Algebraic Approach to Complexity of Constraint Satisfaction Problems

William DeMeo
williamdemeo@gmail.com

Hawaii Logic Seminar

29 September 2016

What is a CSP?

Informally, a **C**onstraint **S**atisfaction **P**roblem consists of

- a list of variables ranging over a finite domain and
- a set of constraints on those variables.

Problem: can we assign values to all the variables so that all of the constraints are satisfied?

Examples

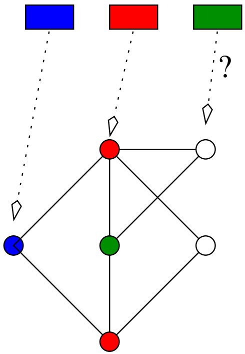
A system of linear equations is a CSP

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1\\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2\\
&\vdots\\
a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m
\end{aligned}$$

Also, a system of nonlinear equations is a CSP

$$\begin{aligned}
a_{11}x_1^2x_3 + a_{12}x_2x_3x_7 + \cdots + a_{1n}x_4x_n^3 &= b_1\\
a_{21}x_2x_5 + a_{22}x_2 \qquad + \cdots + a_{2n}x_4^3 \quad &= b_2\\
&\vdots\\
a_{m1}x_3x_5x_8 + a_{m2}x_2 \qquad + \cdots + a_{mn}x_n \quad &= b_m
\end{aligned}$$

For a fixed k, determining whether a graph is k-colorable is a CSP



Determining whether a given formula $\varphi(x_1, \ldots, x_n)$ is satisfiable is a CSP For example,

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

Determining whether a given formula $\varphi(x_1, \ldots, x_n)$ is satisfiable is a CSP For example,

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

is satisfiable. e.g., $(x, y, z) = (0, 0, 1)$

Algorithms

There is an *efficient* algorithm (Gaussian elimination) for solving any linear system. That is

There is an algorithm that accepts as input a linear system and decides whether that system has a solution.

The running time of the algorithm is bounded by $f(s)$, a *polynomial* in the size s of the system.

Algorithms

There is an *efficient* algorithm (Gaussian elimination) for solving any linear system. That is

There is an algorithm that accepts as input a linear system and decides whether that system has a solution.

The running time of the algorithm is bounded by $f(s)$, a *polynomial* in the size s of the system.

The input, a particular system, is an instance of the problem LINEAR SYSTEM.

Similarly

There is an algorithm that accepts as input a graph and decides whether the graph is 2-colorable.

Running time bounded by $f(s)$, a *polynomial* in the size s of the graph.

The input, a particular graph, is an instance of the problem 2-COLORABILITY.

There is an algorithm that accepts as input a formula, $\varphi = \varphi_1 \wedge \varphi_2 \wedge \cdots \wedge \varphi_k$ (each φ_i bijunctive) and decides whether φ is satisfiable.

Running time bounded by $f(s)$, a *polynomial* in the length s of a string describing φ.

The intput formula φ is an instance of the problem 2-SAT.

We say that all these algorithms run in polynomial time.

No polynomial-time algorithm is known for, NONLINEAR SYSTEM, 3-COLORABILITY, or 3-SAT.

No polynomial-time algorithm is known for, NONLINEAR SYSTEM, 3-COLORABILITY, or 3-SAT.

However, any candidate solution to either of these problems can be checked in polynomial-time.

No polynomial-time algorithm is known for, NONLINEAR SYSTEM, 3-COLORABILITY, or 3-SAT.

However, any candidate solution to either of these problems can be checked in polynomial-time.

Thus these problems are solvable in nondeterministic polynomial time.

Let X and Y be two problems. We write $X \leq_{\mathrm{p}} Y$ to indicate that Y is at least as hard as X.

Let X and Y be two problems. We write $X \leq_{\mathrm{p}} Y$ to indicate that Y is at least as hard as X.

Somewhat more precisely: any algorithm for solving Y can be transformed into an algorithm for X without drastically increasing its running time.

Let X and Y be two problems. We write $X \leq_{\text{p}} Y$ to indicate that Y is at least as hard as X.

Somewhat more precisely: any algorithm for solving Y can be transformed into an algorithm for X without drastically increasing its running time.

It is possible for $X \leq_{\text{p}} Y \leq_{\text{p}} X$. In that case, write $X \equiv_{\text{p}} Y$.

$\mathbb{P}$ is the class of all problems solvable in polynomial time. Its members are called tractable.

$\mathbb{NP}$ is the class of problems solvable in nondeterministic polynomial time.

$\mathbb{P}$ is the class of all problems solvable in polynomial time. Its members are called tractable.

$\mathbb{NP}$ is the class of problems solvable in nondeterministic polynomial time.

- $\mathbb{P} \subseteq \mathbb{NP}$
- Both $\mathbb{P}$ and $\mathbb{NP}$ are downsets, i.e.,
  $Y \in \mathbb{P}$ & $X \leq_{\mathrm{p}} Y \implies X \in \mathbb{P}$

$\mathbb{P}$ is the class of all problems solvable in polynomial time. Its members are called tractable.

$\mathbb{NP}$ is the class of problems solvable in nondeterministic polynomial time.

- $\mathbb{P} \subseteq \mathbb{NP}$
- Both $\mathbb{P}$ and $\mathbb{NP}$ are downsets, i.e., $Y \in \mathbb{P}$ & $X \leq_{\mathrm{p}} Y \implies X \in \mathbb{P}$

The maximal members of $\mathbb{NP}$ are called $\mathbb{NP}$-complete.

3-COLORABILITY, NONLINEAR SYSTEM, and 3-SAT are known to be $\mathbb{NP}$-complete.

$\$2^{20}$ question: $\mathbb{P} \stackrel{?}{=} \mathbb{NP}$.

$\$2^{20}$ question: $\mathbb{P} \stackrel{?}{=} \mathbb{NP}$.

If $\mathbb{P} = \mathbb{NP}$ then all of the above distinctions go away.

Most problems we care about can be solved efficiently. Just build bigger computers.

\$$2^{20}$ question: $\mathbb{P} \stackrel{?}{=} \mathbb{NP}$.

If $\mathbb{P} = \mathbb{NP}$ then all of the above distinctions go away.

Most problems we care about can be solved efficiently. Just build bigger computers.

In particular, this talk becomes pointless. So assume $\mathbb{P} \neq \mathbb{NP}$.

$\$2^{20}$ question: $\mathbb{P} \stackrel{?}{=} \mathbb{NP}$.

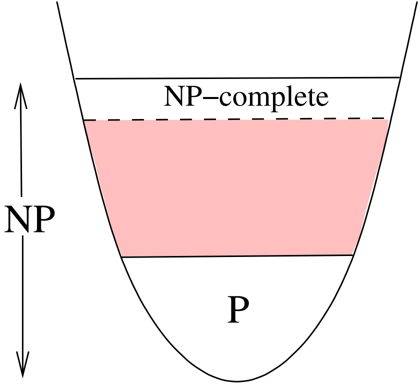
If $\mathbb{P} = \mathbb{NP}$ then all of the above distinctions go away.

Most problems we care about can be solved efficiently. Just build bigger computers.

In particular, this talk becomes pointless. So assume $\mathbb{P} \neq \mathbb{NP}$.

Ladner, 1975

If $\mathbb{P} \neq \mathbb{NP}$, then there are problems in $\mathbb{NP} \setminus \mathbb{P}$ that are not $\mathbb{NP}$-complete.



If $\mathbb{P} \neq \mathbb{NP}$ then the pink area is nonempty.

Oversimplified Definition of CSP

Input

- *variables:* $\mathcal{V} = \{v_1, v_2, \dots\}$
- *domain:* $\mathcal{D}$
- *constraints:* $C_1, C_2, \dots$

Oversimplified Definition of CSP

Input

- *variables:* $\mathcal{V} = \{v_1, v_2, \ldots\}$
- *domain:* $\mathcal{D}$
- *constraints:* $C_1, C_2, \ldots$

Output

- "yes" if there is a *solution* $f : \mathcal{V} \to \mathcal{D}$
  (assigning values to variables and satisfying all C_i)
- "no" otherwise

Slightly more formally...

Let D be a set, n a positive integer

An *n-ary relation on D* is a subset of D^n

Slightly more formally...

Let D be a set, n a positive integer

An *n-ary relation on D* is a subset of D^n

$\mathrm{Rel}_n(D)$ denotes the set of all n-ary relations on D

$\mathrm{Rel}(D) = \bigcup_{n<\omega} \mathrm{Rel}_n(D)$ is the set of all finitary relations on D

Slightly more formal defintion of CSP

Let D be a finite set and $\mathcal{R} \subseteq \mathrm{Rel}(D)$

CSP$(D, \mathcal{R})$ is the following decision problem:

Instance:

- variables: $V = \{ v_1, \dots, v_n \}$ (a finite set)
- constraints: $(C_1, \dots, C_m)$ (a finite list)

Each C_i is a pair $(\mathbf{s}_i,\ R_i)$, where

$$\mathbf{s}_i(j) \in V \quad \text{and} \quad R_i \in \mathcal{R}$$

Slightly more formal defintion of CSP

Let D be a finite set and $\mathcal{R} \subseteq \text{Rel}(D)$

CSP$(D, \mathcal{R})$ is the following decision problem:

Instance:

- variables: $V = \{ v_1, \dots, v_n \}$ (a finite set)
- constraints: $(C_1, \dots, C_m)$ (a finite list)

Each C_i is a pair $(\mathbf{s}_i,\ R_i)$, where

$$\mathbf{s}_i(j) \in V \quad \text{and} \quad R_i \in \mathcal{R}$$

Question: Does there exist a solution?

an assignment $f\colon V \to D$ such that, for all $i \leq m$,

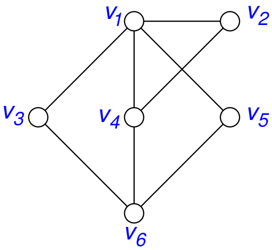
$$(f(\mathbf{s}_i(1)), \dots, f(\mathbf{s}_i(p))) \in R_i$$

Example: 3-colorability

$D = \{r, g, b\}, \quad \mathcal{R} = \{R\}$

$R = \{ (x, y) \in D \times D : x \neq y \}$

Then $\mathrm{CSP}(D, \mathcal{R})$ is the 3-colorability problem



$$
\begin{aligned}
V &= \{v_1, \dots, v_6\} \\
\mathbf{s}_1 &= (v_1, v_2) \\
\mathbf{s}_2 &= (v_2, v_4) \\
&\vdots \\
\mathbf{s}_m &= (v_5, v_6)
\end{aligned}
$$

$R_i = R$ for every i.

Example: 2-SAT

$D = \{0, 1\}$
For a bijunctive clause $\varphi(x, y)$,

$$R_\varphi = \{ \langle a, b\rangle \in D^2 : \varphi(a, b) = 1 \}$$

Example: 2-SAT

$D = \{0, 1\}$
For a bijunctive clause $\varphi(x, y)$,

$$R_\varphi = \{ \langle a, b \rangle \in D^2 : \varphi(a, b) = 1 \}$$

x	y	$x \vee y'$
0	0	1
0	1	0
1	0	1
1	1	1

Example: 2-SAT

$D = \{0, 1\}$
For a bijunctive clause $\varphi(x, y)$,

$$R_\varphi = \{\ \langle a, b\rangle \in D^2 : \varphi(a, b) = 1\ \}$$

x	y	$x \vee y'$
0	0	1
1	0	1
1	1	1

So $R_{x \vee y'} = \{\langle 00\rangle, \langle 10\rangle, \langle 11\rangle\}$

Example: 2-SAT

$D = \{0, 1\}$
For a bijunctive clause $\varphi(x, y)$,

$$R_\varphi = \{\, \langle a, b\rangle \in D^2 : \varphi(a, b) = 1 \,\}$$

x	y	$x \vee y'$
0	0	1
1	0	1
1	1	1

So $R_{x\vee y'} = \{\langle 00\rangle, \langle 10\rangle, \langle 11\rangle\}$

$\mathcal{R} = \{R_{x\vee y}, R_{x\vee y'}, R_{x'\vee y}, R_{x'\vee y'}\}$

Example: 2-SAT

$D = \{0, 1\}$
For a bijunctive clause $\varphi(x, y)$,

$$R_\varphi = \{ \langle a, b\rangle \in D^2 : \varphi(a, b) = 1 \}$$

x	y	$x \vee y'$
0	0	1
1	0	1
1	1	1

So $R_{x\vee y'} = \{\langle 00\rangle, \langle 10\rangle, \langle 11\rangle\}$

$\mathcal{R} = \{R_{x\vee y}, R_{x\vee y'}, R_{x'\vee y}, R_{x'\vee y'}\}$

2-SAT is CSP$(D, \mathcal{R})$

Schaefer's Dichotomy

Schaefer, 1978

Let $D = \{0, 1\}$. There are six families $\mathcal{R}_0, \dots, \mathcal{R}_5$ such that

$$\text{CSP}(D, \mathcal{R}) \in \mathbb{P} \iff \mathcal{R} \subseteq \mathcal{R}_i, \text{ some } i < 6$$

Otherwise $\text{CSP}(D, \mathcal{R})$ is $\mathbb{NP}$-complete.

The six families

$\mathcal{R}_0 = \{\, R : \langle 0, 0, \dots, 0\rangle \in R \,\}$ ("All False")
$\mathcal{R}_1 = \{\, R : \langle 1, 1, \dots, 1\rangle \in R \,\}$ ("All True")
$\mathcal{R}_2 = \{R_{x\vee y},\ R_{x\vee y'},\ R_{x'\vee y},\ R_{x'\vee y'}\}$ (bijunctive)
$\mathcal{R}_3 = \Gamma$ (Horn)
$\mathcal{R}_4 = \Gamma^{\partial}$ (dual-Horn)
$\mathcal{R}_5$ (affine, i.e., linear system over $\mathbb{F}_2$)

Two Motivating Questions

1. Dichotomy Conjecture
   Every CSP$(D, \mathcal{R})$ either lies in $\mathbb{P}$ or is $\mathbb{NP}$-complete.

Two Motivating Questions

1. Dichotomy Conjecture
   Every CSP$(D, \mathcal{R})$ either lies in $\mathbb{P}$ or is $\mathbb{NP}$-complete.
2. Tractability Problem
   Characterize those CSPs that lie in $\mathbb{P}$.

Two Motivating Questions

1. Dichotomy Conjecture
   Every CSP$(D, \mathcal{R})$ either lies in $\mathbb{P}$ or is $\mathbb{NP}$-complete.
2. Tractability Problem
   Characterize those CSPs that lie in $\mathbb{P}$.

What would a characterization look like? What language could we use?

Why is 2-SAT tractable, but 3-SAT is not?

2-SAT: $\mathcal{R}_2 = \{R_{x\vee y},\ R_{x\vee y'},\ R_{x'\vee y},\ R_{x'\vee y'}\}$

3-SAT: $\Lambda = \{\lambda_0, \lambda_1, \dots, \lambda_7\}$

Why is 2-SAT tractable, but 3-SAT is not?

2-SAT: $\mathcal{R}_2 = \{R_{x\vee y},\ R_{x\vee y'},\ R_{x'\vee y},\ R_{x'\vee y'}\}$

3-SAT: $\Lambda = \{\lambda_0, \lambda_1, \dots, \lambda_7\}$

$$M(x, y, z) = \begin{cases} 0 & \text{if at least 2 of } x, y, z \text{ equal } 0 \\ 1 & \text{otherwise} \end{cases}$$

"Majority Operation"

Why is 2-SAT tractable, but 3-SAT is not?

2-SAT: $\mathcal{R}_2 = \{R_{x\vee y},\ R_{x\vee y'},\ R_{x'\vee y},\ R_{x'\vee y'}\}$

3-SAT: $\Lambda = \{\lambda_0, \lambda_1, \dots, \lambda_7\}$

$$M(x,y,z) = \begin{cases} 0 & \text{if at least 2 of } x,y,z \text{ equal } 0 \\ 1 & \text{otherwise} \end{cases}$$

"Majority Operation"

M preserves each $R \in \mathcal{R}_2$:

$$\begin{array}{ccc} \langle a_1, & b_1\rangle & \in R \\ \langle a_2, & b_2\rangle & \in R \\ \langle a_3, & b_3\rangle & \in R \text{ implies} \\ \langle M(a_1,a_2,a_3), & M(b_1,b_2,b_3)\rangle & \in R \end{array}$$

But M fails to preserve each $\lambda \in \Lambda$

For example, with $\lambda = \lambda_{x \vee y \vee z'} = \{0, 1\}^3 - \{\langle 001 \rangle\}$

$$\begin{aligned}
\langle 1, 0, 0 \rangle &\in \lambda \\
\langle 0, 0, 1 \rangle &\in \lambda \\
\langle 0, 1, 1 \rangle &\in \lambda \text{ but} \\
\langle 0, 0, 1 \rangle &\notin \lambda
\end{aligned}$$

Polymorphisms

Definition

Let $R \in \mathrm{Rel}_k(D)$ and $f\colon D^n \to D$. We say f *preserves* R if

$$(a_{11}, \dots, a_{1k}), \dots, (a_{n1}, \dots, a_{nk}) \in R \implies \\ \big(f(a_{11}, \dots, a_{n1}), \dots, f(a_{1k}, \dots, a_{nk})\big) \in R$$

Polymorphisms

Definition

Let $R \in \mathrm{Rel}_k(D)$ and $f\colon D^n \to D$. We say f *preserves* R if

$$(a_{11}, \dots, a_{1k}), \dots, (a_{n1}, \dots, a_{nk}) \in R \implies \\ \big(f(a_{11}, \dots, a_{n1}), \dots, f(a_{1k}, \dots, a_{nk})\big) \in R$$

f is an *n-ary operation* on D.

Polymorphisms

Definition

Let $R \in \mathrm{Rel}_k(D)$ and $f\colon D^n \to D$. We say f *preserves* R if

$$(a_{11}, \dots, a_{1k}), \dots, (a_{n1}, \dots, a_{nk}) \in R \implies (f(a_{11}, \dots, a_{n1}), \dots, f(a_{1k}, \dots, a_{nk})) \in R$$

$$\begin{array}{cccccc}
a_{11} & a_{12} & \dots & a_{1k} & \in & R \\
a_{21} & a_{22} & \dots & a_{2k} & \in & R \\
\vdots & \vdots & & \vdots & & \vdots \\
a_{n1} & a_{n2} & \dots & a_{nk} & \in & R \\
\downarrow f & \downarrow f & & \downarrow f & & \\
\star & \star & \dots & \star & \in & R
\end{array}$$

Definition

Let $\mathcal{R}$ be a set of relations on D.

$\mathrm{Pol}(\mathcal{R})$ is the set of operations preserving all members of $\mathcal{R}$.

These are the polymorphisms of $\mathcal{R}$.

Definition

Let $\mathcal{R}$ be a set of relations on D.

Pol($\mathcal{R}$) is the set of operations preserving all members of $\mathcal{R}$.

These are the polymorphisms of $\mathcal{R}$.

Let F be a set of operations on D.

Inv(F) is the set of relations preserved by all operations in F.

Definition

Let $\mathcal{R}$ be a set of relations on D.

$\mathrm{Pol}(\mathcal{R})$ is the set of operations preserving all members of $\mathcal{R}$.

These are the polymorphisms of $\mathcal{R}$.

Let F be a set of operations on D.

$\mathrm{Inv}(F)$ is the set of relations preserved by all operations in F.

Important point: $(D, \mathrm{Pol}(\mathcal{R}))$ *is an algebraic structure*

Theorem

Let $\mathcal{S}, \mathcal{R} \subseteq \mathsf{Rel}(D)$. Then

$$\mathsf{Pol}(\mathcal{S}) \subseteq \mathsf{Pol}(\mathcal{R}) \implies \mathsf{CSP}(\mathcal{R}) \leq_{\mathsf{p}} \mathsf{CSP}(\mathcal{S}).$$

Theorem

Let $\mathcal{S}, \mathcal{R} \subseteq \mathsf{Rel}(D)$. Then

$$\mathsf{Pol}(\mathcal{S}) \subseteq \mathsf{Pol}(\mathcal{R}) \implies \mathsf{CSP}(\mathcal{R}) \leq_{\mathsf{p}} \mathsf{CSP}(\mathcal{S}).$$

Thus, the richer the algebraic structure, the easier the corresponding CSP

Schaefer proved that on $D = \{0, 1\}$, there are 4 key polymorphisms:

$$M(x, y, z) \text{ (majority)}$$
$$x \wedge y$$
$$x \vee y$$
$$P(x, y, z) = x \oplus y \oplus z = x - y + z$$

$(\{0, 1\}, \mathcal{R})$ is tractable iff one of these is a polymorphism of $\mathcal{R}$

Schaefer proved that on $D = \{0, 1\}$, there are 4 key polymorphisms:

$$M(x, y, z) \text{ (majority)}$$
$$x \wedge y$$
$$x \vee y$$
$$P(x, y, z) = x \oplus y \oplus z = x - y + z$$

$(\{0, 1\}, \mathcal{R})$ is tractable iff one of these is a polymorphism of $\mathcal{R}$

(Un)fortunately, things are more complicated when $|D| > 2$.

Galois Connection

One can go back and forth between relational and algebraic structures

$$\begin{array}{ccc} \textbf{Relational} & & \textbf{Algebraic} \\ (D, \mathcal{R}) & \longrightarrow & (D, \mathrm{Pol}(\mathcal{R})) \\ (D, \mathrm{Inv}(F)) & \longleftarrow & (D, F) \end{array}$$

$$\mathrm{CSP}(D, \mathcal{R}) \equiv_{\mathrm{p}} \mathrm{CSP}(D, \mathrm{Inv}(\mathrm{Pol}(\mathcal{R})))$$

Galois Connection

One can go back and forth between relational and algebraic structures

$$\begin{array}{ccc} \textbf{Relational} & & \textbf{Algebraic} \\ (D, \mathcal{R}) & \longrightarrow & (D, \mathrm{Pol}(\mathcal{R})) \\ (D, \mathrm{Inv}(F)) & \longleftarrow & (D, F) \end{array}$$

$\mathrm{CSP}(D, \mathcal{R}) \equiv_{\mathrm{p}} \mathrm{CSP}(D, \mathrm{Inv}(\mathrm{Pol}(\mathcal{R})))$

Perhaps expressive power of algebra can help classify CSPs.

The Relational Clone

For a set $\mathcal{R}$ of relations on D, let $\langle\mathcal{R}\rangle = \mathrm{Inv}\big(\mathrm{Pol}(\mathcal{R})\big)$.

$\langle\mathcal{R}\rangle$ is called the relational clone generated by $\mathcal{R}$.

It coincides with the set of relations definable from $\mathcal{R}$ by *primitive positive formulas*.

The Relational Clone

For a set $\mathcal{R}$ of relations on D, let $\langle\mathcal{R}\rangle = \mathrm{Inv}\big(\mathrm{Pol}(\mathcal{R})\big)$.

$\langle\mathcal{R}\rangle$ is called the relational clone generated by $\mathcal{R}$.

It coincides with the set of relations definable from $\mathcal{R}$ by *primitive positive formulas*.

$$\varphi(x_1, \dots, x_n) =$$
$$(\exists y_1)(\exists y_2)\cdots(\exists y_m)\big(R_1(z_{1_1}, \dots, z_{1_k}) \wedge \dots \wedge R_t(z_{t_1}, \dots, z_{t_j})\big)$$

Here $R_1 \dots, R_t \in \mathcal{R}$ and every $z_{i_j} \in \{x_1, \dots, x_n, y_1, \dots, y_m\}$

Algebraic Facts

For an algebra $\mathbf{A} = \langle A, F \rangle$ define

$$\mathrm{CSP}(\mathbf{A}) = \mathrm{CSP}(A, \mathrm{Inv}(F))$$

Algebraic Facts

For an algebra $\mathbf{A} = \langle A, F \rangle$ define

$$\mathrm{CSP}(\mathbf{A}) = \mathrm{CSP}(A, \mathrm{Inv}(F))$$

Let $\mathbf{A}$ and $\mathbf{B}$ be algebras

$\mathbf{B}$ a subalgebra of $\mathbf{A} \implies \mathrm{CSP}(\mathbf{B}) \leq_{\mathrm{p}} \mathrm{CSP}(\mathbf{A})$.

$\mathbf{B}$ a homomorphic image of $\mathbf{A} \implies \mathrm{CSP}(\mathbf{B}) \leq_{\mathrm{p}} \mathrm{CSP}(\mathbf{A})$.

$\mathrm{CSP}(\mathbf{A}^n) \equiv_{\mathrm{p}} \mathrm{CSP}(\mathbf{A})$

Bulatov, Jeavons, Krokhin, 2000

If $(D, \mathcal{R})$ is a "core" and every polymorphism is essentially unary, then $\mathrm{CSP}(\mathcal{R})$ is $\mathbb{NP}$-complete.

f is *essentially unary* if $f(x_1, \ldots, x_n) = g(x_j)$ for some unary g and some $j \leq n$.

Bulatov, Jeavons, Krokhin, 2000

If $(D, \mathcal{R})$ is a "core" and every polymorphism is essentially unary, then CSP$(\mathcal{R})$ is $\mathbb{NP}$-complete.

f is *essentially unary* if $f(x_1, \dots, x_n) = g(x_j)$ for some unary g and some $j \leq n$.

Corollary

3-COLORABILITY, NONLINEAR SYSTEM, and 3-SAT are $\mathbb{NP}$-complete.

Informal reformulation of the dichotomy conjecture

If $\mathbf{A}$ has some kind of decent algebraic structure then $\mathrm{CSP}(\mathbf{A}) \in \mathbb{P}$ otherwise $\mathrm{CSP}(\mathbf{A})$ is $\mathbb{NP}$-complete.

Definition

Let $n > 1$. An n-ary operation f is called a *weak near-unanimity operation* if

$$\begin{gathered} f(x, x, \dots, x) = x \text{ and} \\ f(y, x, x, x, \dots, x) = f(x, y, x, x, \dots, x) = \cdots \\ = f(x, x, \dots, x, y) \end{gathered}$$

Note: no essentially unary operation is WNU

Bulatov, Larose, Zádori, McKenzie, Maróti

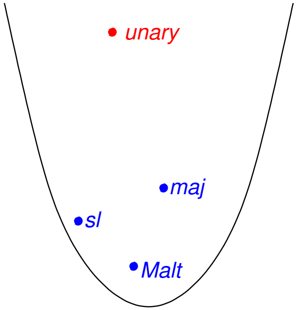
If $\mathcal{R}$ is a core and $\mathrm{Pol}(\mathcal{R})$ has no WNU operation then $\mathrm{CSP}(\mathcal{R})$ is $\mathbb{NP}$-complete.

Reformuated Dichotomy Conjecture

Let $\mathcal{R}$ be a core. Then CSP($\mathcal{R}$) is tractable if and only if it has a WNU polymorphism. Otherwise, it is $\mathbb{NP}$-complete.

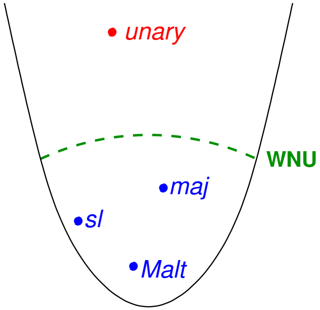
Reformuated Dichotomy Conjecture

Let $\mathcal{R}$ be a core. Then CSP$(\mathcal{R})$ is tractable if and only if it has a WNU polymorphism. Otherwise, it is $\mathbb{NP}$-complete.



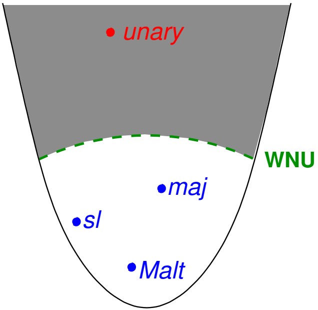
Reformuated Dichotomy Conjecture

Let $\mathcal{R}$ be a core. Then CSP$(\mathcal{R})$ is tractable if and only if it has a WNU polymorphism. Otherwise, it is $\mathbb{NP}$-complete.



Reformuated Dichotomy Conjecture

Let $\mathcal{R}$ be a core. Then CSP($\mathcal{R}$) is tractable if and only if it has a WNU polymorphism. Otherwise, it is $\mathbb{NP}$-complete.



Reformuated Dichotomy Conjecture

Let $\mathcal{R}$ be a core. Then CSP($\mathcal{R}$) is tractable if and only if it has a WNU polymorphism. Otherwise, it is $\mathbb{NP}$-complete.

Supporting Examples

- 2-SAT, 2-COLARABILITY, LINEAR SYSTEM have a WNU.

Reformuated Dichotomy Conjecture

Let $\mathcal{R}$ be a core. Then CSP$(\mathcal{R})$ is tractable if and only if it has a WNU polymorphism. Otherwise, it is $\mathbb{NP}$-complete.

Supporting Examples

- 2-SAT, 2-COLARABILITY, LINEAR SYSTEM have a WNU.
- Let **A** be an abelian group, $n = |A|$. Choose integers k, l with $kl \equiv 1 \pmod{n}$. Then

$$f(x_1, \dots, x_k) = l(x_1 + \cdots + x_k)$$

is a WNU operation.

Two General Techniques for Tractable Algorithms

Method 1

If $\mathrm{Pol}(\mathcal{R})$ contains a “cube operation” then $\mathrm{CSP}(\mathcal{R}) \in \mathbb{P}$

Two General Techniques for Tractable Algorithms

Method 1

If Pol($\mathcal{R}$) contains a "cube operation" then CSP($\mathcal{R}$) $\in \mathbb{P}$

Examples of cube operations:

$P(x, y, z) = x - y + z$
$M(x, y, z) =$ majority

Essentially a generalization of Gaussian elimination.

Algebras with a cube operation possess "few subpowers". This algebraic property is used to prove that the algorithm terminates in polynomial time.

Method 2

If Pol($\mathcal{R}$) contains WNU operations $v(x, y, z)$ and $w(x, y, z, u)$ satisfying $v(y, x, x) = w(y, x, x, x)$, then CSP($\mathcal{R}$) $\in \mathbb{P}$.

Method 2

If $\mathrm{Pol}(\mathcal{R})$ contains WNU operations $v(x, y, z)$ and $w(x, y, z, u)$ satisfying $v(y, x, x) = w(y, x, x, x)$, then $\mathrm{CSP}(\mathcal{R}) \in \mathbb{P}$.

Examples: majority, semilattice

Algebras with these operations have a property called "congruence meet-semidistributivity."

Current State of Affairs

The two general techniques do not cover all cases of a WNU.
What to do next?

Two possible directions:

1. Find a completely new algorithm.
2. Combine the two existing algorithms.

I am exploring both approaches.