2016 IEEE 57th Annual Symposium on Foundations of Computer Science

Testing Assignments to Constraint Satisfaction Problems

Hubie Chen
University of the Basque Country
and
IKERBASQUE, Basque Foundation for Science
hubie.chen@ehu.es

Matt Valeriote
McMaster University
matt@math.mcmaster.ca

Yuichi Yoshida
National Institute of Informatics
and
Preferred Infrastructure, Inc.
yyoshida@nii.ac.jp

***Abstract*—For a finite relational structure A, let CSP(A) denote the CSP instances whose constraint relations are taken from A. The resulting family of problems CSP(A) has been considered heavily in a variety of computational contexts. In this article, we consider this family from the perspective of property testing: given an instance of a CSP and query access to an assignment, one wants to decide whether the assignment satisfies the instance, or is far from so doing. While previous work on this scenario studied concrete templates or restricted classes of structures, this article presents comprehensive classification theorems.**

Our first contribution is a dichotomy theorem completely characterizing the structures A such that CSP(A) is constant-query testable: (i) If A has a majority polymorphism and a Maltsev polymorphism, then CSP(A) is constant-query testable with one-sided error. (ii) Else, testing CSP(A) requires a super-constant number of queries.

Let ∃CSP(A) denote the extension of CSP(A) to instances which may include existentially quantified variables. Our second contribution is to classify all structures A in terms of the number of queries needed to test assignments to instances of ∃CSP(A), with one-sided error. More specifically, we show the following trichotomy (i) If A has a majority polymorphism and a Maltsev polymorphism, then ∃CSP(A) is constant-query testable with one-sided error. (ii) Else, if A has a $(k+1)$-ary near-unanimity polymorphism for some $k \geq 2$, and no Maltsev polymorphism then ∃CSP(A) is not constant-query testable (even with two-sided error) but is sublinear-query testable with one-sided error. (iii) Else, testing ∃CSP(A) with one-sided error requires a linear number of queries.

***Keywords*-Constraint Satisfaction Problems; Property Testing;**

I. Introduction

A. Background

In property testing, the goal is to design algorithms that distinguish objects satisfying some predetermined property P from objects that are far from satisfying P. More specifically, for $\varepsilon, \delta > 0$, an algorithm is called an (ε, δ)-*tester* for a property P, if given an input I, it accepts with probability at least $1-\delta$ if the input satisfies P, and it rejects with probability at least $1-\delta$ if the input I is ε-far from satisfying P. Roughly speaking, we say that I is ε-*far* from P if we must modify at least an ε-fraction of I to make I satisfy P. When $\delta = 1/3$, we simply call it an ε-*tester*. A tester is called a *one-sided error tester* if it always accepts when I satisfies P. In contrast, a standard tester is sometimes called a *two-sided error tester*. As one motivation of property testing is to design algorithms that run in time sublinear in the input size, we assume query access to the input, and we measure the efficiency of a tester by its *query complexity*. We refer to [16], [24], [25] for surveys on property testing.

In *constraint satisfaction problems* (for short, CSP*s*), one is given a set of variables and a set of constraints imposed on the variables, and the task is to find an assignment of the variables that satisfies all of the given constraints. By restricting the relations used to specify constraints, it is known that certain restricted versions of the CSP coincide with many fundamental problems such as SAT, graph coloring, and solvability of systems of linear equations. To formally define these restricted versions of the CSP (and hence, these problems), we consider *relational structures* $\mathbf{A} = (A;\Gamma)$, where A is a finite set and Γ consists of a finite set of finitary relations over A. In this context, Γ is sometimes referred to as a *constraint language* over A and $\mathbf{A}$ as a *template*. Then, we define CSP($\mathbf{A}$) to be those instances of the CSP whose constraint relations are taken from Γ. In recent years, computational aspects of CSP($\mathbf{A}$) have been heavily studied, in the decision setting [19], [9], [2], [5], in counting complexity [10], [14], in computational learning theory [19], [13], and in optimization and approximation [23], [26], [12], [27], [28]. See also the survey by Barto [4] for an overview of this line of research.

In this paper, we consider the problem family CSP($\mathbf{A}$) from the perspective of property testing, in particular, we consider the task of testing assignments to CSPs. Relative to a relational structure $\mathbf{A}$, an input consists of a tuple $(\mathscr{I}, \varepsilon, f)$, where $\mathscr{I}$ is an instance of CSP($\mathbf{A}$) with weights on the variables, ε is an error parameter, and f is an assignment to $\mathscr{I}$. In the studied model, the tester has full access to $\mathscr{I}$ and query access to f, that is, a variable x can be queried to obtain the value of $f(x)$. In this sense, assignment testing lies in the *massively parameterized model* [22]. We say that f is ε-*far* from satisfying $\mathscr{I}$ if one must modify at least an ε-fraction of f (with respect to the weights) to make f a satisfying assignment of $\mathscr{I}$, and we say that f is ε-*close* otherwise. It is always assumed that $\mathscr{I}$ has a satisfying assignment as otherwise we can immediately reject the input (in this context, one does not care about time complexity). The objective of assignment testing of CSPs is to correctly decide whether f is a satisfying assignment of $\mathscr{I}$ or is ε-far from being so with probability at least $2/3$. When f does not satisfy $\mathscr{I}$ but is ε-close to satisfying $\mathscr{I}$, we can output anything.

In assignment testing, we say that the query complexity of a tester is constant/sublinear/linear if it is constant/sublinear/linear in the number of variables of an instance. The main problem addressed in this paper is to reveal the relationship between a relational structure $\mathbf{A}$ and the number of queries needed to test $\mathrm{CSP}(\mathbf{A})$ and a related problem class $\exists\mathrm{CSP}(\mathbf{A})$.

B. Contributions

While previous works on testing assignments to the problems $\mathrm{CSP}(\mathbf{A})$ studied concrete templates $\mathbf{A}$ or restricted classes of structures, this article presents comprehensive classification theorems.

The first contribution of this paper is a dichotomy theorem that *completely characterizes* the constant-query testable CSPs. Before describing our characterization, we introduce the algebraic notion of a polymorphism which is key to the description and obtention of our results. Let R be an r-ary relation on a set A. A (k-ary) operation $f : A^k \to A$ is said to be a *polymorphism* of R (or R is *preserved* by f) if for any set of k r-tuples $(a_1^1,\dots,a_r^1),(a_1^2,\dots,a_r^2),\dots,(a_1^k,\dots,a_r^k) \in R$, the tuple $(f(a_1^1,\dots,a_1^k),\dots,f(a_r^1,\dots,a_r^k))$ also belongs to R. An operation f is a *polymorphism* of a relational structure $\mathbf{A}$ if it is a polymorphism of each of its relations. We define the *algebra* of $\mathbf{A}$, denoted by $\mathrm{Alg}(\mathbf{A})$, to be the pair $(A;\mathrm{Pol}(\mathbf{A}))$, where $\mathrm{Pol}(\mathbf{A})$ is the set of all polymorphisms of $\mathbf{A}$.

Definition I.1. *Let A be a nonempty set. A* majority operation *on A is a ternary operation $m : A^3 \to A$ such that $m(b,a,a) = m(a,b,a) = m(a,a,b) = a$ for all a, $b \in A$. A* Maltsev operation *on A is a ternary operation $p : A^3 \to A$ such that $p(b,a,a) = p(a,a,b) = b$ for all a, $b \in A$. For $k \geq 2$, an operation $n : A^{(k+1)} \to A$ is a $(k+1)$-ary* near unanimity operation *on A if for all a, $b \in A$, $n(b,a,a,\dots,a) = n(a,b,a,\dots,a) = \cdots = n(a,a,\dots,a,b) = a$. (Note that a majority operation is a 3-ary near-unanimity operation.)*

Theorem I.2. *Let $\mathbf{A}$ be a relational structure. The following dichotomy holds.*

1. *If $\mathbf{A}$ has a majority polymorphism and a Maltsev polymorphism, then $\mathrm{CSP}(\mathbf{A})$ is constant-query testable (with one-sided error).*
2. *Else, testing $\mathrm{CSP}(\mathbf{A})$ requires a super-constant number of queries.*

This theorem generalizes characterizations of constant-query testable List H-homomorphisms [30] and Boolean CSPs [7] to general CSPs. In Section III we will describe the particularly nice structure of relations over templates that have majority and Maltsev polymorphisms and use this to prove the theorem. For the moment, let us consider a number of example templates to which the positive result of this theorem applies.

Example I.3. The template $\mathbf{A}$ over the Boolean domain $\{0,1\}$ whose only relation is $\neq$ has both majority and Maltsev polymorphisms. Note that $\mathrm{CSP}(\mathbf{A})$ coincides with the graph 2-coloring problem.

More generally, the template $\mathbf{A}$ over a finite domain where each relation is a bijection on A has both majority and Maltsev polymorphisms, and instances of $\mathrm{CSP}(\mathbf{A})$ for such templates $\mathbf{A}$ coincide with instances of the problem which is the subject of the *unique games conjecture* [20]. □

Example I.4. Another class of finite structures that have both majority and Maltsev polymorphisms are those that have a *discriminator* operation as a polymorphism. On a set A the discriminator operation $d_A(x,y,z)$ is the operation such that if $x = y$ then $d(x,y,z) = z$ and if $x \neq y$, $d(x,y,z) = x$. From this definition, it is immediate that d_A is a Maltsev operation on A, and that $d(x,d(x,y,z),z)$ is a majority operation on A. Any finite product of finite fields will have a discriminator term operation ([11]) and so any finite relational structure whose relations are compatible with the operations of such a ring will have majority and Maltsev polymorphism. □

Example I.5. For p a prime number, let $\mathbb{F}_p$ be the field of size p, and let $\mathbb{R}$ be the ring $\mathbb{F}_2 \times \mathbb{F}_3 \times \mathbb{F}_5$. Then as noted in Example I.4, $\mathbb{R}$ has a discriminator term operation. Let $\mathbf{R}$ be the structure with domain R and set of relations Γ consisting of intersections of the following binary relations on R: For $p = 2$, 3, or 5,

- $C_p = \{((a_2,a_3,a_5),(b_2,b_3,b_5)) \mid a_p = b_p\}$,
- For $a \in \mathbb{F}_p$, $E_a = \{((a_2,a_3,a_5),(b_2,b_3,b_5)) \mid a_p = a\}$,
- For $b \in \mathbb{F}_p$, $E_b = \{((a_2,a_3,a_5),(b_2,b_3,b_5)) \mid b_p = b\}$,

So relations in Γ can express that pairs of elements in R are congruent modulo 2, 3, or 5 in the corresponding coordinate and/or that a certain coordinate is equal to some fixed value. These relations are invariant under the discriminator term operation of $\mathbb{R}$ and so according to Theorem I.2, $\mathrm{CSP}(\mathbf{R})$ has constant query complexity. □

Examples of structures that satisfy the first condition of Theorem I.2 but that do not have a discriminator operation as a polymorphism can be derived from finite Heyting algebras.

Example I.6. Consider the five-element Heyting algebra $\mathbb{M}$ presented in [18, Figure 1]. (Heyting algebras are bounded distributive lattices that also have a binary "implication" operation; they serve as algebraic models of propositional intuitionistic logic.) This algebra has universe $M = \{0,a,b,e,1\}$; the two equivalence relations α and β that partition M into blocks $\{\{0,a\},\{b,e,1\}\}$ and $\{\{0,b\},\{a,e,1\}\}$ (respectively) are preserved by the operations of the algebra. Since $\mathbb{M}$ has majority and Maltsev term operations (the operations

$(x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$ and $(x \to y) \to z) \wedge (z \to y) \to x)$ respectively), then the structure $\mathbf{M} = (M; \alpha, \beta)$ has majority and Maltsev polymorphisms. The only other non-trivial binary relation on M that is definable by a primitive-positive formula over $\mathbf{M}$ is $\alpha \cap \beta$. $\square$

Bulatov and Marx provide yet another example of a structure having both a majority and a Maltsev polymorphism, in [8, Example 1.1].

We next consider existentially quantified CSPs (∃CSPs for short). The difference between CSPs and ∃CSPs is that, in an instance of ∃CSP, existentially quantified variables may appear. So, an instance of ∃CSP may be defined as a primitive positive formula (pp-formula) over a relational structure. Primitive positive formulas are known as *conjunctive queries* in the database theory literature; they are arguably the most heavily studied class of database queries, and the problem ∃CSP can be associated with the problem of *conjunctive query evaluation*.

For a relational structure $\mathbf{A} = (A; \Gamma)$, we define $\exists\mathrm{CSP}(\mathbf{A})$ to be the collection of instances of ∃CSP whose constraint relations are taken from Γ. Our second contribution is to provide a complete classification of all structures $\mathbf{A}$ in terms of the number of queries needed to test assignments of instances of $\exists\mathrm{CSP}(\mathbf{A})$ with one-sided error:

Theorem I.7. *Let $\mathbf{A}$ be a relational structure. Then, the following trichotomy holds.*

(1) *If $\mathbf{A}$ has a majority polymorphism and a Maltsev polymorphism, then $\exists\mathrm{CSP}(\mathbf{A})$ is constant-query testable with one-sided error.*
(2) *Else, if $\mathbf{A}$ has a $(k+1)$-ary near-unanimity polymorphism for some $k \geq 2$, and no Maltsev polymorphism then $\exists\mathrm{CSP}(\mathbf{A})$ is not constant-query testable (even with two-sided error) but is sublinear-query testable with one-sided error.*[1]
(3) *Else, testing $\exists\mathrm{CSP}(\mathbf{A})$ with one-sided error requires a linear number of queries.*

Let us point out that the problem families $\mathrm{CSP}(\mathbf{A})$ and $\exists\mathrm{CSP}(\mathbf{A})$ exhibit the same dichotomy for constant-query testability, and in particular the positive result there is robust with respect to the introduction of quantifiers. An implication of Theorem I.7 is this: if the dichotomy for sublinear-query testability for $\mathrm{CSP}(\mathbf{A})$ was not the same as that for $\exists\mathrm{CSP}(\mathbf{A})$, then the positive result for that dichotomy would not enjoy this robustness property, and hence such a positive result would have to crucially exploit the absence of quantifiers. Hence Theorem I.7 reveals information about the form of a potential trichotomy for $\mathrm{CSP}(\mathbf{A})$.

Example I.8. Consider the relational structure $\mathbf{A}$ over the Boolean domain $\{0,1\}$ whose only relation is $\leq$. This structure is readily verified to have a majority polymorphism, and does not have a Maltsev polymorphism: for any Maltsev operation p, it holds that applying p to the tuples $(1,1), (0,1), (0,0)$, which are in the relation $\leq$, yields $(1,0)$, which is not in the relation $\leq$. Thus, Theorem I.7 implies that $\exists\mathrm{CSP}(\mathbf{A})$ is not constant-query testable but is sublinear-query testable with one-sided error. $\square$

[1] We remark that the combination of having a $(k+1)$-ary near unanimity polymorphism for some $k \geq 2$ and a Maltsev polymorphism is equivalent to having majority and Maltsev polymorphisms [11].

Example I.9. We can generalize the previous example as follows. Let D be any finite set of size greater than or equal to 2, and consider the *dual discriminator operation* Δ defined as follows: $\Delta(x,y,z)$ is equal to x if $x = y$, and is equal to z otherwise. Consider the relational structure $\mathbf{A}$ with universe D and the following relations: each unary relation; each graph of a permutation (on D); and, each *two-fan relation*. Here, a *two-fan relation* is a binary relation $R \subseteq D \times D$ such that there exist elements $a, b \in D$ where $R = (\{a\} \times \pi_2(R)) \cup (\pi_1(R) \times \{b\})$. It is straightforward to verify that Δ is a majority polymorphism of $\mathbf{A}$. On the other hand, let $a, b \in D$ be arbitrary elements, and consider the relation $S = (\{a\} \times D) \cup (D \times \{b\})$. The relation S is a two-fan relation and so is a relation of $\mathbf{A}$, but does not have a Maltsev polymorphism; we argue this as follows. Let a' be an element of D distinct from a, and let b' be an element of D distinct from b. We have that the tuples $(a', b), (a, b), (a, b')$ are in S, but if we apply any Maltsev polymorphism p to them, we obtain (a', b') which is not in S. The structure $\mathbf{A}$ thus does not have a Maltsev polymorphism; we obtain by Theorem I.7 that $\exists\mathrm{CSP}(\mathbf{A})$ is not constant-query testable but is sublinear-query testable with one-sided error. $\square$

C. *Proof outline*

We now outline our proofs of Theorems I.2 and I.7.

$\mathbf{A}$ *has majority and Maltsev polymorphisms* $\Rightarrow$ $\exists\mathrm{CSP}(\mathbf{A})$ *is constant-query testable.*: We first look at (1) of Theorem I.2 and I.7. As ∃CSPs are a generalization of CSPs, it suffices to consider ∃CSPs. Let $(\mathscr{I}, \varepsilon, f)$ be an input of the assignment testing of $\exists\mathrm{CSP}(\mathbf{A})$. First, we preprocess $\mathscr{I}$ so that it becomes 2-consistent (see Section II for the formal definition). Using the 2-consistency of $\mathscr{I}$ and the majority polymorphism of $\mathbf{A}$ we can assume that for each variable x of $\mathscr{I}$, the set of allowed values for x forms a domain A_x that is the universe of an algebra $\mathbb{A}_x$ that is a factor (i.e., a homomorphic image of a subalgebra) of $\mathrm{Alg}(\mathbf{A})$, the algebra of polymorphisms of $\mathbf{A}$. Also, we can assume that for each pair of variables x, y of $\mathscr{I}$ there is a unique binary constraint of $\mathscr{I}$ with scope (x,y) and constraint relation R_{xy}, with R_{xy} the universe of some subdirect subalgebra of $\mathbb{A}_x \times \mathbb{A}_y$. Furthermore these are the only constraints of $\mathscr{I}$.

In order to test whether f satisfies $\mathscr{I}$, we use three types of reductions: a factoring reduction, a splitting reduction, and an isomorphism reduction. Each reduction produces an instance $\mathscr{I}'$ and an assignment f' such that f' satisfies $\mathscr{I}'$ if f satisfies $\mathscr{I}$, and f' is $\Omega(\varepsilon)$-far from satisfying $\mathscr{I}'$ if f

is ε-far from satisfying $\mathscr{I}$. For simplicity, we focus on how we create a new instance $\mathscr{I}'$ here.

The objective of the factoring reduction is to factor, for each variable x of $\mathscr{I}$, the domain A_x by any congruence θ of $\mathbb{A}_x$ (i.e., an equivalence relation on A_x that is compatible with the operations of $\mathbb{A}_x$) for which none of the constraint relations of $\mathscr{I}$ distinguish between θ-related values of A_x.

After ensuring that all of the domains A_x of $\mathscr{I}$ cannot be factored, we then employ a splitting reduction to ensure that for each variable x of $\mathscr{I}$ the algebra $\mathbb{A}_x$ is subdirectly irreducible, i.e., cannot be represented as a subdirect product of non-trivial algebras. For any variable x for which $\mathbb{A}_x$ can be represented as a subdirect product of non-trivial algebras $\mathbb{A}_x^1$ and $\mathbb{A}_x^2$ we replace the variable x by the new variables x_1 and x_2 and the domain A_x by the domains A_x^1 and A_x^2. For any other variable y of $\mathscr{I}$, we "split" the constraint relation R_{yx} (and its inverse R_{xy}) into two relations R_{yx_1} and R_{yx_2} that are together equivalent to the original one. We then add these two new relations (and their inverses) to $\mathscr{I}$, along with A_x, now regarded as a binary relation from the variable x_1 to x_2.

After performing the splitting reduction and the factoring reduction, we next define a binary relation $\sim$ on the set of variables of $\mathscr{I}$ such that $x \sim y$ if and only if the constraint relation R_{xy} is the graph of an isomorphism from $\mathbb{A}_x$ to $\mathbb{A}_y$. Using 2-consistency and the fact that the domains of $\mathscr{I}$ are subdirectly irreducible and cannot be factored, it follows that, unless $\mathscr{I}$ is trivial, the relation $\sim$ will be a non-trivial equivalence relation. Within each $\sim$-class, the domains are isomorphic via the corresponding constraint relations of $\mathscr{I}$, and this allows us to produce an isomorphism-reduced instance $\mathscr{I}'$ by restricting $\mathscr{I}$ to a set of variables representing each of the $\sim$-classes.

After performing this isomorphism reduction, the resulting instance may have domains which can be further factored, allowing us to apply the factoring reduction to produce a smaller instance. We show that if we reach a point at which none of the three reductions can be applied, the instance must be trivial, either having just a single variable, or for which $|A_x| = 1$ for all variables x. We also show that this point will be reached after applying the reductions at most $|A|$-times.

$\mathrm{CSP}(\mathbf{A})$ *is constant-query testable* $\Rightarrow$ $\mathbf{A}$ *has majority and Maltsev polymorphisms.:* Now we look at (2) of Theorem I.2 and the hardness part of (2) of Theorem I.7. As $\exists$CSPs are a generalization of CSPs, it suffices to consider CSPs. We show that if $\mathbf{A}$ does not have these two types of polymorphisms, then we cannot test $\mathrm{CSP}(\mathbf{A})$ with a constant number of queries. We use that having these two types of polymorphisms is equivalent to $\mathbf{A}$ having a Maltsev polymorphism and that the variety of algebras generated by $\mathrm{Alg}(\mathbf{A})$ is congruence meet semidistributive [17]. The paper [21] provides a characterization of this condition in terms of the existence of two special polymorphisms of $\mathbf{A}$. When the variety generated by $\mathrm{Alg}(\mathbf{A})$ is not congruence meet semidistributive, then it can be easily shown from [7], [30] that testing $\mathrm{CSP}(\mathbf{A})$ requires a linear number of queries. When $\mathbf{A}$ does not have a Maltsev polymorphism, then we can reduce $\mathrm{CSP}(\mathbf{A}')$ to $\mathrm{CSP}(\mathbf{A})$, where the structure $\mathbf{A}'$ has a binary non-rectangular relation. Then, by replacing the 2-SAT relations with this binary non-rectangular relation, we can reuse the argument for showing a super-constant lower bound for 2-SAT in [15] to obtain a super-constant lower bound for $\mathrm{CSP}(\mathbf{A})$.

$\mathbf{A}$ *has a* $(k+1)$*-ary near-unanimity polymorphism, for some* $k \geq 2 \Rightarrow \exists\mathrm{CSP}(\mathbf{A})$ *is sublinear-query testable.:* Now we consider the testability part of (2) of Theorem I.7. It is known that, if $\mathbf{A}$ has a $(k+1)$-ary near unanimity polymorphism, then $\mathrm{CSP}(\mathbf{A})$ is sublinear-query testable with one-sided error in the unweighted case [7]. We slightly modify their argument so that we can handle weights.

$\exists\mathrm{CSP}(\mathbf{A})$ *is sublinear-query testable with one-sided error* $\Rightarrow$ $\mathbf{A}$ *has a* $(k+1)$*-ary near unanimity polymorphism, for some* $k \geq 2$*.:* Finally, we consider (3) of Theorem I.7. We use that $\mathbf{A}$ has a $(k+1)$-ary near unanimity polymorphism for some $k \geq 2$ if and only if the variety of algebras generated by $\mathrm{Alg}(\mathbf{A})$ is congruence meet semidistributive and congruence modular [17], [3]. We already mentioned that if this variety is not congruence meet semidistributive then $\exists\mathrm{CSP}(\mathbf{A})$ is not sublinear-query testable (even with two-sided error). To complete the argument we show that if the variety is not congruence modular then, by building on ideas developed in [13], we can reduce the problem of testing assignments of a circuit in monotone NC^1 to $\exists\mathrm{CSP}(\mathbf{A})$. Note that majority functions are in monotone NC^1 [29], and we can easily show a linear lower bound for one-sided error testers that test assignments of majority functions. Hence, we get a linear lower bound for $\exists\mathrm{CSP}(\mathbf{A})$.

D. Related work

Assignment testing of CSPs was implicitly initiated by [15]. There, it was shown that 2-CSPs are testable with $O(\sqrt{n})$ queries and require $\Omega(\log n/\log\log n)$ queries for any fixed $\varepsilon > 0$. On the other hand, 3-SAT [6], 3-LIN [6], and Horn SAT [7] require $\Omega(n)$ queries to test.

The universal algebraic approach was first used in [30] to study the assignment testing of the list H-homomorphism problem. For graphs G, H, and list constraints $L_v \subseteq V(H)$ $(v \in V(G))$, we say that a mapping $f : V(G) \to V(H)$ is a *list homomorphism* from G to H with respect to the list constraints L_v $(v \in V(G))$ if $f(v) \in L_v$ for any $v \in V(G)$ and $(f(u), f(v)) \in E(H)$ for any $(u,v) \in E(G)$. Then, the corresponding assignment testing problem, parameterized by a graph H, is the following: The input is a tuple $(G, \{L_v\}_{v \in V(H)}, f, \varepsilon)$, where G is a (weighted) graph, $L_v \subseteq V(H)$ $(v \in V(G))$ are list constraints, $f : V(G) \to V(H)$ is a mapping given as a query access, and ε is an error parameter. The goal is testing whether f is a list H-homomorphism from G or ε-far from being so, where ε-farness is defined analogously to testing assignments of CSPs. It was shown in [30] that the algebra (or the variety) associated with the list H-homomorphism characterizes the query complexity, and that list H-homomorphism is constant-query (resp., sublinear-query) testable if and only if H is a reflexive complete graph or an irreflexive complete bipartite graph (resp., a bi-arc graph).

Testing assignments of Boolean CSPs was studied in [7], and in that paper relational structures were classified into three categories: (i) structures $\mathbf{A}$ for which CSP($\mathbf{A}$) is constant-query testable, (ii) structures $\mathbf{A}$ for which CSP($\mathbf{A}$) is not constant-query testable but sublinear-query testable, and (iii) structures $\mathbf{A}$ for which CSP($\mathbf{A}$) is not sublinear-query testable. They also relied on the fact that algebras (or varieties) can be used to characterize query complexity.

E. Organization

Section II introduces the basic notions used throughout this paper. In this extended abstract, we concentrate on CSPs (instead of ∃CSPs) with majority and Maltsev polymorphisms, and show their constant-query testability in Section III. Discussion of our other results are deferred to the full version.

II. Preliminaries

For an integer k, let $[k]$ denote the set $\{1,\dots,k\}$.

Constraint satisfaction problems: For an integer $k \geq 1$, a k-ary relation on a domain A is a subset of A^k. A *constraint language* on a domain A is a finite set of relations on A. A *relational structure*, or simply a *structure* $\mathbf{A} = (A;\Gamma)$ consists of a non-empty set A and a constraint language Γ on A.

For a structure $\mathbf{A} = (A;\Gamma)$, we define the problem CSP($\mathbf{A}$) as follows. An instance $\mathscr{I} = (V, A, \mathscr{C}, \boldsymbol{w})$ consists of a set of variables V, a set of constraints $\mathscr{C}$, and a weight function $\boldsymbol{w}$ with $\sum_{x\in V} \boldsymbol{w}(x) = 1$. Here, each constraint $C \in \mathscr{C}$ is of the form $\langle (x_1,\dots,x_k), R\rangle$, where $x_1,\dots,x_k \in V$ are variables, R is a relation in Γ and k is the arity of R. An *assignment* for $\mathscr{I}$ is a mapping $f : V \to A$, and f is called a *satisfying assignment* if f satisfies all the constraints, that is, $(f(x_1),\dots,f(x_k)) \in R$ for every constraint $\langle (x_1,\dots,x_k), R\rangle \in \mathscr{C}$.

Algebras and Varieties:: Let $\mathbb{A} = (A;F)$ be an algebra. A set $B \subseteq A$ is a *subuniverse* of $\mathbb{A}$ if every operation $f \in F$ restricted to B has image contained in B. For a nonempty subuniverse B of an algebra $\mathbb{A}$, $f|_B$ is the restriction of f to B. The algebra $\mathbb{B} = (B, F|_B)$, where $F|_B = \{f|_B \mid f \in F\}$ is a *subalgebra* of $\mathbb{A}$. Algebras $\mathbb{A}, \mathbb{B}$ are of the *same type* if they have the same number of operations and corresponding operations have the same arities. Given algebras $\mathbb{A}, \mathbb{B}$ of the same type, the *product* $\mathbb{A} \times \mathbb{B}$ is the algebra with the same type as $\mathbb{A}$ and $\mathbb{B}$ with universe $A \times B$ and operations computed coordinate-wise. A subalgebra $\mathbb{C}$ of $\mathbb{A} \times \mathbb{B}$ is a *subdirect product* of $\mathbb{A}$ and $\mathbb{B}$ if the projections of C to A and C to B are both onto. An equivalence relation θ on A is called a *congruence* of an algebra $\mathbb{A}$ if θ is a subalgebra of $\mathbb{A} \times \mathbb{A}$. The collection of congruences of an algebra naturally forms a lattice under the inclusion ordering, and this lattice is called the *congruence lattice* of the algebra. Given a congruence θ on A, we can form the *homomorphic image* $\mathbb{A}/\theta$, whose elements are the equivalence classes of $\mathbb{A}$ and the operations are defined so that the natural mapping from $\mathbb{A}$ to $\mathbb{A}/\theta$ is a homomorphism. An operation $f(x_1,\dots x_n)$ on a set A is *idempotent* if $f(a,a,\dots,a) = a$ for all $a \in A$, an algebra $\mathbb{A}$ is *idempotent* if each of its operations is, and a class of algebras is idempotent if each of its members is. We note that if $\mathbb{A}$ is *idempotent*, then for any congruence θ of $\mathbb{A}$, the θ-classes are all subuniverses of $\mathbb{A}$.

A *variety* is a class of algebras of the same type closed under the formation of homomorphic images, subalgebras, and products. For any algebra $\mathbb{A}$, there is a smallest variety containing $\mathbb{A}$, denoted by $\mathscr{V}(\mathbb{A})$ and called the *variety generated* by $\mathbb{A}$. It is well known that any variety is generated by an algebra and that any member of $\mathscr{V}(\mathbb{A})$ is a homomorphic image of a subalgebra of a power of $\mathbb{A}$.

A. Assignment problems

An *assignment problem* consists of a set of *instances*, where each instance $\mathscr{I}$ has associated with it a set of variables V, a domain A_v for each variable $v \in V$, and a weight function $\boldsymbol{w} : V \to [0,1]$ with $\sum_{v\in V} \boldsymbol{w}(v) = 1$. An assignment of $\mathscr{I}$ is a mapping f defined on V with $f(x) \in A_x$ for each variable $x \in V$. Each instance $\mathscr{I}$ of an assignment problem has associated with it a notion of a *satisfying assignment*. For two assignments f and g for $\mathscr{I}$, we define their distance as $\mathrm{dist}(f,g) := \sum_{x\in V: f(x)\neq g(x)} \boldsymbol{w}(x)$. We define $\mathrm{dist}_{\mathscr{I}}(f) = \min_g \mathrm{dist}(f,g)$, where g is over all satisfying assignments of $\mathscr{I}$. For $\varepsilon \in [0,1]$, we say that an assignment f for $\mathscr{I}$ is *ε-far from satisfying* $\mathscr{I}$ if $\mathrm{dist}_{\mathscr{I}}(f) > \varepsilon$. In the *assignment testing problem* corresponding to an assignment problem, we are given an instance $\mathscr{I}$ of the assignment problem and a query access to an assignment f for $\mathscr{I}$, that is, we can obtain the value of $f(x)$ by querying $x \in V$. Then, we say that an algorithm is a *tester* for the assignment problem if it accepts with probability at least $2/3$ when f is a satisfying assignment of $\mathscr{I}$, and rejects with probability at least $2/3$ when f is ε-far from satisfying $\mathscr{I}$. The *query complexity* of a tester is the number of queries to f.

We can naturally view CSP($\mathbf{A}$) as an assignment problem: for each instance on a set of (free) variables V, the associated assignments are the mappings from V to A, and the notion of satisfying assignments is as described above. Note that an input to the assignment testing problem corresponding to CSP($\mathbf{A}$) is a tuple $(\mathscr{I}, \varepsilon, f)$, where $\mathscr{I}$ is an instance of CSP($\mathbf{A}$), ε is an error parameter, and f is an assignment to $\mathscr{I}$. In order to distinguish $\mathscr{I}$ from the tuple $(\mathscr{I}, \varepsilon, f)$, we always call the former an *instance* and the latter an *input*.

1) Gap-preserving local reductions: We will frequently use the following reduction when constructing algorithms as well as when showing lower bounds.

Definition II.1 (Gap-preserving local reduction)**.** *Given assignment problems $\mathscr{P}$ and $\mathscr{P}'$, there is a* (randomized) gap-preserving local reduction from $\mathscr{P}$ to $\mathscr{P}'$ *if there exist a function $t(n)$ and constants c_1, c_2 satisfying the following: given a $\mathscr{P}$-instance $\mathscr{I}$ of with variable set V and an assignment f for $\mathscr{I}$, there exist a $\mathscr{P}'$-instance $\mathscr{I}'$ with variable set V' and an assignment f' for $\mathscr{I}'$ such that the following hold:*

1) $|V'| \le t(|V|)$.
2) *If f is a satisfying assignment of $\mathscr{I}$, then f' is a satisfying assignment of $\mathscr{I}'$.*
3) *For any $\varepsilon \in (0,1)$, if* $\mathrm{dist}_{\mathscr{I}}(f) \ge \varepsilon$, *then* $\Pr[\mathrm{dist}_{\mathscr{I}'}(f') \ge c_1\varepsilon] \ge 9/10$ *holds, where the probability is over internal randomness.*
4) *Any query to f' can be answered by making at most c_2 queries to f.*

A *linear reduction* is defined to be a gap-preserving local reduction for which the function $t(n) = O(n)$, $c_1 = O(1)$, and $c_2 = O(1)$.

Lemma II.2 ([30])**.** *Let $\mathscr{P}$ and $\mathscr{P}'$ be assignment problems. Suppose that there exists an ε-tester for $\mathscr{P}'$ with query complexity $q(n,\varepsilon)$ for any $\varepsilon \in (0,1)$, where n is the number of variables in the given instance of $\mathscr{P}'$, and that there exists a gap-preserving local reduction from $\mathscr{P}$ to $\mathscr{P}'$ with a function t and $c_1 = c_2 = O(1)$. Then, there exists an ε-tester for $\mathscr{P}$ with query complexity $O(q(t(n), O(\varepsilon)))$ for any $\varepsilon > 0$, where n is the number of variables in the given instance of $\mathscr{P}$. In particular, linear reductions preserve constant-query and sublinear-query testability.*

III. Constant-Query Testability

In this section, assume that $\mathbf{A} = (A;\Gamma)$ is a structure that has a majority polymorphism $m(x,y,z)$ and a Maltsev polymorphism $p(x,y,z)$. It is known, [11], that this is equivalent to the variety $\mathscr{A}$ generated by the algebra $\mathrm{Alg}(\mathbf{A})$ being *congruence distributive* and *congruence permutable* and also to $\mathbf{A}$ having a $(k+1)$-ary near unanimity polymorphism for some $k \ge 2$ and a Maltsev polymorphism. This means that for each algebra $\mathbb{B} \in \mathscr{A}$, the lattice of congruences of $\mathbb{B}$ satisfies the distributive law and that for each pair of congruences α and β of $\mathbb{B}$, the relations $\alpha \circ \beta$ and $\beta \circ \alpha$ are equal. Such varieties are also said to be *arithmetic*.

An important feature of $\mathscr{A}$ (and in fact of any congruence distributive variety generated by a finite algebra) is that every subdirectly irreducible member of $\mathscr{A}$ has size bounded by $|A|$ ([11]). We will make use of the fact that an algebra is *subdirectly irreducible* if and only if the intersection of all of its non-trivial congruences is non-trivial. This is equivalent to the algebra having a smallest non-trivial congruence. In this section, we will show that $\mathrm{CSP}(\mathbf{A})$ is constant-query testable. Some of the ideas found in this section were inspired by the paper [8].

For our analysis, it is useful to introduce $\mathrm{CSP}(\mathscr{V})$ for a variety $\mathscr{V}$. An instance of $\mathrm{CSP}(\mathscr{V})$ is of the form $(V, \{A_x\}_{x\in V}, \mathscr{C}, \boldsymbol{w})$. Each A_x is the domain of an algebra, denoted by $\mathbb{A}_x$, in $\mathscr{V}$, and each constraint in $\mathscr{C}$ is of the form $\langle (x_1,\dots,x_k), R\rangle$, where R is the domain of a subalgebra $\mathbb{R}$ of $\mathbb{A}_{x_1} \times \cdots \times \mathbb{A}_{x_k}$. In particular, R is also the domain of an algebra in $\mathscr{V}$. The definitions of 2-consistency and an assignment testing problem naturally carry over to instances of $\mathrm{CSP}(\mathscr{V})$.

Let $\mathscr{I} = (V, \{A_x\}_{x\in V}, \mathscr{C}, \boldsymbol{w})$ be an instance of $\mathrm{CSP}(\mathscr{A})$. Since $\mathscr{A}$ is arithmetic, we can assume that each constraint in $\mathscr{C}$ is binary [1]. Hence, we also write

$$\mathscr{I} = (V, \{A_x\}_{x\in V}, \{R_{xy}\}_{(x,y)\in V^2}, \boldsymbol{w})$$

or simply $\mathscr{I} = (V, \{A_x\}, \{R_{xy}\}, \boldsymbol{w})$. Moreover, we can assume that $\mathscr{I}$ is 2-consistent because the set of satisfying assignments does not change after making $\mathscr{I}$ 2-consistent. For $x \in V$, R_{xx} is the equality relation 0_{A_x} on the set A_x, and for distinct variables $x \ne y \in V$, R_{xy} denotes the (unique) binary constraint relation from A_x to A_y. We always have $R_{yx} = R_{xy}^{-1} = \{(b,a) \mid (a,b) \in R_{xy}\}$ for any $x, y \in V$. We note that by 2-consistency, it follows that for distinct variables x and y, the relation $R_{x,y}$ is subdirect in $A_x \times A_y$. Throughout the remainder of this section, we will assume that any instance of $\mathrm{CSP}(\mathscr{A})$ considered will be 2-consistent and has only binary constraints.

Since $\mathscr{A}$ is assumed to be congruence permutable (then for any $x \ne y \in V$, the binary relation R_{xy} is *rectangular*, that is, $(a,c),(a,d),(b,d) \in R_{xy}$ implies $(b,c) \in R_{xy}$. As noted in Lemma 2.10 of [8], this is equivalent to R_{xy} being a *thick mapping*. This means that there are congruences θ_{xy} of $\mathbb{A}_x$ and θ_{yx} of $\mathbb{A}_y$ such that modulo the congruence $\theta_{xy} \times \theta_{yx}$ on $\mathbb{R}_{xy}$, the relation R_{xy} is the graph of an isomorphism ϕ_{xy} from $\mathbb{A}_x/\theta_{xy}$ to $\mathbb{A}_y/\theta_{yx}$ and such that for all $a \in A_x$ and $b \in A_y$, $(a,b) \in R_{xy}$ if and only if $\phi_{xy}(a/\theta_{xy}) = b/\theta_{yx}$. In this situation, we say that R_{xy} is a thick mapping with respect to θ_{xy}, θ_{yx} and ϕ_{xy}. For future reference, we note that if for some variables $x \ne y$, the congruence $\theta_{xy} = 0_{A_x}$ then the relation R_{yx} is the graph of a surjective homomorphism from $\mathbb{A}_y$ to $\mathbb{A}_x$.

A. A factoring reduction

Let $\mathscr{I} = (V, \{A_x\}, \{R_{xy}\}, \boldsymbol{w})$ be an instance of $\mathrm{CSP}(\mathscr{A})$ and for each $x \in V$ let $\mu_x = \bigwedge_{y \ne x} \theta_{xy}$, a congruence of $\mathbb{A}_x$. We say that A_x is *prime* if μ_x is the equality congruence 0_{A_x} and *factorable* otherwise. Roughly speaking, if A_x is not prime, then we can factor A_x by μ_x without changing the problem, because no constraint of $\mathscr{I}$ distinguishes values within any μ_x-class. Formally, we define the factoring reduction as in Algorithm 1.

Let $(\mathscr{I}, \varepsilon, f)$ be an input of $\mathrm{CSP}(\mathscr{A})$ and let $(\mathscr{I}', \varepsilon', f') = \textsc{Factor}(\mathscr{I}, \varepsilon, f)$. It is clear that since the instance $\mathscr{I}$ of $\mathrm{CSP}(\mathscr{A})$ is assumed to be 2-consistent then the instance $\mathscr{I}'$ will also be 2-consistent. Furthermore, the sizes of the

Algorithm 1

```
1: procedure FACTOR($\mathscr{I} = (V, \{A_x\}, \{R_{xy}\}, \boldsymbol{w}), \varepsilon, f$)
2:   for $x \in V$ do $A_x \leftarrow A_x/\mu_x$ and $f(x) \leftarrow f(x)/\mu_x$.
3:   for $(x,y) \in V \times V$ do
4:     $R_{xy} \leftarrow \{(a/\mu_x, b/\mu_y) \mid (a,b) \in R_{xy}\}$.
5:   return $(\mathscr{I}, \varepsilon, f)$.
```

domains of $\mathscr{I}'$ are no larger than the sizes of the domains of $\mathscr{I}$. Now we show that the factoring reduction is a linear reduction.

Lemma III.1. *Let $(\mathscr{I}, \varepsilon, f)$ be an input of* CSP$(\mathscr{A})$ *and let $(\mathscr{I}', \varepsilon', f') =$* FACTOR$(\mathscr{I}, \varepsilon, f)$. *If $(\mathscr{I}', \varepsilon', f')$ is testable with $q(\varepsilon')$ queries, then $(\mathscr{I}, \varepsilon, f)$ is testable with $q(O(\varepsilon))$ queries.*

Proof: We show that the factoring reduction is a linear reduction. Let $\mathscr{I} = (V, \{A_x\}, \{R_{xy}\}, \boldsymbol{w})$ and $\mathscr{I}' = (V', \{A'_x\}_{x \in V}, \{R'_{xy}\}, \boldsymbol{w}')$ be the original instance and the reduced instance, respectively.

Note that $|V'| = |V|$ and we can determine the value of $f'(x)$ by querying $f(x)$.

If f satisfies $\mathscr{I}$, then f' also satisfies $\mathscr{I}'$. Suppose that f' is ε-close to satisfying $\mathscr{I}'$ and let g' be a satisfying assignment of $\mathscr{I}'$ with $\mathrm{dist}_{\mathscr{I}'}(f', g') \leq \varepsilon$. Then, we define g to be any assignment for $\mathscr{I}$ such that for $x \in V$, $g(x)$ is taken to be an arbitrary element in the μ_x-class $g'(x)$. Then, g satisfies $\mathscr{I}$ and $\mathrm{dist}_{\mathscr{I}'}(f, g) = \mathrm{dist}_{\mathscr{I}}(f', g') \leq \varepsilon$.

To summarize, the factoring reduction is a gap-preserving local reduction with $t(n) = n$, $c_1 = 1$, and $c_2 = 1$. ■

B. Reduction to instances with subdirectly irreducible domains

In this section, we provide a reduction that produces instances whose domains are all subdirectly irreducible. Suppose that $\mathbb{A}$ is a subdirect product of two algebras $\mathbb{A}_1$, $\mathbb{A}_2$ from $\mathscr{A}$ and that $\mathbb{R}$ is a subdirect product of $\mathbb{A}$ and $\mathbb{B}$ for some $\mathbb{B} \in \mathscr{A}$. We can project the relation R onto the factors of $\mathbb{A}$ to obtain two new binary relations from A_1 to B and from A_2 to B, respectively:

$$R_1 = \{(a_1, b) \mid \exists (a_1, c_2) \in A \text{ with } ((a_1, c_2), b) \in R\},$$
$$R_2 = \{(a_2, b) \mid \exists (c_1, a_2) \in A \text{ with } ((c_1, a_2), b) \in R\}.$$

The following shows that the relation R can be recovered from the relations R_1, R_2, and A (considered as a relation from A_1 to A_2).

Lemma III.2. *For all $a_1 \in A_1$, $a_2 \in A_2$, and $b \in B$, the following are equivalent:*

- $((a_1, a_2), b) \in R$
- $(a_1, b) \in R_1$, $(a_2, b) \in R_2$ *and* $(a_1, a_2) \in A$.

Proof: One direction of this claim follows by construction. For the other, suppose that $(a_1, b) \in R_1$, $(a_2, b) \in R_2$ and $(a_1, a_2) \in A$. Then there are elements $c_i \in A_i$, for $i = 1, 2$, with (a_1, c_2), $(c_1, a_2) \in A$, $((a_1, c_2), b)$, $((c_1, a_2), b) \in R$. Since R is subdirect in $A \times B$ and $(a_1, a_2) \in A$ then there is some $d \in B$ with $((a_1, a_2), d) \in R$. Applying the majority term of $\mathscr{A}$ coordinate-wise to the tuples $((a_1, c_2), b)$, $((c_1, a_2), b)$, and $((a_1, a_2), d)$ from R we produce the tuple $((a_1, a_2), b) \in R$, as required. ■

Algorithm 2

```
1: procedure SPLIT($\mathscr{I} = (V, \{A_x\}, \{R_{xy}\}, \boldsymbol{w}), \varepsilon, f$)
2:   while there exists $x \in V$ such that $\mathbb{A}_x$ is not subdirectly
       irreducible or trivial do
3:     Replace $\mathbb{A}_x$ in $\mathscr{I}$ with an isomorphic non-trivial sub-
         direct product of $\mathbb{A}_x^1 \times \mathbb{A}_x^2$ for some quotients $\mathbb{A}_x^1$,
         $\mathbb{A}_x^2$ of $\mathbb{A}_x$ such that $\mathbb{A}_x^1$ is subdirectly irreducible.
4:     $V \leftarrow (V \setminus \{x\}) \cup \{x_1, x_2\}$, where $x_1$ and $x_2$ are newly
         introduced variables.
5:     Remove the domain $A_x$ and add the domains $A_x^1$ and
         $A_x^2$ over the variables $x_1$ and $x_2$ respectively.
6:     $\mathscr{C} \leftarrow \mathscr{C} \setminus \{\langle (x,x), R_{xx} \rangle, \langle (x,y), R_{xy} \rangle, \langle (y,x), R_{yx} \rangle\}_{y \in V \setminus \{x\}}$.
7:     $\mathscr{C} \leftarrow \mathscr{C} \cup \{\langle (x_1,x_1), 0_{A_{x_1}} \rangle, \langle (x_2,x_2), 0_{A_{x_2}} \rangle,$
         $\langle (x_1,x_2), A_x \rangle, \langle (x_2,x_1), A_x^{-1} \rangle\}$.
8:     $\mathscr{C} \leftarrow \mathscr{C} \cup \{\langle (x_1,y), (R_{xy})_1 \rangle, \langle (x_2,y), (R_{xy})_2 \rangle,$
         $\langle (y,x_1), (R_{xy})_1^{-1} \rangle, \langle (y,x_2), (R_{xy})_2^{-1} \rangle\}_{y \in V \setminus \{x\}}$.
9:     Remove $x$ from the domain of $\boldsymbol{w}$ and add $x_1$ and $x_2$.
10:    Set $\boldsymbol{w}(x_1) = \boldsymbol{w}(x)/2$ and $\boldsymbol{w}(x_2) = \boldsymbol{w}(x)/2$.
11:    Remove $x$ from the domain of $f$ and add $x_1$ and $x_2$.
12:    Set $f(x_1) \in A_x^1$ and $f(x_2) \in A_x^2$ so that
         $(f(x_1), f(x_2)) = f(x)$.
13:  return $(\mathscr{I}, \varepsilon/2^{|A|}, f)$.
```

Lemma III.2 allows us to split a domain of an instance of CSP$(\mathscr{A})$ into subdirectly irreducible domains. Formally, we define the splitting reduction as in Algorithm 2.

Let $(\mathscr{I}, \varepsilon, f)$ be an input of CSP$(\mathscr{A})$ and let $(\mathscr{I}', \varepsilon', f') =$ SPLIT$(\mathscr{I}, \varepsilon, f)$. It is clear that, since $\mathscr{I}$ is assumed to be a 2-consistent instance of CSP$(\mathscr{A})$ then the splitting reduction constructs another 2-consistent instance $\mathscr{I}'$ of CSP$(\mathscr{A})$ whose domains are all subdirectly irreducible and so have size bounded by $|A|$ (and are no bigger than the domains of $\mathscr{I}$). The next lemma shows that if a domain of an instance $\mathscr{I}$ is prime, then after splitting it, the resulting subdirect factors will also be prime.

Lemma III.3. *Let $\mathscr{I}'$ be the instance of* CSP$(\mathscr{A})$ *obtained by splitting a domain $\mathbb{A}_x$ of another instance $\mathscr{I}$ into two subdirect factors $\mathbb{A}_{x_1}$ and $\mathbb{A}_{x_2}$ as in the* SPLIT *procedure. If the domain $\mathbb{A}_x$ is prime in $\mathscr{I}$ then the domains $\mathbb{A}_{x_1}$ and $\mathbb{A}_{x_2}$ are prime in $\mathscr{I}'$.*

Proof: Let $\mathscr{I} = (V, \{A_x\}, \{R_{xy}\}, \boldsymbol{w})$ be given and suppose that the domain $\mathbb{A}_x$ is a subdirect product of the algebras $\mathbb{A}_{x_1}$ and $\mathbb{A}_{x_2}$. To produce $\mathscr{I}'$ from $\mathscr{I}$ by splitting $\mathbb{A}_x$, we replace the variable x and the domain A_x with the variables x_1 and x_2 and the corresponding domains A_{x_1} and A_{x_2}. For each $y \in V$ with $x \neq y$, we replace the constraint $\langle (x,y), R_{xy}\rangle$ with the constraints $\langle (x_1,y),(R_{xy})_1\rangle$ and $\langle (x_2,y),(R_{xy})_2\rangle$ and add the constraint $\langle (x_1,x_2),A_x\rangle$.

If the domain $\mathbb{A}_x$ is prime in $\mathscr{I}$ then there is $k \geq 1$ and variables $y_i \in V \setminus \{x\}$, for $1 \leq i \leq k$, such that $\bigwedge_{1\leq i\leq k} \theta_{xy_i} = 0_{A_x}$. To show that $\mathbb{A}_{x_1}$ is prime in $\mathscr{I}'$ it will suffice to show that $\left(\bigwedge_{1\leq i\leq k} \theta_{x_1y_i}\right) \wedge \theta_{x_1x_2} = 0_{A_{x_1}}$.

To establish this, suppose that (a_1,a_1') belongs to the left hand side of this equality. We will show that $a_1 = a_1'$. We have that $(a_1,a_1') \in \theta_{x_1y_i}$ for $1 \leq i \leq k$ and $(a_1,a_1') \in \theta_{x_1x_2}$. From the latter membership it follows that there is some $c \in A_{x_2}$ such that (a_1,c), $(a_1',c) \in A_x$. From $(a_1,a_1') \in \theta_{x_1y_i}$ it follows that there is some $u \in A_{y_i}$ with (a_1,u), $(a_1',u) \in (R_{xy_i})_1$. We can conclude that there are d, $d' \in A_{y_i}$ with $((a_1,d),u)$, $((a_1',d'),u) \in R_{xy_i}$. We then have that $((a_1,d),(a_1',d')) \in \theta_{xy_i}$. We can now apply the majority term of $\mathscr{A}$ coordinate-wise to the following three pairs of members of θ_{xy_i} to establish that $((a_1,c),(a_1',c)) \in \theta_{xy_i}$: $((a_1,d),(a_1',d'))$, $((a_1,c),(a_1,c))$, and $((a_1',c),(a_1',c))$. We've shown that (a_1,c) and (a_1',c) are θ_{xy_i}-related for all $i \leq k$ and so we have that $(a_1,c) = (a_1',c)$, which implies that $a_1 = a_1'$, as required. Thus $\mathbb{A}_{x_1}$ is prime in $\mathscr{I}'$ and by symmetry, $\mathbb{A}_{x_2}$ is also prime. ■

Now we show that the splitting reduction is a gap-preserving local reduction.

Lemma III.4. *Let $(\mathscr{I},\varepsilon,f)$ be an input of* CSP$(\mathscr{A})$ *and let $(\mathscr{I}',\varepsilon',f') = $* Split$(\mathscr{I},\varepsilon,f)$. *If $(\mathscr{I}',\varepsilon',f')$ is testable with $q(\varepsilon')$ queries, then $(\mathscr{I},\varepsilon,f)$ is testable with $q(O(\varepsilon))$ queries.*

Proof: We show that the splitting reduction is a linear reduction.

Let $\mathscr{I} = (V,\{A_x\},\{R_{xy}\},\boldsymbol{w})$ and $\mathscr{I}' = (V',\{A_x'\},\{R_{xy}'\},\boldsymbol{w}')$ be the original instance and the reduced instance, respectively.

In the reduction, every variable x of V is ultimately split into variables $x_1,\ldots,x_{k_x}$ from V' and the domain $\mathbb{A}_x$ is replaced by subdirectly irreducible domains $\mathbb{A}_x^1,\ldots,\mathbb{A}_x^{k_x}$ corresponding to these variables such that $\mathbb{A}_x$ is isomorphic to a subdirect product of these new domains. Since each of the domains has size bounded by $|A|$, then $k_x \leq |A|$ for all $x \in V$ and so after completely splitting $\mathbb{A}_x$ into the k_x factors, we have that $\boldsymbol{w}(x) \leq 2^{|A|}\boldsymbol{w}'(x_i)$ for each $i \in [k_x]$. We also have that $\sum_{i\in[k_x]} \boldsymbol{w}'(x_i) = \boldsymbol{w}(x)$ for each $x \in V$.

To determine the value of $f'(x_i)$, where x_i is added when splitting the variable x, we only need to know the value of $f(x)$.

If f satisfies $\mathscr{I}$, then f' satisfies $\mathscr{I}'$ by Lemma III.2. Suppose that f' is $\varepsilon/(2^{|A|})$-close to satisfying $\mathscr{I}'$ and let g' be a satisfying assignment for $\mathscr{I}'$ with $\mathrm{dist}(f',g') \leq \varepsilon/(2^{|A|})$. Because the tuple $(g'(x_1),\ldots,g'(x_{k_x}))$ is in A_x, we can naturally define an assignment g for $\mathscr{I}$ by setting $g(x) = (g'(x_1),\ldots,g'(x_{k_x})) \in A_x$. Then g is a satisfying assignment from Lemma III.2. Moreover, $\mathrm{dist}(f,g) = \sum_{x\in V:\exists i\in[k_x],g'(x_i)\neq f'(x_i)} \boldsymbol{w}(x) \leq \sum_{x\in V}\sum_{i\in[k_x]:g'(x_i)\neq f'(x_i)} 2^{|A|}\boldsymbol{w}'(x_i) = 2^{|A|}\mathrm{dist}(f',g') \leq \varepsilon$.

To summarize, the splitting reduction is a gap-preserving local reduction with $t(n) = |A|$, $c_1 = 1$, and $c_2 = 2^{|A|}$. ■

C. Isomorphism reduction

By applying the factoring reduction and then the splitting reduction to an instance of CSP$(\mathscr{A})$ we end up with an instance whose domains are either trivial or subdirectly irreducible and prime. For such an instance, we have the following property.

Lemma III.5. *Let $\mathscr{I} = (V,\{A_x\},\{R_{xy}\},\boldsymbol{w})$ be an instance of* CSP$(\mathscr{A})$ *such that $|V| > 1$ and such that every domain is either trivial or is subdirectly irreducible and prime. Then, for each variable $x \in V$, there is at least one variable $y \neq x$ so that $\theta_{xy} = 0_{A_x}$ and for such variables y, the relation R_{yx} is the graph of a surjective homomorphism from $\mathbb{A}_y$ to $\mathbb{A}_x$.*

Proof: If $|A_x| = 1$ then the result follows trivially. Otherwise, we have that the congruence $\mu_x = \bigwedge_{y\neq x} \theta_{xy}$ of $\mathbb{A}_x$ is equal to 0_{A_x}, since $\mathbb{A}_x$ is prime. But, since this algebra is subdirectly irreducible, it follows that for some $y \neq x$, $\theta_{xy} = 0_{A_x}$. Since R_{yx} is a thick mapping with $\theta_{xy} = 0_{A_x}$ it follows that R_{yx} is the graph of a surjective homomorphism from $\mathbb{A}_y$ to $\mathbb{A}_x$. ■

Let $\mathscr{I} = (V,\{A_x\},\{R_{xy}\},\boldsymbol{w})$ be an instance of CSP$(\mathscr{A})$ with $|V| > 1$ and with the property that every domain is either trivial or is subdirectly irreducible and prime. Define the relation $\sim$ on V by $x \sim y$ if and only if the relation R_{xy} is the graph of an isomorphism from $\mathbb{A}_x$ to $\mathbb{A}_y$. Using the 2-consistency of $\mathscr{I}$, the relation $\sim$ is naturally an equivalence relation on V. The following corollary to Lemma III.5 establishes that unless all of the domains of $\mathscr{I}$ are trivial, the relation $\sim$ is non-trivial.

Corollary III.6. *For $\mathscr{I} = (V,\{A_x\},\{R_{xy}\},\boldsymbol{w})$ an instance of* CSP$(\mathscr{A})$ *as in Lemma III.5, if $x \in V$ is such that the domain A_x has maximal size and has at least two elements, then there is some $y \in V$ with $x \neq y$ and $x \sim y$.*

Proof: If A_x has maximal size and has at least two elements, then let $y \in V$ be a variable such that $x \neq y$ and R_{yx} the graph of a surjective homomorphism from $\mathbb{A}_y$ to $\mathbb{A}_x$. Since A_x has maximal size, it follows that $|A_y| = |A_x|$ and so R_{yx} is the graph of an isomorphism from $\mathbb{A}_y$ to $\mathbb{A}_x$. ■

For a variable $x \in V$, let $[x] := x/\sim$ denote the $\sim$-class of V that x belongs to. Let $S \subseteq V$ be an arbitrary complete system of representatives of this equivalence relation and for any $\sim$-class u, let $s(u) \in V$ be the unique element $x \in S$ such that $x \in u$. In particular $[s(u)] = u$ holds.

Given an assignment f for $\mathscr{I}$, we can test the input $(\mathscr{I},\varepsilon,f)$ in two steps. First, we test whether the values of f in the $\sim$-classes of V are consistent using a consistency algorithm (Algorithm 3) and then we test the input obtained by contracting the $\sim$-classes using Algorithm 4. Explanations of these two steps are contained in the next two subsections.

Algorithm 3

```
1: procedure CONSISTENCY($\mathscr{I},\varepsilon,f$)
2:   Sample a set $U$ of $\Theta(1/\varepsilon)$ $\sim$-classes of $\mathscr{I}$. In each sampling, $u$ is chosen with probability $\overline{w}(u)$.
3:   for each $u \in U$ do
4:     Sample a set $S$ of $\Theta(1/\varepsilon)$ variables in $u$. In each sampling, a variable $x \in u$ is chosen with probability $w(x)/\overline{w}(u)$.
5:     if there are two variables $x,y \in S$ with $f(y) \neq R_{xy}(f(x))$ then Reject.
6:   Accept.
```

Algorithm 4

```
1: procedure ISO($\mathscr{I},\varepsilon,f$)
2:   for each $\sim$-class $u$ do
3:     Sample a variable $x \in u$ with probability $w(x)/\overline{w}(u)$, and let $x_u$ be the sampled variable.
4:     $V' \leftarrow V' \cup \{u\}$, $A'_u \leftarrow A_{s(u)}$, and $w'(u) \leftarrow \overline{w}(u)$.
5:     $f'(u) \leftarrow R_{x_u s(u)}(f(x_u))$.
6:   for each pair $(u,u')$ of $\sim$-classes do $R'_{uu'} \leftarrow R_{x_u x_{u'}}$.
7:   return $((V',\{A'_x\},\{R'_{xy}\},w'),\varepsilon/2,f')$.
```

1) Testing $\sim$-consistency: We say that the input $(\mathscr{I},\varepsilon,f)$ is *$\sim$-consistent* if, for each $x \sim y$, $(f(x),f(y)) \in R_{xy}$.

For a $\sim$-class $u \subseteq V$ and $b \in A_{s(u)}$, we define $\overline{w}(u,b) = \sum_{y\in u: f(y)=R_{s(u)y}(b)} w(y)$ and $\overline{w}(u) = \sum_{b\in A_{s(u)}} \overline{w}(u,b)$. Note that $\overline{w}(u)$ is also equal to $\sum_{x\in u} w(x)$, the sum of the weights of the variables in u.

In order to test $\sim$-consistency, we run Algorithm 3. We defer the proof of the following lemma to the full version.

Lemma III.7. *Algorithm 3 tests $\sim$-consistency with query complexity $O(1/\varepsilon^2)$.*

2) Isomorphism reduction: Using Algorithm 3, we can reject an input $(\mathscr{I},\varepsilon,f)$ if it is far from satisfying $\sim$-consistency. In this subsection we will consider a reduction from $(\mathscr{I},\varepsilon,f)$ to another input $(\mathscr{I}',\varepsilon',f')$ assuming that $(\mathscr{I},\varepsilon,f)$ is close to satisfying $\sim$-consistency.

Our reduction, as described in Algorithm 4, contracts the variables in each $\sim$-class to a single variable from that class. It should be clear that since the instance $\mathscr{I}$ of CSP($\mathscr{A}$) is assumed to be 2-consistent, the reduction will produce another 2-consistent instance $\mathscr{I}'$ of CSP($\mathscr{A}$). As the next lemma shows, unless the domains of $\mathscr{I}$ all have size one, some of the domains of $\mathscr{I}'$ will no longer be prime.

Lemma III.8. *Let $(\mathscr{I},\varepsilon,f)$ be an input of CSP($\mathscr{A}$) for which domains of $\mathscr{I}$ are either trivial or prime and subdirectly irreducible and let $(\mathscr{I}',\varepsilon',f') = \text{ISO}(\mathscr{I},\varepsilon,f)$. If some domain of $\mathscr{I}$ has more than one element, then any domain of $\mathscr{I}'$ of maximal size will not be prime, unless $\mathscr{I}'$ has only one variable.*

Algorithm 5

```
1: procedure ISO'($\mathscr{I},\varepsilon,f$)
2:   if CONSISTENCY($\mathscr{I},\varepsilon/20,f$) rejects then Reject.
3:   else return ISO($\mathscr{I},\varepsilon,f$).
```

Proof: Suppose that $\mathscr{I}'$ has more than one variable. This is equivalent to there being more than one $\sim$-class for $\mathscr{I}$. Let x be a variable of $\mathscr{I}'$ with $|A_x|$ of maximal size and let y be any other variable of $\mathscr{I}'$. Note that according to the construction of $\mathscr{I}'$ from $\mathscr{I}$, both x and y are also variables of $\mathscr{I}$ with $x \not\sim y$. Furthermore, $|A_x|$ has maximal size amongst all of the domains of $\mathscr{I}$ and so the relation R_{yx} cannot be the graph of a surjective homomorphism from $\mathbb{A}_y$ to $\mathbb{A}_x$. If it were, then it would be the graph of an isomorphism, contradicting that $x \not\sim y$. Thus the congruence $\theta_{xy} \neq 0_{A_x}$. Since $\mathbb{A}_x$ is subdirectly irreducible it follows that $\mu_x = \bigwedge_{y\neq x} \theta_{xy}$ is also not equal to 0_{A_x} and so A_x is not prime in $\mathscr{I}'$. ■

We defer the proof of the following lemma to the full version.

Lemma III.9. *Let $(\mathscr{I},\varepsilon,f)$ be an input of CSP($\mathscr{A}$) and suppose that f is $\varepsilon/20$-close to satisfying $\mathscr{I}$. Let $(\mathscr{I}',\varepsilon',f') = \text{ISO}(\mathscr{I},f)$. If $(\mathscr{I}',\varepsilon',f')$ is testable with $q(\varepsilon')$ queries, then $(\mathscr{I},\varepsilon,f)$ is testable with $q(O(\varepsilon))$ queries.*

Finally, we combine Algorithm 3 and Algorithm 4. to produce Algorithm 5 and make use of it in the following.

Lemma III.10. *Let $(\mathscr{I},\varepsilon,f)$ be an input of CSP($\mathscr{A}$) and suppose that ISO($\mathscr{I},f$) returned another instance $(\mathscr{I}',\varepsilon',f')$. If $(\mathscr{I}',\varepsilon',f')$ is testable with $q(\varepsilon')$ queries, then $(\mathscr{I},\varepsilon,f)$ is testable with $q(O(\varepsilon))$ queries.*

Proof: Consider Algorithm 5. If f satisfies $\mathscr{I}$, then the $\sim$-consistency test always accepts, and hence we always accept with probability $2/3$ from Lemma III.9. Suppose that f is ε-far from satisfying $\mathscr{I}$. If f is $\varepsilon/20$-far from satisfying $\sim$-consistency, then the $\sim$-consistency test rejects with probability at least $2/3$. If f is $\varepsilon/20$-close to satisfying $\sim$-consistency, then we reject with probability at least $2/3$ by Lemma III.9. ■

D. Putting things together

Combining the reductions introduced so far we can design a shrinking reduction, which shrinks the maximum size of the domains of an instance of CSP($\mathscr{A}$).

Lemma III.11. *Let $(\mathscr{I},\varepsilon,f)$ be an input of CSP($\mathscr{A}$), and suppose that SHRINK($\mathscr{I},\varepsilon,f$) returned another instance $(\mathscr{I}',\varepsilon',f')$. If we can test $(\mathscr{I}',\varepsilon',f')$ with $q(\varepsilon')$ queries, then we can test $(\mathscr{I},\varepsilon,f)$ with $q(O(\varepsilon))$ queries. Moreover, the reduction reduces the maximum size of a domain of the given input, if this maximum is greater than one and the reduced instance has more than one variable.*

Algorithm 6

```
1: procedure SHRINK($\mathscr{I}, \varepsilon, f$)
2:     $(\mathscr{I}, \varepsilon, f) \leftarrow$ FACTOR($\mathscr{I}, \varepsilon, f$).
3:     $(\mathscr{I}, \varepsilon, f) \leftarrow$ SPLIT($\mathscr{I}, \varepsilon, f$).
4:     if ISO$'(\mathscr{I}, \varepsilon, f)$ rejects then Reject.
5:     else $(\mathscr{I}, \varepsilon, f) \leftarrow$ the output returned by ISO$'$.
6:     $(\mathscr{I}, \varepsilon, f) \leftarrow$ FACTOR($\mathscr{I}, \varepsilon, f$).
7:     return $(\mathscr{I}, \varepsilon, f)$.
```

Proof: We note that at each step of the algorithm, the domains of the instances that are produced are no larger than the domains of the original instance. Furthermore, if any of the domains of the original instance has size greater than one, then it follows from Lemma III.8 that the maximal size of the domains of the output instance will be smaller than that of the original instance, as long as the output instance has more than one variable. ■

Theorem III.12. *Let* $\mathbf{A}$ *be a structure that has majority and Maltsev polymorphisms. Then,* $\mathrm{CSP}(\mathbf{A})$ *is constant-query testable with one-sided error.*

Proof: By applying the shrinking reduction at most $|A|$ times, we get an instance for which every variable has a domain of size one or which has only one variable. In either case, the testing becomes trivial. ■

Acknowledgment

H. C. is supported by the Spanish Project MINECO COMMAS TIN2013-46181-C2-R, Basque Project GIU15/30, Basque Grant UFI11/45, and KAKENHI No. 24106001. M. V. is supported by a grant from the Natural Sciences and Engineering Research Council of Canada. Y. Y. is supported by JSPS Grant-in-Aid for Young Scientists (B) (No. 26730009), MEXT Grant-in-Aid for Scientific Research on Innovative Areas (No. 24106003), and JST, ERATO, Kawarabayashi Large Graph Project.

References

[1] K. A. Baker and A. F. Pixley, "Polynomial interpolation and the chinese remainder theorem for algebraic systems," *Mathematische Zeitschrift*, vol. 143, no. 2, pp. 165–174, 1975.

[2] L. Barto, "The collapse of the bounded width hierarchy," *Journal of Logic and Computation*, 2014.

[3] ——, "Finitely related algebras in congruence distributive varieties have near unanimity terms," *Canadian Journal of Mathematics*, vol. 65, no. 1, pp. 3–21, 2013.

[4] ——, "The constraint satisfaction problem and universal algebra," *The Bulletin of Symbolic Logic*, vol. 21, pp. 319–337, 9 2015.

[5] L. Barto and M. Kozik, "Constraint satisfaction problems solvable by local consistency methods," *Journal of the ACM*, vol. 61, no. 1, 2014.

[6] E. Ben-Sasson, P. Harsha, and S. Raskhodnikova, "Some 3CNF properties are hard to test," *SIAM Journal on Computing*, vol. 35, no. 1, 2005.

[7] A. Bhattacharyya and Y. Yoshida, "An algebraic characterization of testable Boolean CSPs," in *ICALP*, 2013, pp. 123–134.

[8] A. Bulatov and D. Marx, "The complexity of global cardinality constraints," *Logical Methods in Computer Science*, vol. 6, pp. 1–27, 2010.

[9] A. A. Bulatov, "Complexity of conservative constraint satisfaction problems," *ACM Transactions on Computational Logic*, vol. 12, no. 4, 2011.

[10] ——, "The complexity of the counting constraint satisfaction problem," *Journal of the ACM*, vol. 60, no. 5, p. 34, 2013.

[11] S. Burris and H. P. Sankappanavar, *A course in universal algebra*, ser. Graduate Texts in Mathematics. Springer-Verlag, New York-Berlin, 1981, vol. 78.

[12] S. O. Chan, J. R. Lee, P. Raghavendra, and D. Steurer, "Approximate constraint satisfaction requires large LP relaxations," in *FOCS*, 2013, pp. 350–359.

[13] H. Chen and M. Valeriote, "Learnability of solutions to conjunctive queries: The full dichotomy," in *COLT*, 2015, pp. 326–337.

[14] M. E. Dyer and D. Richerby, "An effective dichotomy for the counting constraint satisfaction problem," *SIAM Journal on Computing*, vol. 42, no. 3, pp. 1245–1274, 2013.

[15] E. Fischer, E. Lehman, I. Newman, S. Raskhodnikova, R. Rubinfeld, and A. Samorodnitsky, "Monotonicity testing over general poset domains," in *STOC*, 2002, pp. 474–483.

[16] O. Goldreich, Ed., *Property Testing*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, vol. 6390.

[17] D. Hobby and R. McKenzie, *The structure of finite algebras*, ser. Contemporary Mathematics. American Mathematical Society, Providence, RI, 1988, vol. 76, revised edition: 1996.

[18] K. Idziak and P. M. Idziak, "Decidability problem for finite Heyting algebras," *Journal of Symbolic Logic*, vol. 53, no. 3, pp. 729–735, 1988.

[19] P. Idziak, P. Markovic, R. McKenzie, M. Valeriote, and R. Willard, "Tractability and learnability arising from algebras with few subpowers," *SIAM Journal on Computing*, vol. 39, no. 7, pp. 3023–3037, 2010.

[20] S. Khot, "On the power of unique 2-prover 1-round games," in *STOC*, 2002, pp. 767–775.

[21] M. Kozik, A. Krokhin, M. Valeriote, and R. Willard, "Characterizations of several Maltsev conditions," *Algebra Universalis*, vol. 73, no. 3-4, pp. 205–224, 2015.

[22] I. Newman, "Property testing of massively parametrized problems - a survey," *Property Testing*, vol. 6390, no. Chapter 8, pp. 142–157, 2010.

[23] P. Raghavendra, "Optimal algorithms and inapproximability results for every CSP?" in *STOC*, 2008, pp. 245–254.

[24] D. Ron, "Algorithmic and analysis techniques in property testing," *Foundations and Trends® in Theoretical Computer Science*, vol. 5, pp. 73–205, 2010.

[25] R. Rubinfeld and A. Shapira, "Sublinear time algorithms," *SIAM Journal on Discrete Mathematics*, vol. 25, no. 4, pp. 1562–1588, 2011.

[26] J. Thapper and S. Zivny, "The power of linear programming for valued CSPs," in *FOCS*, 2012, pp. 669–678.

[27] ——, "The complexity of finite-valued CSPs," in *STOC*, 2013, pp. 695–704.

[28] ——, "Sherali-adams relaxations for valued CSPs," in *ICALP*, 2015, pp. 1058–1069.

[29] L. G. Valiant, "Short monotone formulae for the majority function," *Journal of Algorithms*, vol. 5, no. 3, pp. 363–366, 1984.

[30] Y. Yoshida, "Testing list H-homomorphisms," *Computational complexity*, pp. 1–37, 2014.