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 $\mathcal{I} = (V, \mathcal{C})$:

- V: variable set over a finite domain D.
- $\mathcal{C}$: set of constant-arity constraints.

Find an assignment $f : V \to D$ 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 $\pi : D \to D$).

Constraint Satisfaction Problems (CSPs)

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

Definition (CSP)

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

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

Example (q-Coloring)

- $D = \{1, \ldots, q\}$.
- Γ has only one relation $R = \left\{(a, b) \in \{1, \ldots, q\}^2 \mid a \neq b\right\}$.

Constraint Satisfaction Problems (CSPs)

Definition (CSP instance of $\mathrm{CSP}_D(\Gamma)$)

An instance (denoted $\mathcal{I} = (V, \mathcal{C})$) of $\mathrm{CSP}_D(\Gamma)$ is specified by

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

Constraint Satisfaction Problems (CSPs)

Definition (CSP instance of $\mathrm{CSP}_D(\Gamma)$)

An instance (denoted $\mathcal{I} = (V, \mathcal{C})$) of $\mathrm{CSP}_D(\Gamma)$ is specified by

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

A large number of works on finding satisfying assignments $f : V \to D$, i.e., $f|_S \in R$ for every constraint $(R, S) \in \mathcal{C}$.

Testing Assignments to CSPs

Can we decide if an assignment

- satisfies a CSP instance or
- not?

⇒ Yes, in linear time (in $|V| + |\mathcal{C}|$).

Testing Assignments to CSPs

Can we decide if an assignment

- satisfies a CSP instance or
- not?

⇒ Yes, in linear time (in $|V| + |\mathcal{C}|$).

Can we more quickly test if an assignment

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

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

This work:
A characterization of constant-time testable Γ.

Testing Assignments to CSPs

Definition (Testing $\mathrm{CSP}_D(\Gamma)$)

Input:

- $\epsilon \in (0, 1)$
- a (satisfiable) instance $\mathcal{I} = (V, \mathcal{C})$ of $\mathrm{CSP}_D(\Gamma)$
- 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 ϵ-far from satisfying $\mathcal{I}$, i.e.,

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

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

Known Results

How does Γ affect the worst-case query complexity?

CSP	Query complexity
2-Coloring	$O(1)$
2-SAT	$\Omega(\frac{\log n}{\log \log n})$, $O(\sqrt{n})$ [FLN+02]
3-Coloring, 3-SAT, 3-LIN	$\Omega(n)$ [BSHR05]
Horn 3-SAT	$\Omega(n)$ [BY13]

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

Main Result

Theorem (Dichotomy for general CSPs)

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

- *If Γ satisfies $\mathscr{A}$, then $\mathrm{CSP}_D(\Gamma)$ is constant-query testable.*
- *Otherwise, $\mathrm{CSP}_D(\Gamma)$ is not constant-query testable.*

What's algebra?

Polymorphism

Definition (Polymorphism)

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

$$
\begin{array}{c}
(a_1^1, \ldots, a_r^1) \in R \\
\vdots \\
(a_1^k, \ldots, a_r^k) \in R \\
\downarrow f \\
(b_1, \ldots, b_r) \in R
\end{array}
$$

Polymorphism

Definition (Polymorphism)

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

$$
\begin{array}{c}
(a_1^1, \ldots, a_r^1) \in R \\
\vdots \\
(a_1^k, \ldots, a_r^k) \in R \\
\downarrow f \\
(b_1, \ldots, b_r) \in R
\end{array}
$$

$\mathrm{Pol}(\Gamma)$: the set of polymorphisms of Γ.
$(D, \mathrm{Pol}(\Gamma))$ is the algebra associated with Γ.

Polymorphism

Example

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

$$\begin{gathered}(1,0,0) \in R \\ (0,1,0) \in R \\ \downarrow \text{min} \\ (0,0,0) \in R\end{gathered}$$

- (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 CSP(Γ), we only have to look at polymorphisms!

Theorem ([Yos14])

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

Main Result

Theorem (Dichotomy for general CSPs)

The following holds:

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

(ii) *Otherwise,* $\mathrm{CSP}(\Gamma)$ *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* $\mathrm{Pol}(\Gamma)$ *has a majority and a Maltsev operation (arithmetic), then* $\mathrm{CSP}(\Gamma)$ *is constant-query testable.*

(ii) *Otherwise,* $\mathrm{CSP}(\Gamma)$ *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...

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

Constant-Query Tester for Arithmetic CSPs

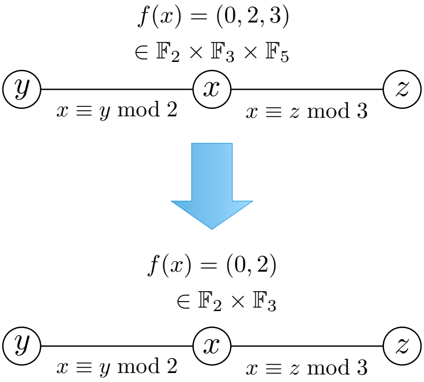
The idea is transforming the given input $(\mathcal{I}, f)$ to a trivial one by a sequence of reductions.

- 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 $\mathbb{F}_2 \times \mathbb{F}_3 \times \mathbb{F}_5$.

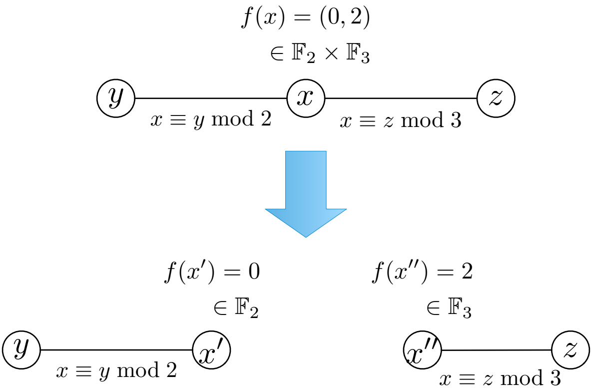
Factoring Reduction

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



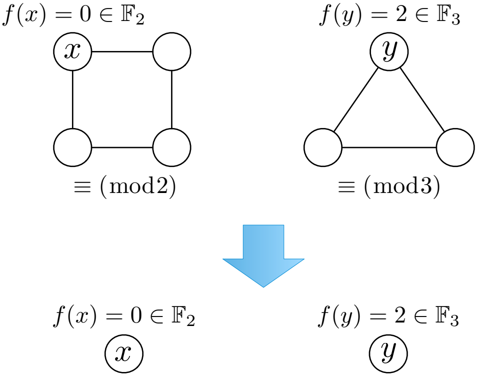
Splitting Reduction

Split variables (Chinese remainder theorem):



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.



Trivial instance!

Constant-Query Tester for Arithmetic CSPs

The details are complicated:

- We need to preprocess $\mathcal{I}$.
- ϵ-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:

$$\text{Majority + Maltsev} \Leftrightarrow \text{Constant-query testability.}$$

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:

Majority + Maltsev $\Leftrightarrow$ Constant-query testability.

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

Conjecture

$\mathrm{CSP}_D(\Gamma)$ is sublinear-query testable if and only if $\mathrm{Pol}(\Gamma)$ 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.