Constraints in Universal Algebra

Ross Willard

University of Waterloo, CAN

SSAOS 2014
September 8, 2014
Lecture 2

Outline

Lecture 1: Intersection problems and congruence SD($\wedge$) varieties

Lecture 2: Constraint problems in ternary groups (and generalizations)

Lecture 3: Constraint problems in Taylor varieties

Review

Let $\mathbf{A}$ be a finite algebra.

Definition. The **constraint satisfaction problem for A** (or CSP($\mathbf{A}$)) is a decision problem.

An **instance of CSP(A)** consists of an integer $n > 1$ (the **degree**) and a list $(s_1, C_1), \ldots, (s_p, C_p)$ of "specifications" of subalgebras of $\mathbf{A}^n$ (of a certain kind).

- The (s_i, C_i) are called "constraints."
- Each s_i is a non-empty subset of $\{1, 2, \ldots, n\}$. (The "scope.")
- C_i is a non-empty subuniverse of $\mathbf{A}^{s_i}$. (The "constraint relation.")
- The subalgebra of $\mathbf{A}^n$ "specified" by (s_i, C_i) is

$$[\![s_i, C_i]\!] \ := \ \{\mathbf{a} \in A^n \, : \, \mathrm{proj}_{s_i}(\mathbf{a}) \in C_i\}.$$

- The **solution-set** of the instance is $[\![s_1, C_1]\!] \cap \cdots \cap [\![s_p, C_p]\!]$.

The question: Given an instance, does it have a solution?

Review (continued)

Definition. An instance $(s_1, C_1), \ldots, (s_p, C_p)$ of CSP($\mathbf{A}$) of degree n is **(2,3)-minimal** if:

- For any two constraints $(s_i, C_i), (s_j, C_j)$,

  if $t \subseteq s_i \cap s_j$ and $1 \leq |t| \leq 2$, then $\mathrm{proj}_t(C_i) = \mathrm{proj}_t(C_j)$.

- For every 3-element subset $t \subseteq \{1, \ldots, n\}$ there exists a constraint (s_i, C_i) such that $t \subseteq s_i$.

(New: **(3,3)-minimal** defined similarly — change ≤ 2 to ≤ 3 in first item. Called **3-minimal** in the literature. Note: 3-minimal $\Rightarrow$ (2,3)-minimal.)

Theorem (Barto 2014 ms, improving Barto, Kozik 2009 (& Bulatov))

Suppose $\mathbf{A}$ is finite and $\mathbf{HSP}(\mathbf{A})$ is congruence SD($\wedge$). Then every (2,3)-minimal instance of CSP($\mathbf{A}$) has a solution.

k-CSP

Definition. Fix $k \geq 2$.

- An instance $(s_1, C_1), \ldots, (s_p, C_p)$ of CSP($\mathbf{A}$) is a k-**ary instance** if $|s_i| \leq k$ for every i.
- k-CSP($\mathbf{A}$) denotes the restriction of CSP($\mathbf{A}$) to k-ary instances.

Central Problem of CSP (Feder, Vardi) – Dichotomy

Given $\mathbf{A}$ and k, either

1. Find a polynomial-time algorithm solving k-CSP($\mathbf{A}$), or
2. Show that k-CSP($\mathbf{A}$) is NP-complete.

As 2ℓ-CSP($\mathbf{A}$) can be reduced to 2-CSP($\mathbf{A}^\ell$), it suffices to solve the Central Problem for 2-CSP($\mathbf{A}$).

Pre-processing

3-CSP($\mathbf{A}$) is slightly more convenient than 2-CSP($\mathbf{A}$).

Given any instance of 3-CSP($\mathbf{A}$), there is any easy (poly-time) algorithm which either:

1. Discovers a "contradiction" (i.e., a short proof that the instance has no solution); or
2. Returns a 3-minimal instance of 3-CSP($\mathbf{A}$) which has the same solution-set (possibly empty) as the original instance.

Let's call this "**enforcing 3-minimality**."

In light of this algorithm, to solve the Central Problem, we need only to consider 3-minimal instances of 3-CSP($\mathbf{A}$).

Example

Let $\mathbf{A} = (\{0,1,2\},\ x{-}y{+}z \pmod 3)$.

Consider the instance $(s_1, C_1),\ (s_2, C_2),\ (s_3, C_3),\ (s_4, C_4)$ of degree 4 where

$$s_1 = \{1,2,3\}, \quad s_2 = \{1,2,4\}, \quad s_3 = \{1,3,4\}, \quad s_4 = \{2,3,4\}$$

and

$$\begin{aligned}
C_1 &= \{(a_1,a_2,a_3) : a_1 + a_2 + a_3 = 0 \pmod 3)\} \\
C_2 &= \{(a_1,a_2,a_4) : a_1 + a_2 + a_4 = 0 \pmod 3)\} \\
C_3 &= \{(a_1,a_3,a_4) : a_1 + a_3 + a_4 = 1 \pmod 3)\} \\
C_4 &= \{(a_2,a_3,a_4) : a_2 + a_3 + a_4 = 1 \pmod 3)\}.
\end{aligned}$$

This is a 3-minimal instance of 3-CSP($\mathbf{A}$).

Quiz: Does it have a solution?

Example (continued)

No solution!

$$
\begin{array}{rrcrcl}
 & a_1 + a_2 + a_3 & = & 0 & \pmod 3 \\
 & a_1 + a_2 + a_4 & = & 0 & \pmod 3 \\
 & a_1 + a_3 + a_4 & = & 1 & \pmod 3 \\
+ & a_2 + a_3 + a_4 & = & 1 & \pmod 3 \\
\hline
 & 0 & = & 2 &
\end{array}
$$

If you didn't notice this, you could always use Gaussian elimination.

G.E. works for **any** instance of 3-CSP$((\mathbb{Z}_3, x{-}y{+}z))$.

Can replace $\mathbb{Z}_3$ with any $(\mathbb{Z}_p, x{-}y{+}z)^d$, p prime, $d \geq 1$.

Theorem (*Nine Chapters*, 2nd century BCE, China)

Let $\mathbf{A} = (G, x{-}y{+}z)$ where G is a finite abelian group of prime exponent. Gaussian elimination solves 3-CSP$(\mathbf{A})$ in poly-time.

Generalizing Gaussian Elimination

Definition

A **ternary group** is an algebra $\mathbf{A} = (G,\ m(x,y,z))$, where $(G,\cdot)$ is a group and $m(x,y,z) = xy^{-1}z$.

Note: in any ternary group $\mathbf{A}$,

- $m(x,y,z)$ is a *Maltsev* operation. (I.e., $m(x,y,y) = x = m(y,y,x)$.)
- The non-empty subuniverses of $\mathbf{A}^n$ are the cosets of subgroups of G^n.

Goal: to solve 3-CSP($\mathbf{A}$) for each finite ternary group $\mathbf{A}$.

Some computational group theory

Definition. Let G be a group.

- A **tower of subgroups for** G is a descending sequence

$$1 = H_m \leq H_{m-1} \leq \cdots \leq H_1 \leq H_0 = G$$

  of subgroups starting at G and ending at the trivial subgroup 1.

- A **complete left transversal system (CLTS)** for this tower is a sequence $(L_1, \ldots, L_m)$ where each L_i is a complete set of representatives of the left cosets of H_i in H_{i-1}.

If $(L_1, \ldots, L_m)$ is a CLTS for this tower, then

$$\begin{aligned} G = H_0 &= L_1 H_1 = L_1(L_2 H_2) = \cdots \\ &= L_1(L_2(\cdots(L_m H_m))) = L_1 L_2 \cdots L_m. \end{aligned}$$

Moreover, every $g \in G$ can be **uniquely expressed** by a product $g_1 g_2 \cdots g_m$ with each $g_i \in L_i$.

Suppose:

- Ω is a finite group (typically of size $2^{O(n)}$), whose
  - elements are represented by strings of length $O(n)$;
  - product and inverse operations are "efficiently computable."
- $K_1, \dots, K_m \leq \Omega$ are subgroups satisfying
  - $\bigcap_{i=1}^{m} K_i = 1$.
  - $[\Omega : K_i] \leq d$ for all i.
  - Membership in each K_i is "efficiently testable."
- $G \leq \Omega$ where G is given to us by a generating set X.

[E.g., $\Omega = S_n$, $m = n$, $K_i = \mathrm{Stab}(i)$, $X \subseteq \Omega$ arbitrary.]

Theorem (Furst, Hopcroft, Luks 1980; Babai 1979 ms; cf. Hoffmann)

Under the above hypotheses, define the tower
$1 = H_m \leq \cdots \leq H_1 \leq H_0 = G$ by

$$H_i = G \cap (K_1 \cap K_2 \cap \cdots \cap K_i).$$

A CLTS for this tower can be computed in poly$(nmd|X|)$ many steps.

Setup

Fix $\mathbf{A} = (G, xy^{-1}z)$, a finite ternary group.

Let $(s_1, C_1), \ldots, (s_p, C_p)$ be an instance of 3-CSP($\mathbf{A}$) of degree n.

Each C_i is a coset of a subgroup of G^{s_i}; say $C_i = \mathbf{a}_i J_i$ where $J_i \leq G^{s_i}$.

For each i let

$$\begin{aligned} K_i &= \{\mathbf{g} \in G^n : \mathrm{proj}_{s_i}(\mathbf{g}) \in J_i\} \leq G^n \\ \mathbf{b}_i &= \text{any element of } G^n \text{ satisfying } \mathrm{proj}_{s_i}(\mathbf{b}_i) = \mathbf{a}_i. \end{aligned}$$

Then

$$[\![s_i, C_i]\!] = \mathbf{b}_i K_i.$$

Hence $\boxed{[\![s_1, C_1]\!] \cap \cdots \cap [\![s_p, C_p]\!] = \mathbf{b}_1 K_1 \cap \cdots \cap \mathbf{b}_p K_p}$ and we want to decide if the intersection is non-empty.

We want to decide whether $\boxed{\mathbf{b}_1 K_1 \cap \cdots \cap \mathbf{b}_p K_p \neq \varnothing}$, where:

For $1 \leq i \leq p$, $\qquad K_i = \{\mathbf{g} \in G^n : \mathrm{proj}_{s_i}(\mathbf{g}) \in J_i\} \leq G^n.$

Define $K_{p+1}, \ldots, K_{p+n}$ by $K_{p+i} := \{\mathbf{g} \in G^n : \mathbf{g}(i) = 1\} \leq G^n.$

Note that:

- Elements of G^n are represented by strings of length n.
- Products and inverses are "efficiently computable."
- $[G^n : K_i] \leq |G|^3$ for all i.
- $\bigcap_i K_i = 1.$ $\quad$ (In fact, $K_{p+1} \cap \ldots \cap K_{p+n} = 1$.)
- Membership in each K_i is "efficiently testable."
- We can find a generating set of G^n of size $O(n)$.

This is the context for the Furst-Hopcroft-Luks result (with $\Omega = G^n$).

Conclusion: we can compute a CLTS $(L_1, \ldots, L_{p+n})$ for the tower

$$1 = H_{p+n} \leq \cdots \leq H_p \leq \cdots \leq H_1 \leq H_0 = G^n$$

in

$$\text{poly}(n \cdot (p+n) \cdot |G|^3 \cdot O(n)) = \text{poly}(n) \ \text{ many steps}$$

where $H_i = K_1 \cap \cdots \cap K_i$.

Question: $\boxed{\text{Is } \mathbf{b}_1 K_1 \cap \cdots \cap \mathbf{b}_p K_p \neq \varnothing?}$

Key observation: If "yes," then

- $\mathbf{b}_1 K_1 \cap \cdots \cap \mathbf{b}_p K_p$ is a left coset of $K_1 \cap \cdots \cap K_p = H_p$.
- There exist (unique) $\mathbf{g}_1 \in L_1$, $\mathbf{g}_2 \in L_2$, ..., $\mathbf{g}_p \in L_p$ such that

$$\mathbf{b}_1 K_1 \cap \cdots \cap \mathbf{b}_p K_p \ = \ \mathbf{g}_1 \mathbf{g}_2 \cdots \mathbf{g}_p H_p.$$

Algorithm

To determine whether there exist $\mathbf{g}_1 \in L_1$, $\mathbf{g}_2 \in L_2$, ..., $\mathbf{g}_p \in L_p$ such that

$$\mathbf{b}_1 K_1 \cap \cdots \cap \mathbf{b}_p K_p \;=\; \mathbf{g}_1 \mathbf{g}_2 \cdots \mathbf{g}_p H_p. \tag{$*$}$$

$(*)$ is equivalent to

$$\begin{aligned} \mathbf{b}_i K_i \;&=\; \mathbf{g}_1 \mathbf{g}_2 \cdots \mathbf{g}_p K_i \qquad \text{for all } i = 1, \ldots, p \\ &=\; \mathbf{g}_1 \mathbf{g}_2 \cdots \mathbf{g}_i K_i \end{aligned} \tag{$**$}$$

(as $j > i$ implies $g_j \in H_{j-1} \subseteq K_i$.)

$(**)$ gives recursive conditions that determine $\mathbf{g}_1, \ldots, \mathbf{g}_p$ (if they exist):

$$\begin{array}{rclcl} \mathbf{b}_1^{-1}\mathbf{g}_1 & \in & K_1 & \quad\text{and}\quad & \mathbf{g}_1 \in L_1 \\ \mathbf{b}_2^{-1}\mathbf{g}_1\mathbf{g}_2 & \in & K_2 & \quad\text{and}\quad & \mathbf{g}_2 \in L_2 \\ & \vdots & & & \\ \mathbf{b}_p^{-1}\mathbf{g}_1\mathbf{g}_2\cdots\mathbf{g}_p & \in & K_p & \quad\text{and}\quad & \mathbf{g}_p \in L_p \end{array}$$

We can quickly determine whether such $\mathbf{g}_1, \ldots, \mathbf{g}_p$ exist.

Thus:

Theorem (Feder, Vardi 1998)

For each finite ternary group $\mathbf{A} = (G, xy^{-1}z)$, the above algorithm:

- Decides whether a given instance of 3-CSP($\mathbf{A}$) has a solution.
- Runs in polynomial time.

Compact generating sets

From a high-level perspective, the Feder-Vardi algorithm discovers:

$$\begin{array}{rclcl} G^n & = & H_0 & = & L_1 L_2 L_3 \cdots L_{p+n} \\ [\![s_1, C_1]\!] & = & \mathbf{g}_1 H_1 & = & \mathbf{g}_1 L_2 L_3 \cdots L_{p+n} \\ [\![s_1, C_1]\!] \cap [\![s_2, C_2]\!] & = & \mathbf{g}_1 \mathbf{g}_2 H_2 & = & \mathbf{g}_1 \mathbf{g}_2 L_3 \cdots L_{p+n} \\ & & \vdots & & \\ [\![s_1, C_1]\!] \cap \cdots \cap [\![s_p, C_p]\!] & = & \mathbf{g}_1 \mathbf{g}_2 \cdots \mathbf{g}_p H_p & = & \mathbf{g}_1 \mathbf{g}_2 \cdots \mathbf{g}_p L_{p+1} \cdots L_{p+n} \end{array}$$

(or detects an empty intersection and halts).

Fact: for each $i \leq p$,

$$\mathbf{g}_1 \mathbf{g}_2 \cdots \mathbf{g}_i (L_{i+1} \cup \cdots \cup L_{p+n})$$

is a generating set of $[\![s_1, C_1]\!] \cap \cdots \cap [\![s_i, C_i]\!]$ (i.e., as a subalgebra of $\mathbf{A}^n$) of size $\leq |p+n| \cdot |G|^3 = O(n^3)$.

The Few Subpowers Algorithm

Abstracting the Feder-Vardi algorithm, a

"poly-time algorithm for 3-CSP(**A**) via compact generating sets"

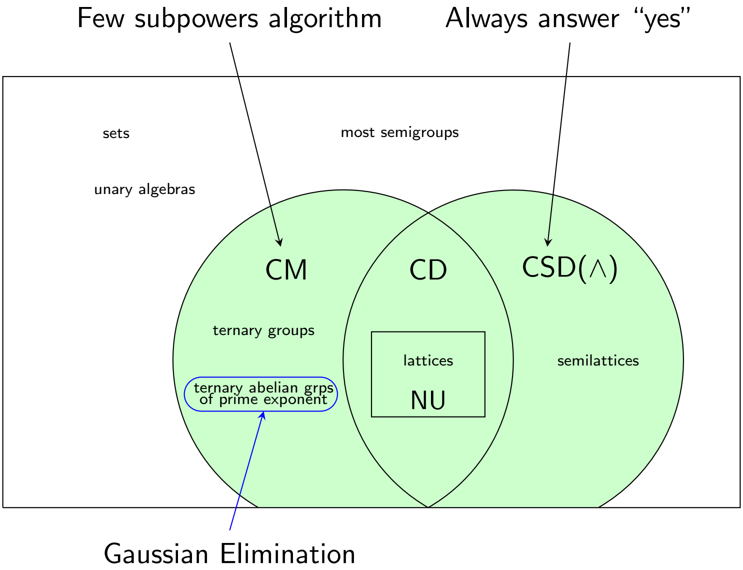
has been worked out for any finite **A** such that **HSP**(**A**) is congruence modular.

Combined work of

- Bulatov, Dalmau 2006, improving Bulatov 2006.
- Idziak *et al* 2010.
- Barto, 2012 ms.

At this level of generality, the algorithm is called the **Few Subpowers Algorithm**.

The 3 algorithms solving 3-minimal instances of 3-CSP(**A**)



Bibliography

L. Babai, Monte Carlo algorithms in graph isomorphism testing, manuscript, 1979.

L. Barto, A proof of the Valeriote conjecture, manuscript, 2012.

A. A. Bulatov, The property of being polynomial for Mal'tsev constraint satisfaction problems, *Algebra i Logika* **45** (2006), 655–686 (Russian).

A. Bulatov and V. Dalmau, A simple algorithm for Mal'tsev constraints, *SIAM J. Comput.* **36** (2006), 16–27.

T. Feder and M. Vardi, The computational structure of monadic SNP and constraint satisfaction: a study through Datalog and group theory, *SIAM J. Comput.* **28** (1998), 57–104.

M. Furst, J. E. Hopcroft, and E. Luks, Polynomial-time algorithms for permutation groups, *FOCS 1980*, IEEE, 36–41.

C. M. Hoffmann, *Group Theoretic Algorithms and Graph Isomorphism*, LNCS **136**, Springer, 1982.

P. Idziak, P. Marković, R. McKenzie, M. Valeriote, and R. Willard, Tractability and learnability arising from algebras with few subpowers, *SIAM J. Comput.* **39** (2010), 3023–3037.