COLORING reduces to n-COLORING.

Constraint satisfaction problem: Complexity

- In 1978 Schaefer proved the dichotomy result between P and NP for a problem over a binary domain that he called Generalized Satisfiability[1].
- In 1990 Hell and Nešetřil shown that $\mathcal{H}$-COLORING is in P if $\mathcal{H}$ is bipartite, and it is NP-complete otherwise[2].
- In 1998 Feder and Vardi worked on the project of finding a large subclass of NP that exhibits a dichotomy[3]. Any such dichotomy has to avoid Ladner's anti-dichotomy result[4] (If P$\neq$ NP, then there are problems in NP which are neither in P nor NP-complete), so they defined the CSP problem over fixed constraint languages.
- In 2017 Zhuk and Bulatov proved that for a finite set of relations over any finite domain D, CSP(Γ) either can be solved in polynomial time, or is NP-complete[5] [6].

[1] Thomas J. Schaefer. The complexity of satisfiability problems. In Conference Record of the Tenth Annual ACM Symposium on Theory of Computing (San Diego, Calif., 1978), pages 216226. ACM, New York, 1978.

[2] Pavol Hell and Jaroslav Nešetřil. On the complexity of H-coloring. J. Combin. Theory Ser. B, 48(1):92110, 1990.

[3] Tom ́as Feder and Moshe Y Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory. SIAM Journal on Computing, 28(1):57104, 1998.

[4] Richard E Ladner. On the structure of polynomial time reducibility. Journal of the ACM (JACM), 22(1):155171, 1975.

[5] D. Zhuk, A proof of the csp dichotomy conjecture, J. ACM, 67(5),August 2020

[6] A. A. Bulatov, A dichotomy theorem for nonuniform CSPs. In 2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS), pages 319–330, 2017

Constraint satisfaction problem: Complexity

Definition 3 (Polymorphism).

We say that an m-ary operation $f : A^m \to A$ *preservers* an n-ary relation $R \in A^n$ (or f is a *polymorphism* of R) if for every choice of k m-tuples in R, applying f component-wise produces a new m-tuple which is also in R.

$$\begin{bmatrix} x_{11} \\ \vdots \\ x_{1m} \end{bmatrix}, ..., \begin{bmatrix} x_{k1} \\ \vdots \\ x_{km} \end{bmatrix} \in R \implies f\left(\begin{bmatrix} x_{11} \\ \vdots \\ x_{1m} \end{bmatrix}, ..., \begin{bmatrix} x_{k1} \\ \vdots \\ x_{km} \end{bmatrix}\right) = \begin{bmatrix} f(x_{11}, ..., x_{k1}) \\ \vdots \\ f(x_{1m}, ..., x_{km}) \end{bmatrix} \in R.$$

Figure 1: Polymorphism

- Operation $major(x, x, y) \approx major(x, y, x) \approx major(y, x, x) \approx x$ is compatible with any unary and binary relation on $\{0, 1\}$;
- Every polymorphism of $\Gamma = \{\leq\}$ has to be monotone. Moreover, any monotone operation on D is a polymorphism of Γ;
- A constant nullary operation $a(\cdot) = a$ is compatible with every relation R that contains a constant tuple $(a, a, ..., a)$;

Constraint satisfaction problem: Complexity

- A set of relations Γ on a fixed domain D is called a relational clone if it contains the equality relation, and is closed under permutations, projection, and intersections (closed under defining new relations via primitive positive formulas).
- Jeavons in 1998 pointed out that for a given CSP language there is a logspace reduction from CSP$(RelClone(\Gamma))$ to CSP(Γ)[7].
- A set of operations O on a fixed domain D is a clone if it contains all projections and is closed under generalized composite, i.e. for a k-ary operation $f \in O$ and m-ary operations $g_1, ..., g_m \in O$ the generalized composite

$$f(g_1(x_1, ..., x_m), ..., g_k(x_1, ..., x_m))$$

  is in O as well.
- There is a Galois duality between relational clones and clones. A relational clone is completely determined by its set of polymorphisms: for any relational structure $\mathcal{A} = (D, \Gamma)$ there exists an algebra $\mathbb{A} = (D, F)$ such that $Clone(F) = Pol(\Gamma)$ where $Pol(\Gamma)$ is the set of all polymorphisms of Γ.

[7] Peter Jeavons. On the algebraic structure of combinatorial problems. Theoretical Computer Science, 200(1-2):185204, 1998.

Constraint satisfaction problem: Complexity

- If all polymorphisms of Γ are unary, then CSP(Γ) is NP-hard;
- The algorithm that is called generalized arc-consistency (a slight weakening of the canonical Datalog program with width 1, in which we only consider one relation at a time in order to remove potential values for the variables) solves any CSP which has an associative, commutative, idempotent polymorphism.
- Bulatov and Dalmau in 2006 provided an algorithm that generalizes Gaussian elimination as well as the algorithm for the general subgroup problem to the case of CSPs with a Mal'cev polymorphism $p(x, y, z) \approx x \approx p(y, y, x)$ for all x, y.
- Zhuk and Bulatov gave two different algorithms that solves in polynomial time all CSP(Γ) for finite Γ that admits a nontrivial polymorphism (not essentially a projection to one of the coordinates) and by that proved CSP dichotomy conjecture.

Unsatisfiable instances of CSP as tautologies

A homomorphism problem between two simle graphs $\mathcal{X} \to \mathcal{A}$:

- For every vertex i in $\mathcal{X} = (V_\mathcal{X}, E_\mathcal{X})$ and vertex j in $\mathcal{A} = (V_\mathcal{A}, E_\mathcal{A})$ set propositional atom p_{ij};
- For every map h from $\mathcal{X}$ to $\mathcal{A}$ set p_{ij} to truth if and only if $h(i) = j$.

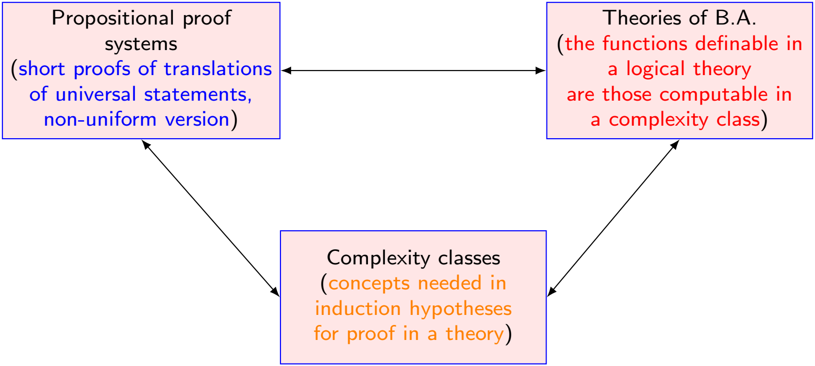
$HOM(\mathcal{X}, \mathcal{A})$ as a propositional formula

$HOM(\mathcal{X}, \mathcal{A})$ is a conjunction of following clauses:

- a clause $(\bigvee_{j \in V_\mathcal{A}} p_{i,j})$ for each $i \in V_\mathcal{X}$ (every vertex of $\mathcal{X}$ must be sent to a vertex of $\mathcal{A}$);
- a clause $(\neg p_{ij_1} \vee \neg p_{ij_2})$ for each $i \in V_\mathcal{X}$ and $j_1, j_2 \in V_\mathcal{A}$ with $j_1 \neq j_2$ (the map must be well-defined);
- a clause $(\neg p_{i_1 j_1} \vee \neg p_{i_2 j_2})$ for every edge $(i_1, i_2) \in E_\mathcal{X}$ and $(j_1, j_2) \notin E_\mathcal{A}$ (a map must be a homomorphism).

- For unsatisfiable instances $\mathcal{X} \to \mathcal{A}$ the formula $\neg HOM(\mathcal{X}, \mathcal{A})$ is a tautology.

Proof complexity



Propositional proof systems (short proofs of translations of universal statements, non-uniform version)

Theories of B.A. (the functions definable in a logical theory are those computable in a complexity class)

Complexity classes (concepts needed in induction hypotheses for proof in a theory)

Propositional proof systems

Definition 4 (Propositional proof system).

A *propositional proof system* is any polynomial time function $P : \{0,1\}^* \to \{0,1\}^*$ whose range is exactly TAUT. For any $\alpha \in$ TAUT, any string ω such that $P(\omega) = \alpha$ is called P-proof of α. P is *p-bounded* if for all tautologies there exists polynomially bounded P-proof.

Theorem 5 (The Cook-Reckhow theorem, 1979).

A p-bounded proof system exists if and only if $NP = coNP$[a].

[a] S. A. Cook and R. A. Reckhow, The relative efficiency of propositional proof systems, J. of Symbolic Logic, 44(1), (1979), pp.36-50.

Proof complexity of CSP

- In 2018 Atserias and Ochremiak showed that, for the most studied proof systems P, the classical log-space reductions between constraint languages preserve the proof complexity of the CSP with respect to proofs in P: if Γ' is obtained from Γ by a finite number of such reductions, then, for any translation of the statement that an instance of CSP is unsatisfiable, efficient proofs of unsatisfiability in P for instances of Γ translate into efficient proofs of unsatisfiability in P for instances of Γ'. Thus, proof systems, for which there exists the constraint language with short unsatisfiability certificates, can be characterized algebraically (in terms of identities).

Theorem 6.

Let Γ be a finite constraint language. Then, exactly one of the following holds:

- Γ *has resolution refutations of constant width (which is equivalent to* Γ *being polynomially solvable by constraint propagation);*
- Γ *has neither bounded-depth Frege refutations of subexponential size, nor Polynomial Calculus over the reals, nor Lasserre/SOS refutations of sublinear degree.*

- We would like to consider proof complexity of all polynomial CSPs, but from another framework.

Theories of Bounded arithmetic

- In theories of Bounded Arithmetic the induction axioms are restricted to bounded formulas of different kinds.
- *Two-sorted* and *three-sorted theories*: the variables $X, Y, Z, \ldots$ (*set variables*) correspond to finite subsets of natural numbers; the variables $\mathscr{X}, \mathscr{Y}, \mathscr{Z}, \ldots$ (*class variables*) correspond to finite sets of finite sets.
    - Σ_0^B - formulas with only bounded number quantifiers and no string quantifiers;
    - Σ_1^B - formulas with bounded number and bounded existential string quantifiers;
    - Σ_∞^B - the set of all second-order bounded formulas;
    - $\Sigma_0^{\mathscr{B}}$ - formulas with arbitrarily many bounded first-order and bounded second-order quantifiers, no class quantifiers;
    - $\Sigma_1^{\mathscr{B}}$ - formulas with arbitrarily many bounded first-order and second-order quantifiers, and class existential quantifiers.

Theories of Bounded arithmetic vs. Proof systems

Definition 7 (Theories of Bounded arithmetic V^1, W^1_1).

1. The two-sorted theory V^1 accepts IND-scheme for all Σ^B_1-formulas and comprehension axiom Σ^B_0-COMP: $\exists Y \leq y \, \forall z < y \, \varphi(\bar{x}, \bar{X}, z) \iff Y(z)$.
Totally definable functions are exactly polynomial time functions FP.
2. The three-sorted theory W^1_1 admits $\Sigma^{\mathscr{B}}_1$-induction and the following two comprehension axiom schemes, namely $\Sigma^{\mathscr{B}}_0$-2COMP:
$\exists Y \leq b \, \forall z \leq b \big(\varphi(\bar{x}, \bar{X}, \bar{\mathscr{X}}, z) \iff Y(z)\big)$, and $\Sigma^{\mathscr{B}}_0$-3COMP:
$\exists \mathscr{Y} \, \forall Z \leq b \big(\varphi(\bar{x}, \bar{X}, \bar{\mathscr{X}}, Z) \iff \mathscr{Y}(Z)\big)$.
Totally definable functions are exactly polynomial space functions FPSPACE.

Definition 8 (Propositional proof systems ER, q.p.c. G).

- The *Extended Resolution proof system* is Resolution with extra initial clauses.

$$\frac{C \cup \{p\} \quad D \cup \{\neg p\}}{C \cup D}.$$

- The *Quantified propositional calculus G* extends classical Sequent calculus LK with quantified rules and allowing quantified propositional formulas.

Theories of Bounded arithmetic vs. Proof systems

Theorem 1 (Translation).

- *Suppose that $\varphi(\bar{x}, \bar{X})$ is a Σ_0^B-formula such that $V^1 \vdash \forall\bar{x}\forall\bar{X}\varphi(\bar{x}, \bar{X})$. Then the propositional family $< \varphi(\bar{x}, \bar{X}) >$ have polynomial size Extended resolution proofs.* [a].
- *Suppose that $\varphi(\bar{x}, \bar{X})$ is a Σ_∞^B-formula such that $W_1^1 \vdash \forall\bar{x}\forall\bar{X}\varphi(\bar{x}, \bar{X})$. Then the propositional family $< \varphi(\bar{x}, \bar{X}) >$ has quantified propositional calculus G-proofs of polynomial size.* [b]

[a] Jan Krajicek. Bounded Arithmetic, Propositional Logic and Complexity Theory. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1995

[b] Alan Skelley. A third-order bounded arithmetic theory for pspace. In Jerzy Marcinkowski and Andrzej Tarlecki, editors, Computer Science Logic, pages 340–354, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg

Proof complexity of a CSP dichotomy proof

- Via log-space reductions, it is enough to consider a CSP language with at most binary constraint relations, so we can restrict ourselves to some sort of a digraph homomorphism problem: $\mathcal{X} \to \mathcal{A}$.
- Zhuk's algorithm solves any tractable CSP(Γ) in p-time and is based on properties of the corresponding algebra, on consistency properties of an instance $\mathcal{X}$ and on Gaussian elimination.[8]
- Instead of one domain D, for every variable $x_i \in \mathcal{X}$ the algorithm considers its own corresponding domain $D_i \leq D$ as an additional unary constraint. The algorithm decides the problem with step-by-step reduction of domains and some in-between modifications of $\mathcal{X}$, $\mathcal{A}$, such that neither reductions nor modifications eliminate all the solutions.
- Soundness of Zhuk's algorithm: if $I = (I_1, I_2, ..., I_k)$ is the computation of Zhuk's algorithm on the instance $\mathcal{X}$ with $I_j = (\mathcal{X}_j, \mathcal{A}_j)$, then the proof of

  $$\forall i < k \ (HOM(\mathcal{X}_i, \mathcal{A}_i) \to HOM(\mathcal{X}_{i+1}, \mathcal{A}_{i+1})) \wedge \neg HOM(\mathcal{X}_k, \mathcal{A}_k)$$

  is the short and independent proof of $\neg HOM(\mathcal{X}, \mathcal{A})$, i.e. the proof of translation $< \neg HOM(\mathcal{X}, \mathcal{A}) >$ is an unsatisfiability certificate in the corresponding proof system.

[8] Dmitriy Zhuk. A proof of the csp dichotomy conjecture. J. ACM, 67(5):1–78, August 2020

Proof complexity of a CSP dichotomy proof

Definition 9 (Theory $V^1_{\mathcal{A}}$).

For any tractable relational structure $\mathcal{A}$,

$$V^1_{\mathcal{A}} := V^1 + \{\text{BA}_{\mathcal{A}}\text{-axioms, CR}_{\mathcal{A}}\text{-axioms, PC}_{\mathcal{A}}\text{-axioms}\}.$$

Additional axiom schemes consist of a finitely many $\forall\Sigma^B_2$-formulas ($\mathcal{A}$ is fixed).

Theorem 10.

For any fixed relational structure $\mathcal{A}$ with tractable CSP($\mathcal{A}$), the theory $V^1_{\mathcal{A}}$ proves the soundness of Zhuk's algorithm.

Theorem 11 (The main theorem).

For any particular relational structure $\mathcal{A}$ such that CSP($\mathcal{A}$) is in P:

1. *Theory W^1_1 proves the soundness of Zhuk's algorithm.*
2. *There exists a p-time algorithm F such that for any unsatisfiable instance $\mathcal{X}$, i.e. such that $\neg HOM(\mathcal{X}, \mathcal{A})$, the output $F(\mathcal{X})$ of F on $\mathcal{X}$ is a propositional proof of $< \neg HOM(\mathcal{X}, \mathcal{A}) >$ in propositional calculus G.*

Open questions

1. Whether the formalization of the algorithm in a weaker theory of bounded arithmetic is possible?
2. Formalization of Bulatov's algorithm: used another methods of universal algebra providing another proof for CSP dichotomy theorem[9], such as separation congruences, quasi-centralizers and decomposition of instance to smaller subinstances.
3. Formalization of algorithms for smaller tractability classes:
   - CSPs with Mal'tsev polymorphisms;
   - CSPs that can be solved by LP relaxations;
   - CSPs with few subpowers - the solution sets always have small "representations".
4. Formalization of some subclasses of other types of CSP:
   - CSP-WNU problem (where we do not know the specific polymorphism, just know that it exists);
   - Valued Constraint Satisfaction Problem (VCSP), where constraint relations are replaced by mappings to the set of rational numbers, and conjunctions are replaced by sum;
   - Quantified Constraint Satisfaction Problem (QCSP), where the relations can be constructed with both $\exists, \forall$ and so forth.

Thank you!

[9] Andrei A. Bulatov. A dichotomy theorem for nonuniform csps. In 2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS), pages 319–330, 2017.