Constraint Satisfaction Problems, Graph Theory, and Universal Algebra

Cliff Bergman

April 13, 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

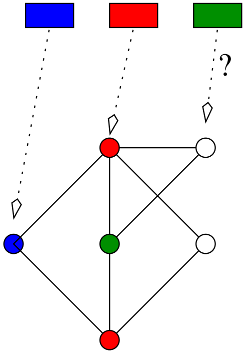
A system of linear equations is a CSP

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

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



Given a propositional formula $\phi(x_1, \ldots, x_n)$, determine whether ϕ is satisfiable

$$\phi(x, y, z) = (x \vee y \vee z') \wedge (x' \vee y \vee z')$$

then

$$\phi(0, 0, 1) = 1$$

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 above by $f(s)$ where f is a polynomial and s is the size 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 above by $f(s)$ where f is a polynomial and s is the size of the system.

The 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.

The running time is bounded by $f(s)$ where f is a polynomial and s is the size of the graph.

The graph is an instance of the problem 2-COLORABILITY.

There is an algorithm that accepts as input a formula, $\phi = \phi_1 \wedge \phi_2 \wedge \cdots \wedge \phi_k$, in which each ϕ_i is bijunctive, and decides whether ϕ is satisfiable.

the running time is bounded by $f(k)$ in which f is a polynomial

ϕ is an instance of the problem 2-SAT.

There is an algorithm that accepts as input a formula, $\phi = \phi_1 \wedge \phi_2 \wedge \cdots \wedge \phi_k$, in which each ϕ_i is bijunctive, and decides whether ϕ is satisfiable.

the running time is bounded by $f(k)$ in which f is a polynomial

ϕ is an instance of the problem 2-SAT.

We say 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 $X \leq_{\mathrm{p}} Y$ 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.

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

$\mathbb{P}$ 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. Almost every problem that mathematicians actually 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. Almost every problem that mathematicians actually care about can be solved efficiently. Just build bigger computers.

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

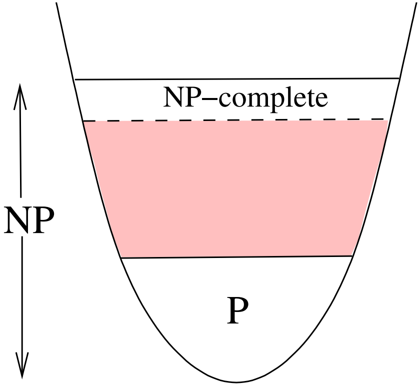
\$$2^{20}$ question: $\mathbb{P} \stackrel{?}{=} \mathbb{NP}$.

If $\mathbb{P} = \mathbb{NP}$ then all of the above distinctions go away. Almost every problem that mathematicians actually care about can be solved efficiently. Just build bigger computers.

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

Theorem (Ladner, 1975)

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



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

Formal Definition of CSP

Let D be a set, n a positive integer
An n-ary relation on D is a subset of D^n

Formal Definition of CSP

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

$$\mathrm{Rel}(D) = \bigcup_{n>0} \mathrm{Rel}_n(D)$$

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

CSP($\langle D, \Delta \rangle$) is the problem:
instance: A finite set $V = \{ v_1, \ldots, v_n \}$ of variables and a finite set $\{ C_1, \ldots, C_m \}$ of constraints

Each constraint C_i is a pair $(\langle x_{i1}, \ldots, x_{i\,p_i} \rangle , \delta_i)$ in which $x_{i1}, \ldots, x_{i\,p_i} \in V$ and $\delta_i \in \Delta$

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

CSP($\langle D, \Delta \rangle$) is the problem:
instance: A finite set $V = \{ v_1, \ldots, v_n \}$ of variables and a finite set $\{ C_1, \ldots, C_m \}$ of constraints

Each constraint C_i is a pair $(\langle x_{i1}, \ldots, x_{i\,p_i} \rangle, \delta_i)$ in which $x_{i1}, \ldots, x_{i\,p_i} \in V$ and $\delta_i \in \Delta$

Question: Does there exist a mapping $f\colon V \to D$ such that for all $i \leq m$, $\langle f(x_{i1}), \ldots, f(x_{i\,p}) \rangle \in \delta_i$?

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

CSP($\langle D, \Delta \rangle$) is the problem:
instance: A finite set $V = \{ v_1, \ldots, v_n \}$ of variables and a finite set $\{ C_1, \ldots, C_m \}$ of constraints

Each constraint C_i is a pair $(\langle x_{i1}, \ldots, x_{i\,p_i} \rangle, \delta_i)$ in which $x_{i1}, \ldots, x_{i\,p_i} \in V$ and $\delta_i \in \Delta$

Question: Does there exist a mapping $f \colon V \to D$ such that for all $i \leq m$, $\langle f(x_{i1}), \ldots, f(x_{i\,p}) \rangle \in \delta_i$?

CSP($\langle D, \Delta \rangle$) always lies in $\mathbb{NP}$.

Example: Linear Equations over $\mathbb{F}_2$

$D = \{0, 1\}$
Δ consists of all relations

$$\delta^b_{n,\mathbf{a}} = \{\, \langle x_1, \ldots, x_n \rangle \in D^n : a_1 x_1 + \cdots a_n x_n = b \}$$

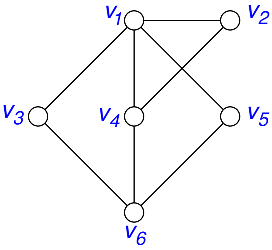
Here, $\mathbf{a} = \langle a_1, a_2, \ldots, a_n \rangle \in D^n$, $b \in D$.

Then CSP($\langle D, \Delta \rangle$) is LINEAR SYSTEM

Example: 3-colorability

$$D = \{r, g, b\}, \qquad \Delta = \{\kappa_3\}$$
$$\kappa_3 = \{\, (x, y) \in D : x \neq y \,\}$$

Then CSP($\langle D, \Delta \rangle$) is the 3-colorability problem



$$V = \{v_1, \ldots, v_6\}$$
$$\langle v_1, v_2 \rangle \in \kappa$$
$$\langle v_1, v_3 \rangle \in \kappa$$
$$\langle v_1, v_4 \rangle \in \kappa$$
$$\langle v_2, v_4 \rangle \in \kappa$$
$$\vdots$$
$$\langle v_5, v_6 \rangle \in \kappa$$

Example: 2-SAT

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

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

Example: 2-SAT

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

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

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 $\phi(x, y)$,

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

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

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

Example: 2-SAT

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

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

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

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

$\Delta = \{\delta_{x\vee y},\ \delta_{x\vee y'},\ \delta_{x'\vee y},\ \delta_{x'\vee y'}\}$

Example: Horn-SAT

Horn formula:

$(x_0 \wedge x_1 \wedge \cdots \wedge x_{i-1} \wedge x_{i+1} \wedge \cdots \wedge x_{n-1}) \rightarrow x_i$

Equivalently:

$x'_0 \vee x'_1 \vee \ldots x'_{i-1} \vee x_i \vee \cdots \vee x'_{n-1}$

Example: Horn-SAT

Horn formula:

$(x_0 \wedge x_1 \wedge \cdots \wedge x_{i-1} \wedge x_{i+1} \wedge \cdots \wedge x_{n-1}) \rightarrow x_i$

Equivalently:

$x_0' \vee x_1' \vee \ldots x_{i-1}' \vee x_i \vee \cdots \vee x_{n-1}'$

Corresponding relation $\gamma_i^n = \{0, 1\}^n - \{\langle 111 \ldots 0 \ldots 111 \rangle\}$

Example: Horn-SAT

Horn formula:

$(x_0 \wedge x_1 \wedge \cdots \wedge x_{i-1} \wedge x_{i+1} \wedge \cdots \wedge x_{n-1}) \to x_i$

Equivalently:

$x'_0 \vee x'_1 \vee \ldots x'_{i-1} \vee x_i \vee \cdots \vee x'_{n-1}$

Corresponding relation $\gamma_i^n = \{0,1\}^n - \{\langle 111 \ldots 0 \ldots 111 \rangle\}$

Horn-SAT is CSP$\langle D, \Gamma \rangle$
$D = \{0,1\}, \quad \Gamma = \{\, \gamma_i^n : 0 \leq i < n \,\}$

Schaefer's Dichotomy

Theorem (Schaefer, 1978)

Let $D = \{0, 1\}$. There are six families $\Delta_0, \ldots, \Delta_5$ such that

$$\mathrm{CSP}(\langle D, \Delta\rangle) \in \mathbb{P} \iff \Delta \subseteq \Delta_i, \textit{ some } i < 6$$

Otherwise $\mathrm{CSP}(\langle D, \Delta\rangle)$ *is* $\mathbb{NP}$*-complete.*

The six families

$\Delta_0 = \{\, \delta : \langle 0, 0, \ldots, 0 \rangle \in \delta \,\}$ ("All False")
$\Delta_1 = \{\, \delta : \langle 1, 1, \ldots, 1 \rangle \in \delta \,\}$ ("All True")
$\Delta_2 = \{\delta_{x \vee y},\ \delta_{x \vee y'},\ \delta_{x' \vee y},\ \delta_{x' \vee y'}\}$ (bijunctive)
$\Delta_3 = \Gamma$ (Horn)
$\Delta_4 = \Gamma^{\partial}$ (dual-Horn)
Δ_5 (affine, i.e., linear system over $\mathbb{F}_2$)

Two Motivating Questions

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

Two Motivating Questions

1. Dichotomy Conjecture
   Every CSP($\langle D, \Delta \rangle$) 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($\langle D, \Delta \rangle$) 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: $\Delta_2 = \{\delta_{x \vee y},\, \delta_{x \vee y'},\, \delta_{x' \vee y},\, \delta_{x' \vee y'}\}$

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

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

2-SAT: $\Delta_2 = \{\delta_{x \vee y},\ \delta_{x \vee y'},\ \delta_{x' \vee y},\ \delta_{x' \vee y'}\}$

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

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

"Majority Operation"

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

2-SAT: $\Delta_2 = \{\delta_{x \vee y},\ \delta_{x \vee y'},\ \delta_{x' \vee y},\ \delta_{x' \vee y'}\}$

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

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

"Majority Operation"

M preserves each $\delta \in \Delta_2$:

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

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

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

Polymorphisms

Definition

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

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

f is an n-ary operation on D.

Polymorphisms

Definition

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

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

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

Definition

Let Δ be a set of relations on D. Then $\mathrm{Pol}(\Delta)$ denotes the set of all operations preserving all members of Δ. These are the polymorphisms of Δ.

Definition

Let Δ be a set of relations on D. Then $\mathrm{Pol}(\Delta)$ denotes the set of all operations preserving all members of Δ. These are the polymorphisms of Δ.

Let F be a set of operations on D. Then $\mathrm{Inv}(F)$ denotes the set of all relations preserved by all operations in F.

Definition

Let Δ be a set of relations on D. Then $\mathrm{Pol}(\Delta)$ denotes the set of all operations preserving all members of Δ. These are the polymorphisms of Δ.

Let F be a set of operations on D. Then $\mathrm{Inv}(F)$ denotes the set of all relations preserved by all operations in F.

Important point: $\langle D, \mathrm{Pol}(\Delta)\rangle$ is an algebraic structure

Theorem

Let $\Gamma, \Delta \subseteq \mathrm{Rel}(D)$*. Then*

$$\mathrm{Pol}(\Gamma) \subseteq \mathrm{Pol}(\Delta) \implies \mathrm{CSP}(\Delta) \leq_{\mathrm{p}} \mathrm{CSP}(\Gamma).$$

Theorem

Let $\Gamma, \Delta \subseteq \mathrm{Rel}(D)$. *Then*

$$\mathrm{Pol}(\Gamma) \subseteq \mathrm{Pol}(\Delta) \implies \mathrm{CSP}(\Delta) \leq_{\mathrm{p}} \mathrm{CSP}(\Gamma).$$

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

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

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

$\langle \{0, 1\}, \Delta \rangle$ is tractable iff one of these four is a polymorphism of Δ

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

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

$\langle \{0, 1\}, \Delta \rangle$ is tractable iff one of these four is a polymorphism of Δ

Unfortunately, things become much more complicated when $|D| > 2$.

One can go back and forth between relational and algebraic structures

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

$$\mathrm{CSP}\langle D, \Delta \rangle \equiv_{\mathrm{p}} \mathrm{CSP}\langle D, \mathrm{Inv}(\mathrm{Pol}(\Delta)) \rangle$$

One can go back and forth between relational and algebraic structures

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

$$\mathrm{CSP}\langle D, \Delta \rangle \equiv_{\mathrm{p}} \mathrm{CSP}\langle D, \mathrm{Inv}(\mathrm{Pol}(\Delta)) \rangle$$

Perhaps the expressive power of algebra can be used to classify CSPs.

For a set of relations, Δ, on D, $\mathrm{Inv}\big(\mathrm{Pol}(\Delta)\big)$ coincides with the set of relations definable from Δ by a positive primitive formula.

For a set of relations, Δ, on D, $\mathrm{Inv}\big(\mathrm{Pol}(\Delta)\big)$ coincides with the set of relations definable from Δ by a positive primitive formula.

$$\phi(x_1, \ldots, x_n) =$$
$$(\exists y_1)(\exists y_2)\cdots(\exists y_m)\big(\delta_1(z_{1_1}, \ldots, z_{1_k}) \wedge \ldots \wedge \delta_t(z_{t_1}, \ldots, z_{t_j})\big)$$

Here $\delta_1 \ldots, \delta_t \in \Delta$ and every $z_{i_j} \in \{x_1, \ldots, x_n, y_1, \ldots, y_m\}$

Algebraic Facts

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

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

Theorem (Bulatov, Jeavons, Krokhin, 2000)

If $\langle D, \Delta \rangle$ is a core and every polymorphism is essentially unary, then CSP(Δ) *is* $\mathbb{NP}$*-complete.*

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

Theorem (Bulatov, Jeavons, Krokhin, 2000)

If $\langle D, \Delta \rangle$ is a core and every polymorphism is essentially unary, then CSP(Δ) *is* $\mathbb{NP}$*-complete.*

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

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 CSP($\mathbf{A}$) $\in \mathbb{P}$ otherwise CSP($\mathbf{A}$) is $\mathbb{NP}$-complete.

Definition

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

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

Note: no essentially unary operation is WNU

Theorem (Bulatov, Larose, Zádori, McKenzie, Maróti)

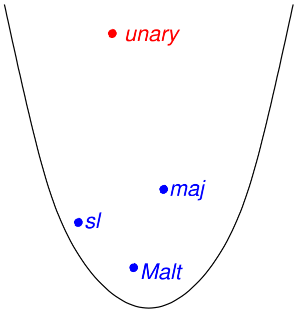
If Δ is a core and $\mathrm{Pol}(\Delta)$ *has no WNU operation then* $\mathrm{CSP}(\Delta)$ *is* $\mathbb{NP}$*-complete.*

Reformuated Dichotomy Conjecture

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

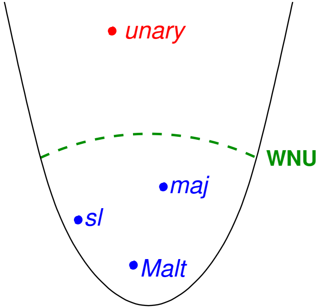
Reformuated Dichotomy Conjecture

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



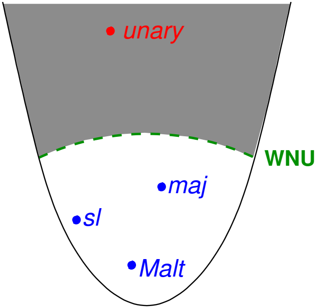
Reformuated Dichotomy Conjecture

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



Reformuated Dichotomy Conjecture

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



Reformuated Dichotomy Conjecture

Let Δ be a core. Then CSP(Δ) 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 Δ be a core. Then CSP(Δ) 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 Pol(Δ) contains a “cube operation” then CSP(Δ) $\in \mathbb{P}$

Two General Techniques for Tractable Algorithms

Method 1

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

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(Δ) 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($\Delta$) $\in \mathbb{P}$.

Method 2

If Pol(Δ) 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($\Delta$) $\in \mathbb{P}$.

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 the second approach.