Testing Assignments to Constraint Satisfaction Problems

Yuichi Yoshida
(NII & PFI)

Joint work with
Hubie Chen (Univ. País Vasco & IKERBASQUE)
Matt Valeriote (McMaster Univ.)
Constraint Satisfaction Problems (CSPs)

Given an instance $\mathcal{I} = (V, C)$:
- $V$: variable set over a finite domain $D$.
- $C$: set of constant-arity constraints.

Find an assignment $f : V \rightarrow D$ that satisfies all the constraints.

Examples:
- $k$-SAT
- $k$-LIN: system of linear equations on $\leq k$ variables over $\mathbb{Z}_q$
- $q$-Coloring
- Unique Games ($x = \pi(y)$ for a bijection $\pi : D \rightarrow D$).
Constraint Satisfaction Problems (CSPs)

We are interested in how the difficulty of the problem changes by varying constraints.

**Definition (CSP)**

A CSP (denoted $\text{CSP}_D(\Gamma)$) is specified by

- finite domain $D = \{1, \ldots, q\}$
- constraint language $\Gamma$: a collection of relations over $D$.
  - relation: a set of $r$-tuples ($r$: arity of $R$)

**Example ($q$-Coloring)**

- $D = \{1, \ldots, q\}$.
- $\Gamma$ has only one relation $R = \{(a, b) \in \{1, \ldots, q\}^2 \mid a \neq b\}$. 
Constraint Satisfaction Problems (CSPs)

Definition (CSP instance of CSP\(_D(\Gamma)\))

An instance (denoted \(I = (V, C)\)) of CSP\(_D(\Gamma)\) is specified by

- a variable set \(V\)
- a set \(C\) of constraints \((R, S)\), where \(R \in \Gamma\); \(S\) is a set of \(ar(R)\) variables.
An instance (denoted $\mathcal{I} = (V, C)$) of CSP\(_D(\Gamma)$) is specified by

- a variable set \( V \)
- a set \( C \) of constraints \((R, S)\), where \( R \in \Gamma \); \( S \) is a set of \( \text{ar}(R) \) variables.

A large number of works on finding satisfying assignments \( f : V \rightarrow D \), i.e., \( f|_S \in R \) for every constraint \((R, S) \in C\).
Testing Assignments to CSPs

Can we decide if an assignment
- satisfies a CSP instance or
- not?

⇒ Yes, in linear time (in $|V| + |C|$).

(Δ)
Testing Assignments to CSPs

Can we decide if an assignment

- satisfies a CSP instance or
- not?

⇒ Yes, in linear time (in $|V| + |C|$).

Can we more quickly test if an assignment

- satisfies a CSP instance or
- is far from satisfying assignments?

⇒ Depends on $\Gamma$, but sometimes we can do even in constant time (independent of $|V|$ and $|C|$).

This work:
A characterization of constant-time testable $\Gamma$. 
Testing Assignments to CSPs

Definition (Testing CSP$_D(\Gamma)$)

Input:
- $\epsilon \in (0, 1)$
- a (satisfiable) instance $\mathcal{I} = (V, C)$ of CSP$_D(\Gamma)$
- weight function $w : V \to \mathbb{R}$ with $\sum_{v \in V} w(v) = 1$.
- a query access to an assignment $f : V \to D$.

Output:
- **Yes** w.p. $\geq 2/3$ if $f$ satisfies $\mathcal{I}$.
- **No** w.p. $\geq 2/3$ if $f$ is $\epsilon$-far from satisfying $\mathcal{I}$, i.e.,

\[
dist(f, g) := \sum \{ w(v) \mid v \in V, f(v) \neq g(v) \} > \epsilon
\]

for any satisfying assignment $g$ of $\mathcal{I}$. 
Known Results

How does $\Gamma$ affect the worst-case query complexity?

<table>
<thead>
<tr>
<th>CSP</th>
<th>Query complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-Coloring</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>2-SAT</td>
<td>$\Omega\left(\frac{\log n}{\log \log n}\right), O\left(\sqrt{n}\right)$ [FLN+02]</td>
</tr>
<tr>
<td>3-Coloring, 3-SAT, 3-LIN</td>
<td>$\Omega(n)$ [BSHR05]</td>
</tr>
<tr>
<td>Horn 3-SAT</td>
<td>$\Omega(n)$ [BY13]</td>
</tr>
</tbody>
</table>

- The **algebra** associated with a CSP determines its query complexity [Yos14].
- **Trichotomy for Boolean CSPs** [BY13]:
  - Constant-query testable.
  - Not constant-query testable, but sublinear-query testable.
  - Not sublinear-query testable.
Main Result

Theorem (Dichotomy for general CSPs)

There exists an algebraic condition $\mathcal{A}$ such that

- If $\Gamma$ satisfies $\mathcal{A}$, then $\text{CSP}_D(\Gamma)$ is constant-query testable.
- Otherwise, $\text{CSP}_D(\Gamma)$ is not constant-query testable.
Main Result

Theorem (Dichotomy for general CSPs)

There exists an algebraic condition $\mathcal{A}$ such that

- If $\Gamma$ satisfies $\mathcal{A}$, then $\text{CSP}_D(\Gamma)$ is constant-query testable.
- Otherwise, $\text{CSP}_D(\Gamma)$ is not constant-query testable.

What’s algebra?
Polymorphism

Definition (Polymorphism)

A function $f : D^k \rightarrow D$ is called a polymorphism of $\Gamma$ if for any $R \in \Gamma$ of arity $r$,

\[
(a_1^1, \ldots, a_r^1) \in R \\
\vdots \\
(a_1^k, \ldots, a_r^k) \in R \\
\downarrow f \\
(b_1, \ldots, b_r) \in R
\]
Polymorphism

Definition (Polymorphism)

A function \( f : D^k \rightarrow D \) is called a polymorphism of \( \Gamma \) if for any \( R \in \Gamma \) of arity \( r \),

\[
\begin{align*}
(a_1^1, \ldots, a_r^1) &\in R \\
\vdots \\
(a_1^k, \ldots, a_r^k) &\in R \\
\downarrow f \\
(b_1, \ldots, b_r) &\in R
\end{align*}
\]

\( \text{Pol}(\Gamma) \): the set of polymorphisms of \( \Gamma \). 
\( (D, \text{Pol}(\Gamma)) \) is the algebra associated with \( \Gamma \).
Polymorphism

Example

- **min** is a polymorphism of Horn $k$-SAT for any $k$. Consider $R = \{(u, v, w) \mid u \land v \Rightarrow w\}$.

  \[
  (1, 0, 0) \in R \\
  (0, 1, 0) \in R \\
  \downarrow \text{min} \\
  (0, 0, 0) \in R
  \]

- *(ternary) majority* is a polymorphism of 2-SAT.
- $x + y + z \pmod{2}$ is a polymorphism of 3-LIN.
- The only polymorphism of 3-SAT is $f(x) = x_i$ (projection).
Polymorphisms Determine Query Complexity

To study query complexity of $\text{CSP}(\Gamma)$, we only have to look at polymorphisms!

**Theorem ([Yos14])**

Let $\Gamma$ and $\Gamma'$ be constraint languages with $\text{Pol}(\Gamma) = \text{Pol}(\Gamma')$. If $\text{CSP}(\Gamma)$ is constant-query testable, then $\text{CSP}(\Gamma')$ is constant-query testable.
Main Result

Theorem (Dichotomy for general CSPs)

The following holds:

(i) If $\text{Pol}(\Gamma)$ has a majority and a Maltsev operation (arithmetic), then $\text{CSP}(\Gamma)$ is constant-query testable.

(ii) Otherwise, $\text{CSP}(\Gamma)$ is not constant-query testable.

majority $m : D^3 \to D$: $m(b, a, a) = m(a, b, a) = m(a, a, b) = a$.

Maltsev $p : D^3 \to D$: $p(b, a, a) = p(a, a, b) = b$. 
Main Result

Theorem (Dichotomy for general CSPs)

The following holds:

(i) If $\text{Pol}(\Gamma)$ has a majority and a Maltsev operation \textit{(arithmetic)}, then $\text{CSP}(\Gamma)$ is constant-query testable.

(ii) Otherwise, $\text{CSP}(\Gamma)$ is not constant-query testable.

**majority** $m : D^3 \to D$: $m(b, a, a) = m(a, b, a) = m(a, a, b) = a$.

**Maltsev** $p : D^3 \to D$: $p(b, a, a) = p(a, a, b) = b$.

We only look at (i) as (ii) is obtained by a simple modification of [FLN+02].
Arithmetic CSPs

Example

- 2-Coloring
- Unique Games
- Modular arithmetic:
  \[ D = \{0, 1, \ldots, 29\} \]
  Relations:
  - \( x \equiv y \pmod{p} \) for \( p \in \{2, 3, 5\} \).
  - \( x \equiv a \pmod{p} \) for \( p \in \{2, 3, 5\} \) and \( a \in \{0, 1, \ldots, p - 1\} \).
- An example derived from finite Heyting algebra...
The idea is transforming the given input \((\mathcal{I}, f)\) to a trivial one by a sequence of reductions.

- Factoring reduction
- Splitting reduction
- Isomorphism reduction
Constant-Query Tester for Arithmetic CSPs

The idea is transforming the given input \((\mathcal{I}, f)\) to a trivial one by a sequence of reductions.

- Factoring reduction
- Splitting reduction
- Isomorphism reduction

We look at these reductions for modular arithmetic. It is convenient to identify the domain \(\{0, 1, \ldots, 29\}\) with \(F_2 \times F_3 \times F_5\).
Factoring Reduction

Shrink the domain of each variable by factoring by an irrelevant congruence:

\[ f(x) = (0, 2, 3) \]
\[ \in \mathbb{F}_2 \times \mathbb{F}_3 \times \mathbb{F}_5 \]

\[ x \equiv y \mod 2 \quad x \equiv z \mod 3 \]

\[ f(x) = (0, 2) \]
\[ \in \mathbb{F}_2 \times \mathbb{F}_3 \]

\[ x \equiv y \mod 2 \quad x \equiv z \mod 3 \]
Splitting Reduction

Split variables (Chinese remainder theorem):

\[ f(x) = (0, 2) \]
\[ \in \mathbb{F}_2 \times \mathbb{F}_3 \]

![Diagram showing the connection between variables y, x, and z]

- \( f(x') = 0 \)
  \[ \in \mathbb{F}_2 \]
- \( f(x'') = 2 \)
  \[ \in \mathbb{F}_3 \]
Isomorphism reduction

Relations in each connected component are isomorphisms. Test the consistency of $f$ within each connected component. If the test passes, contract the connected components.

$$f(x) = 0 \in \mathbb{F}_2$$

$$f(y) = 2 \in \mathbb{F}_3$$

Trivial instance!
The details are complicated:

- We need to preprocess $I$.
- $\epsilon$-farness should be also preserved.
- We should query $f$ on the fly.
- In isomorphism reduction, relations may be just surjective homomorphisms.
- We need to perform these reductions $|D|$ times.

The arithmetic condition is used to guarantee that a factoring reduction followed by a factoring, a splitting, and an isomorphism reduction always shrinks the domain.

The resulting query complexity: $2^{O(|D|)} / \epsilon^2$. 
Conclusions

Dichotomy for testing assignments to CSPs:

Majority + Maltsev ⇔ Constant-query testability.

Also, we achieved a trichotomy for testing ∃CSPs with one-sided
error (∃CSP = a CSP with existentially quantified variables).
Conclusions

Dichotomy for testing assignments to CSPs:

\[
\text{Majority + Maltsev} \iff \text{Constant-query testability.}
\]

Also, we achieved a trichotomy for testing \(\exists\)CSPs with one-sided error (\(\exists\)CSP = a CSP with existentially quantified variables).

Conjecture

\(\text{CSP}_D(\Gamma)\) is sublinear-query testable if and only if \(\text{Pol}(\Gamma)\) has a \(k\)-near-unanimity operation for some \(k \geq 3\).

\(k\)-ary near unanimity \(n : D^k \to D:\)

\[
n(b, a, a, \ldots, a) = n(a, b, a, \ldots, a) = \cdots = n(a, a, \ldots, a, b) = a.
\]

The if direction is true.