Lecture: Constraint satisfaction problem: the algebraic approach

Svetlana N. Selezneva
selezn@cs.msu.ru

Lomonosov Moscow State University
Faculty of Computational Mathematics and Cybernetics

Course: Modern trends in discrete mathematics and computer science

Relations

For any set A, and any natural number n, the set of all n-tuples of elements of A is denoted by A^n.

Any subset of A^n is called an n-ary *relation* over A.

The set of all finitary relations over A is denoted by R_A.

A *constraint language* over A is a subset of R_A.

Constraint satisfaction problem

For any set A, and any constraint language Γ over A, the *constraint satisfaction problem* $CSP(\Gamma)$ is the combinatorial decision problem with

Instance: A triple (V, A, C), where
1) V is a finite set of variables,
2) C is a set of constraints, $C = \{C_1, \ldots, C_m\}$.
Each constraint $C_i \in C$ is a pair (s_i, ρ_i), where
a) s_i is a tuple of variables of lenght n_i, called the *constraint scope*,
b) $\rho_i \in \Gamma$ is an n_i-ary realation over A, called the *constraint relation*.
Question: does there exist a *solution*, that is, a function $\varphi : V \to A$, such that, for each constraint $(s_i, \rho_i) \in C$, with $s_i = (x_{i_1}, \ldots, x_{i_{n_i}})$, the tuple $(\varphi(x_{i_1}), \ldots, \varphi(x_{i_{n_i}}))$ belongs to ρ_i?

Size of an instance

We consider only cases where the set A is finite.

In such cases, the *size* of a problem instance is the length of a string containing all constraint scopes and all tuples of all constraint relations from the instance.

3-Satisfiability

Example 1.
An **instance** of the **3-Satisfiability** problem is a formula

$$F = F_1 \& \ldots \& F_m,$$

where F_i is a *clauses* with three literals.

The **question** is whether there are values for the variables that make the formula true.

3-Satisfiability

Consider the **3-Satisfiability** problem as a CSP.

The corresponding **instance** is $(V, \{0,1\}, C)$, where
1) V is the set of variables appearing in all clauses F_i,
2) $C = \{C_1, \ldots, C_m\}$, where $C_i = (s_i, \rho_i)$, and
a) $s_i = (x_{i_1}, x_{i_2}, x_{i_3})$, if $F_i = x_{i_1}^{a_1} \vee x_{i_2}^{a_2} \vee x_{i_3}^{a_3}$,
b) $\rho_i = \{0,1\}^3 \setminus \{(a_1, a_2, a_3)\}$.

The solutions of this instance are exactly the assignments which make the formula F true.

$CSP(\Gamma_{3Sat})$

Denote the constraint language over $\{0, 1\}$ consisting of all relations expressible as 3-clauses by Γ_{3Sat}.

Then the **CSP**(Γ_{3Sat}) is a CSP defining the **3-Satisfiability** problem.

□

Graph q-Colorability

Example 2.
An **instance** of the **Graph q-Colorability** problem consists of a graph $G = (V, E)$.

The **question** is whether the vertices of G can be labelled with q colours so that adjacent vertices are assigned different colours.

Graph q-Colorability

Consider the **Graph q-Colorability** problem as a CSP.

Let $G = (V, E)$ be a graph, in which $E = \{e_1, \ldots, e_m\}$, $e_i = (v_{i_1}, v_{i_2})$, $v_{i_1}, v_{i_2} \in V$.

The corresponding **instance** is (V, A, C), where
$A = \{0, 1, \ldots, q-1\}$ is the set of the colours,
1) V is the set of all vertices of G,
2) $C = \{C_1, \ldots, C_m\}$, $C_i = (e_i, \neq_A)$, and $\neq_A \subseteq A^2$ is the *disequality* relation over A, defined by

$$\neq_A = \{(a, b) \in A^2 \mid a \neq b\}.$$

The solutions of this instance are exactly the q-coloring of G.

$CSP(\neq_A)$

Therefore, the **Graph q-Colorability** problem corresponds to the **CSP**$(\{\neq_A\})$.

□

Tractable problems

A problem is *tractable* if there exists a deterministic polynomial-time algorithm solving all instances of this problem.

Tractable languages

For any set A, a *finite* constraint language $\Gamma \subseteq R_A$ is said to be *tractable* if $CSP(\Gamma)$ is tractable.

An *infinite* constraint language $\Gamma \subseteq R_A$ is said to be *tractable* if *every finite subset* of Γ is tractable.

A constraint language $\Gamma \subseteq R_A$ is said to be *NP-complete* if $CSP(\Delta)$ is NP-complete for some finite $\Delta \subseteq \Gamma$.

Graph q-Colorability

Example 3.
It is know that the **Graph q-Colorability** problem is tractable when $q \leq 2$ and NP-complete otherwise.

Hence, it follows from Example 2 that the finite constraint language containing the single relation $\neq_A$ is tractable when $|A| \leq 2$ and NP-complete otherwise.

□

Globally tractable languages

For any set A, a *finite* or *infinite* constraint language $\Gamma \subseteq R_A$ is called *globally tractable* if $CSP(\Gamma)$ is tractable.

Clearly if a constraint language is globally tractable then it is tractable.

But it is not immediately clear whether or not the converse holds for all infinite languages.

This is because for a certain infinite constraint language Γ, it may be the case that for each finite subset $\Delta \subseteq \Gamma$ there exists a polynomial-time algorithm $Alg(\Delta)$ solving $CSP(\Delta)$, and there is no *uniform* polynomial-time algorithm solving $CSP(\Gamma)$.

But we don't know examples of this case.

Globally tractable languages

Conjecture 1. Any tractable constraint language is also globally tractable.

$CSP(\Gamma_{Lin})$

Example 4.
Let A be a finite field, and let Γ_{Lin} be the constraint language consisting of all relations over A which contain all solutions to some system of linear equations over A.

Any relation from Γ_{Lin}, and therefore any instance of $CSP(\Gamma_{Lin})$, can be represented by a system of linear equations over A.

$CSP(\Gamma_{Lin})$

Indeed, if $\rho \in \Gamma_{Lin}$ then it is the solution space of the system of linear equations obtained by the following procedure:

Step 1. Pick an arbitrary element $(a_{01}, \ldots, a_{0n}) \in \rho$, and set

$$\rho_0 = \{(b_1 - a_{01}, \ldots, b_n - a_{0n}) \mid (b_1, \ldots, b_n) \in \rho\}.$$

Step 2. For every member $(a_1, \ldots, a_n) \in \rho_0$, form the equation

$$a_1x_1 + \ldots + a_nx_n = 0,$$

and find a basis, $\rho^{\perp}$, of the solution space of the resulting system of equations.

Step 3. For each $(b_1, \ldots, b_n) \in \rho^{\perp}$, output the equation

$$b_1x_1 + \ldots + b_nx_n = b_0,$$

where $b_0 = b_1a_{01} + \ldots + b_na_{0n}$.

$CSP(\Gamma_{Lin})$

Since any instance of $CSP(\Gamma_{Lin})$ may be solved in polynomial time by Gausian elimination, it follows that Γ_{Lin} is *globally* tractable constraint language.

□

Boolean constraints

A constraint language over the set $A = \{0, 1\}$ is known as *Boolean constraint language*.

The complexity of $CSP(\Gamma)$ has been investigated by *T. Schaefer* in 1978 for all Boolean constraint languages, and the following complete classification has been obtained.

Schaefer's theorem

Theorem 1 (T. Schaefer, 1978). *A Boolean constraint language, Γ, is tractable if at least one of the following conditions holds; otherwise it is NP-complete.*

Schaefer's theorem

Theorem 1 (T. Schaefer, 1978). *These conditions are:*
1) *every relation in* $\Gamma \setminus \{0\}$ *contains a tuple in which all entries are* 0*;*
2) *every relation in* $\Gamma \setminus \{0\}$ *contains a tuple in which all entries are* 1*;*
3) *every relation in* Γ *is definable by a formula in conjunctive normal form in which each conjunct has at most one negated variable;*
4) *every relation in* Γ *is definable by a formula in conjunctive normal form in which each conjunct has at most one unnegated variable;*
5) *every relation in* Γ *is definable by a formula in conjunctive normal form in which each conjunct has at most two literals;*
6) *every relation in* Γ *is the set of solutions of a system of linear equations over the finite field* $GF(2)$*.*

Dichotomy theorems

In particular, Theorem 1 established that any Boolean constraint language is either tractable or *NP*-complete.

This result is known as *Schaefer's dichotomy theorem*.

A simialar dichotomy theorem has been obtained by *A. Bulatov* in 2002 for constraint languages over a set A with three elements.

Tractable relations problem

Problem 1. Characterize all tractable constraint languages over finite sets.

A useful first step in deciding the problem is to consider what additional relations can be added to a constraint language without changing the complexity of corresponding problem class.

Relational clones

A set of relations $\Gamma \subseteq R_A$ is called a *relational clone* if it contains all relations expressilbe by a formula involving
1) relations from $\Gamma \cup \{=_A\}$ where $=_A$ is the equality relation over A;
2) *conjunction*;
3) *existential quantification*.

Formulas involving only *conjunction* and *existential quantification* are called *primitive positive* (pp) formulas.

For any set of relations $\Gamma \subseteq R_A$, there is a *unique* smallest relational clone containing Γ, which is denoted $\langle \Gamma \rangle$ and is called the relational clone *generated by* Γ.

Binary Boolean relations

Example 5. Consider the Boolean constraint language $\Gamma = \{\rho_1, \rho_2\}$, where

$$\rho_1 = \{(0,1),(1,0),(1,1)\},$$
$$\rho_2 = \{(0,0),(0,1),(1,0)\}.$$

It is straightforward to check that every binary Boolean relation can be expressed by a pp-formula involving ρ_1 and ρ_2.
For example, the relation $\rho_3 = \{(0,0),(1,0),(1,1)\}$ can be expressed by the formula:

$$\rho_3(x,y) = \exists z \rho_1(x,z) \& \rho_2(z,y).$$

It can be shown that $\langle\Gamma\rangle$ consists of presisely those Boolean relations that can be expressed as a conjunction of unary or binary Boolean relations. $\square$

Polynomial time reduction

Theorem 2. *For any set of relations* $\Gamma \subseteq R_A$ *and any finite set* $\Delta \subseteq \langle \Gamma \rangle$*, there is a polynomial time reduction from* $CSP(\Delta)$ *to* $CSP(\Gamma)$.

Proof. Let $\Delta \subseteq \langle \Gamma \rangle$ be a finite set, where each relation $\rho \in \Delta$ is expressible by pp-formula involving relations from Γ and the equality relation, $=_A$.

Polynomial time reduction

Proof. Any instance $(V, A, C) \in CSP(\Delta)$ can be transformed as follows.

For every constraint $(s, \rho) \in C$, where $s = (v_1, \ldots, v_n)$, and $\rho(v_1, \ldots, v_n)$ is representable by the pp-formula

$$\rho = \exists u_1, \ldots, u_k(\tau_1(w_{11}, \ldots, w_{1n_1})\& \ldots \&\tau_m(w_{m1}, \ldots, w_{mn_m})),$$

where $\tau_1, \ldots, \tau_m \in \Gamma \cup \{=_A\}$,
$w_{11}, \ldots, w_{1n_1}, \ldots, w_{m1}, \ldots, w_{mn_m} \in \{v_1, \ldots, v_n, u_1, \ldots, u_k\}$,

1) add the auxiliary variables $u_1, \ldots, u_k$ to V (renaming if necessary so that none of them occurs before);
2) add the constraints
$((w_{11}, \ldots, w_{1n_1}), \tau_1), \ldots, ((w_{m1}, \ldots, w_{mn_m}), \tau_m)$ to C;
3) remove (s, ρ) from C.

Polynomial time reduction

Proof. The problem instance obtained by this procedure is equivalent to (V, A, C) and belongs to $CSP(\Gamma \cup \{=_A\})$.
This transformation can be carried out in linear time of the size of the instance.

Finally, all constraints of the form $((v_1, v_2), =_A)$ can be eliminated by replacing all occurrences of the variable v_1 with v_2.
This transformation can also be carried out in polynomial time.

□

Sets of relations and relational clones

Corollary 2.1.
1. *A set of relations* $\Gamma \subseteq R_A$ *is tractable if and only if the relational clone* $\langle\Gamma\rangle$ *is tractable.*
2. *Similarly, a set of relations* $\Gamma \subseteq R_A$ *is NP-complete if and only if the relational clone* $\langle\Gamma\rangle$ *is NP-complete.*

How to represent and describe relational clone?

There is a well-known way for this, using *operations*.

Operations

For any set A, and any natural number n, a mapping $f : A^n \to A$ is called an n-ary *operation* on A.

The set of all finitary operations on A is denoted by O_A.

For any m-ary operation $f \in O_A$ and any collection of tuples $\bar{a}_1, \ldots, \bar{a}_m \in A^n$, where $\bar{a}_i = (a_{1i}, \ldots, a_{ni})$, define $f(\bar{a}_1, \ldots, \bar{a}_m)$ to be

$$(f(a_{11}, \ldots, a_{1m}), \ldots, f(a_{n1}, \ldots, a_{nm})) \in A^n.$$

Polymorphisms

An m-ary operation $f \in O_A$ *preserves* an n-ary relation $\rho \in R_A$ (or f is a *polymorphism* of ρ, or ρ is *invariant* under f) if $f(\bar{a}_1, \dots, \bar{a}_m) \in \rho$ for all choices of $\bar{a}_1, \dots, \bar{a}_m \in \rho$.

For any given sets $\Gamma \subseteq R_A$ and $F \subseteq O_A$, let

$$\mathrm{Pol}(\Gamma) = \{f \in O_A \mid f \text{ preserves each relation from } \Gamma\},$$
$$\mathrm{Inv}(F) = \{\rho \in R_A \mid \rho \text{ is invariant under each operation from } F\}.$$

Unary polymorphisms

Theorem 3 (P. Jeavons, 1997). *For any set* $\Gamma \subseteq R_A$ *and any unary operation* $f \in \mathrm{Pol}(\Gamma)$*, let* $f(\Gamma)$ *be the set* $\{f(\rho) \mid \rho \in \Gamma\}$ *where*

$$f(\rho) = \{f(\bar{a}) \mid \bar{a} \in \rho\}.$$

The constraint language Γ *is tractable if and only if* $f(\Gamma)$ *is tractable, and the constraint language* Γ *is NP-complete if and only if* $f(\Gamma)$ *is NP-complete.*

Unary polymorphisms

Proof. Any instance $\mathcal{P} = (V, A, C)$ of $CSP(\Gamma)$ can be transformed into the instance $\mathcal{P}' = (V, A, C')$ of $CSP(f(\Gamma))$ where

$$C' = \{(s, f(\rho)) \mid (s, \rho) \in C\}.$$

Now, if $\varphi : V \to A$ is a solution of the instance $\mathcal{P}$, then $\varphi' : V \to A$, where $\varphi'(x) = \varphi(f(x))$, is a solution of the instance $\mathcal{P}'$ by reason of $f \in \mathrm{Pol}(\Gamma)$.

Since $f \in \mathrm{Pol}(\Gamma)$, any solution of $\mathcal{P}'$ is also a solution of $\mathcal{P}$.

The converse is proved in a similar way. $\square$

Nonsurjective unary polymorphisms

It follows from Theorem 3 that, for a constraint language $\Gamma \subseteq R_A$, if the set $\mathrm{Pol}(\Gamma)$ contains an nonsurjective unary operation f, then we can reduce this consrtraint language to the consrtraint language $f(\Gamma) \subseteq R_B$, where $B \subseteq A$, $B \neq A$, without reducing the corresponding problem class.

Example 5. Let A be a finite set and $\Gamma \subseteq R_A$ be a constraint language such that $f_a \in \mathrm{Pol}(\Gamma)$ where $f_a(x)$ is the a-constant operation, i.e., $f_a(x) = a$, $a \in A$.

Then the constrant language $f_a(\Gamma) \subseteq R_{\{a\}}$ contains precisely tuples in which all entries are a, and this constraint language is tractable.

Hence the constraint language Γ is tractable by Theorem 3.

□

Galois correspondence

We remark that the operators Pol and Inv form a Galois correspondence between R_A and O_A.

Proposition 1. *For any set A, and any $F \subseteq O_A$, the set $\mathrm{Inv}(F)$ is a relational clone. Conversely, any relational clone can be represented in the form $\mathrm{Inv}(F)$ for some set $F \subseteq O_A$. In particular, for any $\Gamma \subseteq R_A$, $\langle \Gamma \rangle = \mathrm{Inv}(\mathrm{Pol}(\Gamma))$.*

Clones of operations

A set of operations $F \subseteq O_A$ is called a *clone* if it contains all operations expressilbe by a *superposition* involving
1) operations from F;
2) all projections (i.e., all operations $f(x_1, \dots, x_n) = x_i$ for $n \geq 1$, and $1 \leq i \leq n$).

For any set of operations $F \subseteq O_A$, there is a *unique* smallest clone containing F, which is denoted $[F]$ and is called the clone *generated by F*.

Galois correspondence

Proposition 2. *For any set* A*, and any* $\Gamma \subseteq R_A$*, the set* $\mathrm{Pol}(\Gamma)$ *is a clone. Conversely, any clone can be represented in the form* $\mathrm{Pol}(\Gamma)$ *for some set* $\Gamma \subseteq R_A$*. In particular, for any* $F \subseteq O_A$*,* $[F] = \mathrm{Pol}(\mathrm{Inv}(F))$*.*

Tractable sets of operations

For any set A, a set $F \subseteq O_A$ is said to be *tractable* if $\mathrm{Inv}(F)$ is tractable.

A set $F \subseteq O_A$ is said to be *NP-complete* if $\mathrm{Inv}(F)$ is NP-complete.

Tractable operations problem

Problem 2. Characterize all tractable sets of operations on finite sets.

$\mathrm{Inv}(\{a\})$ where $a \in A$

Example 6. Let A be a finite set and $\Gamma \subseteq R_A$ be a constraint language such that $f_a \in \mathrm{Pol}(\Gamma)$ where $f_a(x)$ is the a-constant operation, i.e., $f_a(x) = a$, $a \in A$.

Hence the set of operations $\{a\} \subseteq O_A$, where $a \in A$, is tractable by Example 5, and any set $F \subseteq O_A$ containing a constant operation is tractable.

□

$\mathrm{Inv}(\{x - y + z\})$

Example 7. Let A be a finite field. It was shown that $\langle \Gamma_{Lin} \rangle$ is precisely the consrtaint language consisting of all relations which are invariant under the ternary operation $f(x, y, z) = x - y + z \in O_A$.

Hence the set of operations $\{x - y + z\} \subseteq O_A$ is tractable by Example 4, and any set $F \subseteq O_A$ containing $x - y + z$ is tractable.

□

Essentially unary and nonunary operations

An operation $f : A^n \to A$ is called *essentially unary* if there exists a (nonconstant) unary operation $g : A \to A$ and an index $i \in \{1, \ldots, n\}$ such that $f(a_1, \ldots, a_n) = g(a_i)$ for all choices $a_1, \ldots, a_n \in A$.

If g is the identity operation, then f is called a *projection*.

Any operation which is not essentially unary (including all constant operations) is called *essentially nonunary*.

Essentially unary operations

Theorem 4 (P. Jeavons, 1997). *For any finite set A and any set $\Gamma \subseteq R_A$, if $\mathrm{Pol}(\Gamma)$ contains essentially unary operations only, then $CSP(\Gamma)$ is NP-complete.*

Proof. Consider the case when $\mathrm{Pol}(\Gamma)$ contains only permutations.

Essentially unary operations

Proof.
1. If $|A| = 2$, $A = \{0,1\}$, then $\mathrm{Pol}(\Gamma) \subseteq \{x, x \oplus 1\}$. Consider the ternary relation ρ,

$$\rho = \{0,1\}^3 \setminus \{(0,0,0),(1,1,1)\}.$$

For any tuple $(a_1, a_2, a_3) \in A^3$, if $(a_1, a_2, a_3) \in \rho$, then $(a_1 \oplus 1, a_2 \oplus 1, a_3 \oplus 1) \in \rho$.

Hence $\rho \in \langle \Gamma \rangle$, and $CSP(\{\rho\})$ is polynomial-time reducible to $CSP(\Gamma)$ by Theorem 2.

Since $CSP(\{\rho\})$ is NP-complete, we obtain $CSP(\Gamma)$ is also NP-complete.

Essentially unary operations

Proof.

2. If $|A| \geq 3$, then consider the binary relation $\neq_A$.

For any tuple $(a_1, a_2) \in A^2$, if $a_1 \neq_A a_2$, then $g(a_1) \neq_A g(a_2)$ for any permutation $g \in \mathrm{Pol}(\Gamma)$.

Hence $\neq_A \in \langle \Gamma \rangle$, and $CSP(\{\neq_A\})$ is polynomial-time reducible to $CSP(\Gamma)$ by Theorem 2.

Since, by Example 3, $CSP(\{\neq_A\})$ is NP-complete for $|A| \geq 3$, we obtain $CSP(\Gamma)$ is also NP-complete.

□

Schaefer's dichotomy for operations

It is known that if a Boolean constraint language is not contained in one of Schaefer's six tractable classes, then all of its polymorphisms are essentially unary operations.

Hence we may reformulate Schaefer's dichotomy theorem to obtain the following complete classification for Boolean operations.

Theorem 1 (T. Schaefer, 1978). *A set of Boolean operations is tractable if it contains an essentially nonunary operation. Otherwise it is NP-complete.*

Converse of Theorem 4

Example 8. Consider the binary operation $f(x, y)$ on the set $A = \{0, 1, 2\}$ defined by the following table:

$x \setminus y$	0	1	2
0	0	1	1
1	1	1	0
2	2	2	2

It can be shown that the set $\{f\} \subseteq O_A$ is NP-complete, but f is not essentially unary.

Hence the converse of Theorem 4 is not true for $|A| \geq 3$.

□

Consider more powerful tools which are algebras.

Algebras

An *algebra* is an ordered pair $\mathcal{A} = (A, F)$ such that A is a nonempty set and F is a family of finitary operations on A.

The set A is called the *universe* of $\mathcal{A}$, and the operations of F are called *basic*.

An algebra with a finite universe is referred to as a *finite algebra*.

From sets of operations to algebras

To make the translation from sets of operations to algebras we simply note that any set of operations F on a fixed set A can be associated with the algebra $\mathcal{A} = (A, F)$.

Tractable algebras

An algebra $\mathcal{A} = (A, F)$ is said to be tractable if F is tractable.

An algebra $\mathcal{A} = (A, F)$ is said to be *NP*-complete if F is *NP*-complete.

Tractable algebras problem

Problem 3. Characterize all tractable algebras.

Schaefer's dichotomy for algebras

We can reformulate Schaefer's dichotomy theorem as a classification of the complexity of algebras defined on a two-element set.

Theorem 1 (T. Schaefer, 1978). *An algebra with a two-element universe is NP-complete if all of its basic operations are essentially unary. Otherwise it is tractable.*

Advantages of algebras

An advantage of using algebaras instead sets of operations is that we can make use of some standard constructions on algebras to obtain new results about the complexity of constraint languages.

Term operations

For any algebra $\mathcal{A} = (A, F)$, an operation f is called a *term operation* if $f \in [F]$ (or $f \in \mathrm{Pol}(\mathrm{Inv}(F))$ by reason of $[F] = \mathrm{Pol}(\mathrm{Inv}(F))$).

The set of all term operations of $\mathcal{A}$ is denoted $\mathrm{Term}(\mathcal{A})$.

Two algebras with the same universe are called *term equivalent* if they have the same set of term operations.

Surjective algebras

An algebra $\mathcal{A} = (A, F)$ is called *surjective* if all its term operations are surjective.

By Theorem 3, we can restrict our attention to surjective algebras.

Idempotent algebras

An operation $f \in O_A$ is called *idempotent* if it satisfies $f(x, \ldots, x) = x$.

An algebra $\mathcal{A} = (A, F)$ is called *idempotent* if all its term operations are idempotent.

Full idempotent reducts

The *full idempotent reduct* of an algebra $\mathcal{A} = (A, F)$ is the algebra

$$\mathcal{A}_0 = (A, \mathrm{Term}_{id}(\mathcal{A})),$$

where $\mathrm{Term}_{id}(\mathcal{A})$ consists of all idempotent operations from $\mathrm{Term}(\mathcal{A})$.

Theorem 5 (A. Bulatov, P. Jeavons, A. Krokhin, 2005). *A finite surjective algebra $\mathcal{A}$ is tractable if and only if its full idempotent reduct $\mathcal{A}_0$ is tractable. Moreover, $\mathcal{A}$ is NP-complete if and only if its $\mathcal{A}_0$ is NP-complete.*

It follows from Theorem 5 that we may consider only surjective idempotent algebras.

Now we link the complexity of an algebra with the complexity of its *subalgebras* and *homomorphic images*.

In many cases, we can reduce the problem of analyzing the complexity of an algebra to a similar problem involving an algebra with a smaller universe.

Subalgebras

Let $\mathcal{A} = (A, F)$ be an algebra and B be a subset of A such that, for any $f \in F$ and any $b_1, \ldots, b_n \in B$, where n is an arity of f, we have $f(b_1, \ldots, b_n) \in B$.

Then the algebra $\mathcal{B} = (B, F|_B)$ is called a *subalgebra* of $\mathcal{A}$, where $F|_B$ consists of the restrictions of all operations in F to B.

If $B \neq A$, then $\mathcal{B}$ is said to be a *proper* subalgebra.

Subalgebras

Theorem 6 (A. Bulatov, P. Jeavons, A. Krokhin, 2005).
Let $\mathcal{A}$ be a finite algebra.
1. *If $\mathcal{A}$ is tractable, then so is every subalgebra of $\mathcal{A}$.*
2. *If $\mathcal{A}$ has an NP-complete subalgebra, then $\mathcal{A}$ is NP-complete.*

Proof. Let $\mathcal{B} = (B, F|_B)$ be a subalgebra of $\mathcal{A} = (A, F)$. It is easy to check that $\mathrm{Inv}(F|_B) \subseteq \mathrm{Inv}(F)$. Hence, $CSP(\mathrm{Inv}(F|_B))$ can be reduced to $CSP(\mathrm{Inv}(F))$ in constant time.
Now 1 and 2 follow immediately from the existence of this reduction.

□

Homomorphic images

Let $\mathcal{A}_1 = (A_1, F_1)$ and $\mathcal{A}_2 = (A_2, F_2)$ be algebras such that $F_1 = \{f_i^{(1)} \mid i \in I\}$ and $F_2 = \{f_i^{(2)} \mid i \in I\}$ where both $f_i^{(1)}$ and $f_i^{(2)}$ are n_i-ary, for all $i \in I$.

A map $\varphi : A_1 \to A_2$ is called a *homomorphism* from $\mathcal{A}_1$ to $\mathcal{A}_2$ if

$$\varphi(f_i^{(1)}(a_1, \ldots, a_{n_i})) = f_i^{(2)}(\varphi(a_1), \ldots, \varphi(a_{n_i}))$$

holds for all $i \in I$ and all $a_1, \ldots, a_{n_i} \in A_1$.

If the map φ is surjective, then $\mathcal{A}_2$ is said to be a *homomorphic image* of $\mathcal{A}_1$.

Homomorphic images

Theorem 7 (A. Bulatov, P. Jeavons, A. Krokhin, 2005).
Let $\mathcal{A}$ be a finite algebra.
1. *If $\mathcal{A}$ is tractable, then so is every homomorphic image of $\mathcal{A}$.*
2. *If $\mathcal{A}$ has an NP-complete homomorphic image, then $\mathcal{A}$ is NP-complete.*

Proof. Let $\mathcal{B} = (B, F_B)$ be a homomorphic image of $\mathcal{A} = (A, F_A)$, and let φ be the corresponding homomorphism.
We shall show that, for any $\Gamma \subseteq \mathrm{Inv}(F_B)$, $CSP(\Gamma)$ is linear-time reducible to $CSP(\Gamma')$ for some $\Gamma' \subseteq \mathrm{Inv}(F_A)$.

Homomorphic images

Proof. For $\rho \in \mathrm{Inv}(F_B)$, set

$$\varphi^{-1}(\rho) = \{(a_1, \dots, a_n) \in A^n \mid (\varphi(a_1), \dots, \varphi(a_n)) \in \rho\}.$$

It is clear that $\varphi^{-1}(\rho)$ is a relation of the same arity as ρ.
It can be checked that $\varphi^{-1}(\rho) \in \mathrm{Inv}(F_A)$.

Let $\Gamma' = \{\varphi^{-1}(\rho) \mid \rho \in \Gamma\}$. Then Γ' is a finite subset of $\mathrm{Inv}(F_A)$.

Homomorphic images

Proof. Take an instance $\mathcal{P} = (V, B, C)$ of $CSP(\Gamma)$ and construct the instance $\mathcal{P}' = (V, A, C')$ of $CSP(\Gamma')$ where

$$C' = \{(s, \varphi^{-1}(\rho)) \mid (s, \rho) \in C\}.$$

If $\psi : V \to B$ is a solution of $\mathcal{P}$, then any function $\tau : V \to A$ such that $\varphi(\tau(v)) = \psi(v)$ for any $v \in V$ is a solution of $\mathcal{P}'$.

Conversely, if $\tau : V \to A$ is a solution of $\mathcal{P}'$, then $\psi : V \to B$ such that $\psi(x) = \varphi(\tau(x))$ is a solution of $\mathcal{P}$.

□

Examples

Now we give two examples to illustrate the use of Theorem 6 and Theorem 7.

The examples show that *both* of these results can be useful to establish the complexity of certain algebras by reducing the question to an algebra over a smaller set.

An *NP*-complete algebra with tractable proper subalgebras

Example 9. Let $\mathcal{A}$ be the idempotent algebra $(\{0,1,2\},\{f\})$, where the binary operation $f(x,y)$ defined by the following table:

$x \setminus y$	0	1	2
0	0	1	1
1	1	1	0
2	2	2	2

By using Theorem 7, we show that $\mathcal{A}$ is *NP*-complete even though all of its proper subalgebras are tractable.

An *NP*-complete algebra with tractable proper subalgebras

Example 9.

$x \setminus y$	0	1	2
0	0	1	1
1	1	1	0
2	2	2	2

Notice that, as the equalities

$$f(0,2) = 1,\; f(1,2) = 0,\; f(0,1) = f(1,0) = 1$$

show, $\mathcal{A}$ has only one proper subalgebra $\mathcal{B} = (\{0,1\}, f|_{\{0,1\}})$. It is easy to check that $f|_{\{0,1\}} = x \vee y$. Hence, by Theorem 1, the algebra $\mathcal{B}$ is tractable.

An NP-complete algebra with tractable proper subalgebras

Example 9.

$x \setminus y$	0	1	2
0	0	1	1
1	1	1	0
2	2	2	2

On the other hand, consider the algebra $\mathcal{C} = (C, g)$, where $C = \{a, b\}$ and for all $x, y \in C$, $g(x, y) = x$. It is easy to check that the mapping $\varphi : \{0, 1, 2\} \to C$ such that

$$\varphi(0) = \varphi(1) = a,\ \varphi(2) = b,$$

is a homomorphism from $\mathcal{A}$ to $\mathcal{C}$. By Theorem 1, $\mathcal{C}$ is NP-complete. Hence, by Theorem 7, $\mathcal{A}$ is NP-complete.

□

An NP-complete algebra with tractable homomorphic smaller images

Example 10. Let $\mathcal{A}$ be the idempotent algebra $(\{0, 1, 2\}, \{f\})$, where the binary operation $f(x, y)$ defined by the following table:

$x \setminus y$	0	1	2
0	0	1	1
1	0	1	1
2	1	1	2

By using Theorem 6, we show that $\mathcal{A}$ is NP-complete even though all of its smaller homomorphic images are tractable.

An NP-complete algebra with tractable homomorphic smaller images

Example 10.

$x \setminus y$	0	1	2
0	0	1	1
1	0	1	1
2	1	1	2

Since one-element algebras are certainly tractable, we need to consider only two-element homomorphic images $\mathcal{B}$ of $\mathcal{A}$. Let $\mathcal{B} = (\{a, b\}, g)$, where g is a binary operation on $\{a, b\}$, and assume that φ is a homomorphism from $\mathcal{A}$ to $\mathcal{B}$. Then we have that

$$\varphi(f(x, y)) = g(\varphi(x), \varphi(y))$$

for all $x, y \in \{0, 1, 2\}$.

An NP-complete algebra with tractable homomorphic smaller images

Example 10.

$x \setminus y$	0	1	2
0	0	1	1
1	0	1	1
2	1	1	2

Case 1. $\varphi(0) = \varphi(1) = a$, $\varphi(2) = b$. In this case we have

$$\begin{aligned} g(a,a) &= g(\varphi(0),\varphi(0)) = \varphi(f(0,0)) = \varphi(0) = a, \\ g(a,b) &= g(\varphi(0),\varphi(2)) = \varphi(f(0,2)) = \varphi(1) = a, \\ g(b,a) &= g(\varphi(2),\varphi(0)) = \varphi(f(2,0)) = \varphi(1) = a, \\ g(b,b) &= g(\varphi(2),\varphi(2)) = \varphi(f(2,2)) = \varphi(2) = b. \end{aligned}$$

It is easy to check that $g(x,y) = x \& y$ on $\{a,b\}$. Hence, by Theorem 1, the algebra $\mathcal{B}$ is tractable.

An NP-complete algebra with tractable homomorphic smaller images

Example 10.

$x \setminus y$	0	1	2
0	0	1	1
1	0	1	1
2	1	1	2

Case 2. $\varphi(0) \neq \varphi(1)$. It follows that $\varphi(2) \in \{\varphi(0), \varphi(1)\}$.
If $\varphi(2) = \varphi(0)$, then

$$\begin{aligned}\varphi(0) &= \varphi(f(0,0)) = g(\varphi(0), \varphi(0)) = \\ &= g(\varphi(0), \varphi(2)) = \varphi(f(0,2)) = \varphi(1).\end{aligned}$$

If $\varphi(2) = \varphi(1)$, then

$$\begin{aligned}\varphi(0) &= \varphi(f(1,0)) = g(\varphi(1), \varphi(0)) = \\ &= g(\varphi(2), \varphi(0)) = \varphi(f(2,0)) = \varphi(1).\end{aligned}$$

Hence, this second case is impossible, and we have shown that all smaller homomorphic images of $\mathcal{A}$ are tractable.

An *NP*-complete algebra with tractable homomorphic smaller images

Example 10.

$x \setminus y$	0	1	2
0	0	1	1
1	0	1	1
2	1	1	2

On the other hand, the algebra $\mathcal{A}$ has the subalgebra $\mathcal{A}' = (\{0,1\}, f|_{\{0,1\}})$ such that $f|_{\{0,1\}}(x,y) = y$ for all $x, y \in \{0,1\}$. By Theorem 1, $\mathcal{A}'$ is *NP*-complete. Hence, by Theorem 6, $\mathcal{A}$ is *NP*-complete.

□

Factors

A homomorphic image of a subalgebra of an algebra $\mathcal{A}$ is called a *factor* of $\mathcal{A}$.

A factor whose universe contains only a single element is called a *trivial* factor.

Sufficient condition for *NP*-completeness

Using the notion of a factor, we can combine Theorem 4, Theorem 6 and Theorem 7.

Theorem 8.
Let $\mathcal{A}$ be a finite algebra.
1. *If $\mathcal{A}$ is tractable, then so is every factor of $\mathcal{A}$.*
2. *If $\mathcal{A}$ has a nontrivial factor $\mathcal{B}$ all of whose term operations are essentially unary, then $\mathcal{A}$ is NP-complete.*

Operations again

What is the next step?

We show that a failure of the sufficient condition for *NP*-completeness is described by a special operation.

Weak near-unanimity operations

An operation $f : A^n \to A$ is called a *weak near-unanimity* operation (WNU), if $n \geq 2$, $f(x, \dots, x) = x$, and

$$f(y, x, \dots, x) = f(x, y, x, \dots, x) = \dots = f(x, \dots, x, y).$$

Weak near-unanimity operations

Theorem 9 (M. Maroti, R.Mckenzie, 2008).
A finite idempotent algebra $\mathcal{A}$ *does not contain any weak near-unanimity term operations if and only if it has a nontrivial factor* $\mathcal{B}$ *all of whose term operations are essentially unary.*

Sufficient condition for *NP*-completeness

Combining Theorem 8 and Theorem 9, we obtain

Theorem 10. *If a finite idempotent algebra $\mathcal{A}$ does not contain any weak near-unanimity term operations, then $\mathcal{A}$ is NP-complete.*

Sufficient condition for tractability

Theorem 11 (D. Zhuk, A. Bulatov, 2017). *If a finite idempotent algebra $\mathcal{A}$ contains a weak near-unanimity term operation, then $\mathcal{A}$ is tractable.*

Classifying the complexity of finite algebras

Thus, by efforts of many mathematicians, the following beautiful result was obtained.

Theorem 12. *If a finite idempotent algebra $\mathcal{A}$ contains a weak near-unanimity term operation, then $\mathcal{A}$ is tractable. Otherwise, it is NP-complete.*

Return to constraint languages

Now we return to sets of relations and restate Theorem 12 for constraint languages.

Characterization of an idempotent operation

Define $\Gamma_{CON} = \{\rho_a \mid a \in A\}$ where $\rho_a = \{(a)\}$, i.e., Γ_{CON} consists of all unary one-element relations over A.

Proposition 3. *An operation f on a set A is idempotent if and only if it preserves all the relations from Γ_{CON}.*

Proof. Let $f : A^n \to A$ be an operation on a set A.
1. If f is idempotent then $f(a, \ldots, a) = a$ holds for all $a \in A$. Hence f preserves all the relations from Γ_{CON}.
2. If f preserves all the relations from Γ_{CON} then $f(a, \ldots, a) = a$ for all $a \in A$. Hence f is idempotent.

□

Classifying the complexity of constraint languages

Restate Theorem 12 for constraint languages.

Theorem 12. *Suppose* Γ *is a set of relations over a finite set* A*, and* $\Gamma_{CON} \subseteq \Gamma$*. If there exists a weak near-unanimity operation in* $\mathrm{Pol}(\Gamma)$*, then the constraint language* Γ *is tractable; otherwise it is NP-complete.*

References

1. Bulatov A., Jeavons P., Krokhin A. Classifying the complexity of constraints using finite algebras // SIAM J. Comput. 2005. V. 34, N 3. P. 720–742. (This is the base of the lecture)

2. Jeavons P. On the algebraic structure of combinatorial problems // Theoretical Computer Science. 1998. V. 200. P. 185–204.

3. Maroti M., McKenzie R. Existence theorems for weakly symmetric operations // Algebra Universalis. 2008. V. 59. P. 463–489.

4. Zhuk D. An algorithm for constraint satisfaction problem // Proc. of IEEE 47th Int. Symp. on Multiple-Valued Logic. 2017. P. 1–6.