Constant-Query Testability of Assignments to Constraint Satisfaction Problems

HUBIE CHEN, Birkbeck, University of London, United Kingdom
MATT VALERIOTE*, McMaster University, Canada
YUICHI YOSHIDA†, National Institute of Informatics, Japan

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

Our main contribution is a dichotomy theorem completely characterizing the finite structures $\mathbf{A}$ such that CSP($\mathbf{A}$) is constant-query testable:

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

CCS Concepts: • **Theory of computation → Constraint and logic programming**; **Sketching and sampling**;

ACM Reference Format:
Hubie Chen, Matt Valeriote, and Yuichi Yoshida. 2019. Constant-Query Testability of Assignments to Constraint Satisfaction Problems. 1, 1 (January 2019), 26 pages. https://doi.org/0000001.0000001

1 INTRODUCTION

1.1 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 $\epsilon, \delta \in [0, 1]$, 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

*Supported by a grant from the Natural Sciences and Engineering Research Council of Canada
†Supported by JST ERATO Grant Number JPMJER1305 and JSPS KAKENHI Grant Number JP17H04676.

Authors' addresses: Hubie Chen, Birkbeck, University of London, London, United Kingdom, hubie.chen@ehu.es; Matt Valeriote, McMaster University, Hamilton, Canada, matt@math.mcmaster.ca; Yuichi Yoshida, National Institute of Informatics, Chiyoda-ku, Tokyo, 101-8430, Japan, yyoshida@nii.ac.jp.

XXXX-XXXX/2019/1-ART $15.00
https://doi.org/0000001.0000001

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 more than 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 [19, 28, 29] for surveys on property testing.

In *constraint satisfaction problems* (for short, CSPs), 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, one often refers to Γ as a *constraint language* over A, and to $\mathbf{A}$ 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 [2, 4, 9, 22], in counting complexity [10, 16], in computational learning theory [14, 22], and in optimization and approximation [12, 27, 31–33]. See also the survey by Barto [3] for an overview of this line of research. Recently, Bulatov [7] and Zhuk [35] have announced proofs of the Feder-Vardi Dichotomy Conjecture, a conjecture that has driven much of the research on the CSP over the past several years.

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 $(\mathcal{I}, \epsilon, f)$, where $\mathcal{I}$ is an instance of CSP($\mathbf{A}$) with weights on the variables, ϵ is an error parameter, and f is an assignment to $\mathcal{I}$. In the studied model, the tester has full access to $\mathcal{I}$ 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* [26]. We say that f is *ϵ-far* from satisfying $\mathcal{I}$ if one must modify more than an ϵ-fraction of f (with respect to the weights) to make f a satisfying assignment of $\mathcal{I}$, and we say that f is *ϵ-close* otherwise. It is always assumed that $\mathcal{I}$ has a satisfying assignment as otherwise we can immediately reject the input (in this context, time complexity is not taken into account). The objective of assignment testing of CSPs is to correctly decide whether f is a satisfying assignment of $\mathcal{I}$ or is ϵ-far from being so with probability at least $2/3$. When f does not satisfy $\mathcal{I}$ but is ϵ-close to satisfying $\mathcal{I}$, the algorithm can output anything.

In assignment testing, we say that the query complexity of a tester is constant, sublinear, or linear if it is constant, sublinear, or linear (respectively) 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 CSP($\mathbf{A}$).

1.2 Contributions

While previous work on testing assignments to the problems CSP(**A**) studied concrete templates **A** or restricted classes of structures, this article presents a comprehensive classification of the constant query complexity templates. The results in this paper were first announced in [15]. 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 if for any length-k sequence of tuples

$$(a_1^1, \dots, a_r^1), (a_1^2, \dots, a_r^2), \dots, (a_1^k, \dots, a_r^k) \in R,$$

implies

$$(f(a_1^1, \dots, a_1^k), \dots, f(a_r^1, \dots, a_r^k)) \in R.$$

To indicate that f is a polymorphism of R, it is also said that R is *preserved* by f. An operation f is a *polymorphism* of a relational structure **A** if it is a polymorphism of each of its relations. We define the *algebra* of **A**, denoted by Alg(**A**), to be the pair (A; Pol(**A**)), where Pol(**A**) is the set of all polymorphisms of **A**.

Definition 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 $a, b \in A$.*

Theorem 1.1. *Let* **A** *be a relational structure. The following dichotomy holds.*

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

This theorem generalizes characterizations of the constant-query testable *List H-homomorphism problems* [34] and Boolean CSPs [6] to general CSPs. In Section 3 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 this theorem applies.

Example 1. The template **A** over the Boolean domain $\{0, 1\}$ whose only relation is $\neq$ has both majority and Maltsev polymorphisms. In particular, it is readily verified that this relation $\neq$ is preserved by the Maltsev operation on $\{0, 1\}$ defined by $p(x, y, z) = x \oplus y \oplus z$; on the two-element set $\{0, 1\}$, there is a unique majority operation m, and it is readily verified that $\neq$ is preserved by m. Note that CSP(**A**) coincides with the graph 2-coloring problem.

More generally, templates **A** over a finite domain whose relations are graphs of bijections on A have both majority and Maltsev polymorphisms, since they are instances of the next set of examples (Example 2). Instances of CSP(**A**) for such templates **A** coincide with instances of the problem which is the subject of the *unique games conjecture* [25]. □

Example 2. 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 $d(x, y, z)$ 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 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 preserved by the operations of such a ring will have majority and Maltsev polymorphisms. □

Example 3. 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 2, $\mathbb{R}$ 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 $a \in \mathbb{F}_p$, $D_{p,a} = \{((a_2, a_3, a_5), (b_2, b_3, b_5)) \mid a_p = a\}$,
- For $b \in \mathbb{F}_p$, $E_{p,b} = \{((a_2, a_3, a_5), (b_2, b_3, b_5)) \mid b_p = b\}$,

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 1.1, CSP($\mathbf{R}$) has constant query complexity. □

Examples of structures that satisfy the first condition of Theorem 1.1 but that do not have a discriminator operation as a polymorphism can be derived from finite Heyting algebras.

Example 4. Consider the five-element Heyting algebra $\mathbb{M}$ presented in [21, 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 preserved by the operations of $\mathbb{M}$ is $\alpha \cap \beta$. □

Example 5. Bulatov and Marx provide yet another example of a structure having both a majority and a Maltsev polymorphism, in [8, Example 1.1]. Their example is essentially the structure on the 10-element set $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ that has a single ternary relation $R = \{(0, 1, 2), (0, 3, 4), (5, 6, 7), (8, 9, 7)\}$. It can be readily checked that with respect to the usual ordering on A, R is closed under the ternary "in-between" operation and so has a majority polymorphism. It can also be checked that R has a ternary polymorphism $p(x, y, z)$ that satisfies the equations $p(x, x, y) = p(x, y, x) = p(y, x, x) = y$ and so is a Maltsev operation. We note that R is not preserved by the discriminator operation on A. □

Example 6. 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 (note that over the Boolean domain, there is indeed a unique majority operation), 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 1.1 implies that CSP(**A**) is not constant-query testable. From [6] we know that it is sublinear-query testable with one-sided error. □

To conclude this sub-section we present a theorem that addresses the complexity of deciding, for a given relational structure **A**, if CSP(**A**) is constant-query testable.

THEOREM 1.2. *The problem of deciding, for a relational structure* **A**, *if* CSP(**A**) *is constant-query testable is solvable in polynomial time.*

PROOF. According to Theorem 1.1, deciding if CSP(**A**) is constant-query testable amounts to deciding if **A** has majority and Maltsev polymorphisms. From [23] it follows that if **B** is any structure that has both of these types of polymorphisms then CSP(**B**) has bounded width. In the terminology of [13], the condition of having majority and Maltsev polymorphisms is a strong linear Maltsev condition. Since it is the case that a structure will satisfy this condition if and only if the structure obtained from it by expanding it by all one-element unary relations does then we can apply Lemma 3.8 of [13] to produce a polynomial time algorithm that decides, given a structure **A**, if it has both majority and Maltsev polymorphisms. □

1.3 Proof outline

We now present an outline of our proof of Theorem 1.1.

A *has majority and Maltsev polymorphisms* $\Rightarrow$ CSP(**A**) *is constant-query testable.* We first look at (1) of Theorem 1.1. Let $(\mathcal{I}, \epsilon, f)$ be an input to assignment testing of CSP(**A**). First, we preprocess $\mathcal{I}$ so that it becomes 2-consistent and reject if $\mathcal{I}$ has no solution (see Section 3 for the formal definition). Using the 2-consistency of $\mathcal{I}$ and the majority polymorphism of **A** we can assume that for each variable x of $\mathcal{I}$, 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 Alg(**A**), the algebra of polymorphisms of **A**. Also, we can assume that for each pair of variables x, y of $\mathcal{I}$ there is a unique binary constraint of $\mathcal{I}$ with scope (x, y) and constraint relation R_{xy}, with R_{xy} the universe of some subalgebra of $\mathbb{A}_x \times \mathbb{A}_y$. Furthermore these are the only constraints of $\mathcal{I}$.

In order to test whether f satisfies $\mathcal{I}$, we use three types of reductions: a factoring reduction, a splitting reduction, and an isomorphism reduction. Each reduction produces an instance $\mathcal{I}'$ and an assignment f' such that f' satisfies $\mathcal{I}'$ if f satisfies $\mathcal{I}$, and f' is $\Omega(\epsilon)$-far from satisfying $\mathcal{I}'$ if f is ϵ-far from satisfying $\mathcal{I}$. For simplicity, we focus on how we create a new instance $\mathcal{I}'$ here.

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

After ensuring that all of the domains A_x of $\mathcal{I}$ cannot be factored, we then employ a splitting reduction to ensure that for each variable x of $\mathcal{I}$ 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 $\mathcal{I}$, 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 $\mathcal{I}$, 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 $\mathcal{I}$ 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 $\mathcal{I}$ are subdirectly irreducible and cannot be factored, it follows that, unless $\mathcal{I}$ 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 $\mathcal{I}$, and this allows us to produce an isomorphism-reduced instance $\mathcal{I}'$ by restricting $\mathcal{I}$ 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.

In Section 3, we will see how these reductions work on the template in Example 3.

CSP($\mathbf{A}$) *is constant-query testable* $\Rightarrow$ $\mathbf{A}$ *has majority and Maltsev polymorphisms.* Now we look at (2) of Theorem 1.1. We show that if $\mathbf{A}$ does not have these two types of polymorphisms, then we cannot test CSP($\mathbf{A}$) with a constant number of queries. We use the fact that having these two types of polymorphisms is equivalent to $\mathbf{A}$ having a Maltsev polymorphism and that the variety of algebras generated by Alg($\mathbf{A}$) is *congruence meet semidistributive* [20]. When the variety generated by Alg($\mathbf{A}$) is not congruence meet semidistributive, then it can be easily shown from [6, 34] that testing CSP($\mathbf{A}$) requires a linear number of queries. When $\mathbf{A}$ does not have a Maltsev polymorphism, we show that there exists a structure $\mathbf{A}'$ having a binary non-rectangular relation such that we can reduce CSP($\mathbf{A}'$) to CSP($\mathbf{A}$). 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 [17] to obtain a super-constant lower bound for CSP($\mathbf{A}$).

1.4 Related work

Assignment testing of CSPs was implicitly initiated by [17]. 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 $\epsilon > 0$. On the other hand, 3-SAT [5], 3-LIN [5], and Horn SAT [6] require $\Omega(n)$ queries to test.

The universal algebraic approach was first used in [34] 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 $(G, \{L_v\}_{v \in V(G)}, \epsilon, f)$, 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 [34] 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 [6], 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.

1.5 Organization

Section 2 introduces the basic notions used throughout this paper. We show the constant-query testability of CSPs with majority and Maltsev polymorphisms in Section 3. Super-constant lower bounds of CSPs without majority or Maltsev polymorphisms is discussed in Section 4.

2 PRELIMINARIES

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

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 *(finite) relational structure*, or simply a *(finite) structure* $\mathbf{A} = (A; \Gamma)$ consists of a (finite) nonempty set A and a constraint language Γ on A.

For the remainder of this paper we will assume that all relational structures that are mentioned are finite. For a structure $\mathbf{A} = (A; \Gamma)$, we define the problem CSP($\mathbf{A}$) as follows. An instance $\mathcal{I} = (V, A, \mathcal{C}, \boldsymbol{w})$ consists of a set of variables V, a set of constraints $\mathcal{C}$, and a non-negative weight function $\boldsymbol{w}$ with $\sum_{x \in V} \boldsymbol{w}(x) = 1$. Here, each constraint $C \in \mathcal{C}$ is of the form $\langle (x_1, \ldots, x_k), R \rangle$, where $x_1, \ldots, x_k \in V$ are variables, R is a relation in Γ and k is the arity of R. The tuple $(x_1, \ldots, x_k)$ is called the *scope* of the constraint C and R is called the *constraint relation* of C. An *assignment* for $\mathcal{I}$ is a mapping $f : V \to A$, and we say that f is a *satisfying assignment* if f satisfies all the constraints, that is, $(f(x_1), \ldots, f(x_k)) \in R$ for every constraint $\langle (x_1, \ldots, x_k), R \rangle \in \mathcal{C}$.

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 the universe of 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 of $\mathbb{A}$, we can form the *homomorphic image* $\mathbb{A}/\theta$, whose elements are the equivalence classes of θ and the operations are defined so that the natural mapping from $\mathbb{A}$ to $\mathbb{A}/\theta$ is a homomorphism. An operation $f(x_1, \ldots x_n)$ on a set A is *idempotent* if $f(a, a, \ldots, a) = a$ 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 $\mathcal{V}(\mathbb{A})$ 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 $\mathcal{V}(\mathbb{A})$ is a homomorphic image of a subalgebra of a power of $\mathbb{A}$.

Many important properties of the algebras in a variety can be correlated with properties of the congruence lattices of it member algebras. In this work we consider several congruence lattice conditions for varieties, including congruence distributivity, congruence meet semidistributivity, and congruence permutability. Details of these conditions can be found in [20] and more details on the basics of algebras and varieties can be found in [11].

2.1 Assignment problems

An *assignment problem* consists of a set of *instances*, where each instance $\mathcal{I}$ 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 $\mathcal{I}$ is a mapping f defined on V with $f(x) \in A_x$ for each variable $x \in V$. Each instance $\mathcal{I}$ of an assignment problem has associated with it a notion of a *satisfying assignment*. For two assignments f and g for $\mathcal{I}$, we define their distance as $\mathrm{dist}_{\mathcal{I}}(f, g) := \sum_{x\in V:f(x)\neq g(x)} \boldsymbol{w}(x)$. We define $\mathrm{dist}_{\mathcal{I}}(f) = \min_g \mathrm{dist}_{\mathcal{I}}(f, g)$, where g is over all satisfying assignments of $\mathcal{I}$. Then, for $\epsilon \in [0, 1]$, we say that an assignment f for $\mathcal{I}$ is *ϵ-far from satisfying $\mathcal{I}$* if $\mathrm{dist}_{\mathcal{I}}(f) > \epsilon$. In the *assignment testing problem* corresponding to an assignment problem, we are given an instance $\mathcal{I}$ of the assignment problem, $\epsilon \in [0, 1]$, and a query access to an assignment f for $\mathcal{I}$, 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 $\mathcal{I}$, and rejects with probability at least 2/3 when f is ϵ-far from satisfying $\mathcal{I}$. 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 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 $(\mathcal{I}, \epsilon, f)$, where $\mathcal{I}$ is an instance of CSP($\mathbf{A}$), ϵ is an error parameter, and f is an assignment to $\mathcal{I}$. In order to distinguish $\mathcal{I}$ from the tuple $(\mathcal{I}, \epsilon, f)$, we always call the former *instance* and the latter *input*.

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

Definition 2 (Gap-preserving local reduction). *Given assignment problems $\mathcal{P}$ and $\mathcal{P}'$, there is a* (randomized) gap-preserving local reduction from $\mathcal{P}$ to $\mathcal{P}'$ *if there exist a function $t(n)$ and constants c_1, c_2 satisfying the following: given a $\mathcal{P}$-instance $\mathcal{I}$ of with variable set V and an assignment f for $\mathcal{I}$, there exist a $\mathcal{P}'$-instance $\mathcal{I}'$ with variable set V' and an assignment f' for $\mathcal{I}'$ such that the following hold:*

(1) $|V'| \leq t(|V|)$.
(2) *If f is a satisfying assignment of $\mathcal{I}$, then f' is a satisfying assignment of $\mathcal{I}'$.*
(3) *For any $\epsilon \in (0,1)$, if $\mathrm{dist}_{\mathcal{I}}(f) \geq \epsilon$, then $\Pr[\mathrm{dist}_{\mathcal{I}'}(f') \geq c_1\epsilon] \geq 9/10$ 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 $t(n)$ is $O(n)$.

Lemma 2.1 ([34]). *Let $\mathcal{P}$ and $\mathcal{P}'$ be assignment problems. Suppose that there exists an ϵ-tester for $\mathcal{P}'$ with query complexity $q(n,\epsilon)$ for any $\epsilon \in (0,1)$, where n is the number of variables in the given instance of $\mathcal{P}'$, and that there exists a gap-preserving local reduction from $\mathcal{P}$ to $\mathcal{P}'$ with function t. Then, there exists an ϵ-tester for $\mathcal{P}$ with query complexity $O(q(t(n), O(\epsilon)))$ for any $\epsilon > 0$, where n is the number of variables in the given instance of $\mathcal{P}$. In particular, linear reductions preserve constant-query and sublinear-query testability.*

As another application of gap-preserving local reductions, the following fact is known.

Lemma 2.2 (Lemma 6.4 and 6.5 of [34]). *Let $\mathbf{A}, \mathbf{A}'$ be relational structures. If the relations of $\mathbf{A}$ are preserved by the operations of some finite algebra in $\mathcal{V}(\mathrm{Alg}(\mathbf{A}'))$, and* $\mathrm{CSP}(\mathbf{A}')$ *is constant-query testable, then* $\mathrm{CSP}(\mathbf{A})$ *is constant-query testable.*

3 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 $\mathcal{A}$ generated by the algebra $\mathrm{Alg}(\mathbf{A})$ being *congruence distributive* and *congruence permutable*. This means that for each algebra $\mathbb{B} \in \mathcal{A}$, 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 $\mathcal{A}$ (and in fact of any congruence distributive variety generated by a finite algebra) is that every subdirectly irreducible member of $\mathcal{A}$ 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 the problem $\mathrm{CSP}(\mathcal{V})$, for each variety $\mathcal{V}$. An instance of $\mathrm{CSP}(\mathcal{V})$ is of the form $(V, \{A_x\}_{x \in V}, C, \boldsymbol{w})$. Each A_x is the domain of a finite algebra, denoted by $\mathbb{A}_x$, in $\mathcal{V}$, and each constraint in C is of the form

$\langle (x_1, \ldots, x_k), R \rangle$, 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 $\mathcal{V}$. The definition of an assignment testing problem naturally carries over to instances of CSP($\mathcal{V}$).

Let $\mathcal{I} = (V, \{A_x\}_{x \in V}, C, \boldsymbol{w})$ be an instance of CSP($\mathcal{A}$). Since $\mathcal{A}$ has a majority term, we can assume that each constraint in C is binary [1]. Furthermore, we may assume that $\mathcal{I}$ has a solution and is 2-consistent:

- for every $x, y \in V$, there is a unique constraint $C_{xy} = \langle (x, y), R_{xy} \rangle$ of $\mathcal{I}$ with scope (x, y) and the constraint relation R_{xy} is a subdirect product of A_x and A_y,
- for $x \in V$, R_{xx} is the equality relation 0_{A_x} on the set A_x, and
- if $x, y, z \in V$ and $(a, b) \in R_{xy}$ then there is an element $c \in A_z$ such that $(a, c) \in R_{xz}$ and $(b, c) \in R_{yz}$.

Note that from 2-consistency it follows that for all x, $y \in V$, $R_{yx} = R_{xy}^{-1} = \{(b, a) \mid (a, b) \in R_{xy}\}$ for any $x, y \in V$. Under these assumptions, we may write $\mathcal{I}$ as

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

or simply $\mathcal{I} = (V, \{A_x\}, \{R_{xy}\}, \boldsymbol{w})$. It is well known that any CSP instance over a template having a majority polymorphism can be transformed to a 2-consistent instance of the form just described in polynomial time without changing the set of satisfying assignments; see [23] or [8] for more details. So, there is no loss in generality in assuming throughout the rest of this section that any instance of CSP($\mathcal{A}$) considered will be 2-consistent and have only binary constraints.

Since $\mathcal{A}$ is assumed to be congruence permutable, then for any $x \neq y \in V$, the binary relation R_{xy} is *rectangular*, that is, $(a, c), (a, d), (b, d) \in R_{xy}$ implies $(b, c) \in R_{xy}$ (in Lemma 4.8 we show the converse, i.e., a failure of congruence permutability implies a failure of rectangularity). 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 $x \neq y$, the congruence $\theta_{xy} = 0_{A_x}$ then the relation R_{yx} is the graph of a surjective homomorphism from $\mathbb{A}_y$ to $\mathbb{A}_x$.

3.1 A factoring reduction

Let $\mathcal{I} = (V, \{A_x\}, \{R_{xy}\}, \boldsymbol{w})$ be a 2-consistent instance of CSP($\mathcal{A}$) and for each $x \in V$ let $\mu_x = \bigwedge_{y \neq x} \theta_{xy}$, 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 $\mathcal{I}$ distinguishes values within any μ_x-class. Formally, we define the factoring reduction as in Algorithm 1.

Let $(\mathcal{I}, \epsilon, f)$ be an input of CSP($\mathcal{A}$) and let $(\mathcal{I}', \epsilon', f') = \textsc{Factor}(\mathcal{I}, \epsilon, f)$. It is clear that since the instance $\mathcal{I}$ of CSP($\mathcal{A}$) is assumed to be 2-consistent then the instance $\mathcal{I}'$ will also be 2-consistent. Furthermore, the sizes of the domains of $\mathcal{I}'$ are no larger than the sizes of the domains of $\mathcal{I}$. Now we show that the factoring reduction is a linear reduction.

Algorithm 1

procedure FACTOR($\mathcal{I} = (V, \{A_x\}, \{R_{xy}\}, \boldsymbol{w}), \epsilon, f$)
    for $x \in V$ **do**
        $A_x \leftarrow A_x/\mu_x$.
        $f(x) \leftarrow f(x)/\mu_x$.
    for $(x, y) \in V \times V$ **do**
        $R_{xy} \leftarrow \{(a/\mu_x, b/\mu_y) \mid (a, b) \in R_{xy}\}$.
    return $(\mathcal{I}, \epsilon, f)$.

LEMMA 3.1. *Let $(\mathcal{I}, \epsilon, f)$ be an input of* CSP($\mathcal{A}$) *and let $(\mathcal{I}', \epsilon', f') =$ FACTOR$(\mathcal{I}, \epsilon, f)$. If $(\mathcal{I}', \epsilon', f')$ is testable with $q(\epsilon')$ queries, then $(\mathcal{I}, \epsilon, f)$ is testable with $q(O(\epsilon))$ queries.*

PROOF. We show that the factoring reduction is a linear reduction. Let the original and reduced instances be

$$\mathcal{I} = (V, \{A_x\}, \{R_{xy}\}, \boldsymbol{w}) \text{ and } \mathcal{I}' = (V', \{A'_x\}_{x \in V}, \{R'_{xy}\}, \boldsymbol{w}')$$

respectively.

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

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

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

EXAMPLE 7. [Example 3, continued] Let $(\mathcal{I}, \epsilon, f)$ be an input of CSP($\mathbf{R}$), where

$$\mathcal{I} = (V, \{A_x\}, \{R_{xy}\}, \boldsymbol{w})$$

is a 2-consistent instance. So each A_x is equal to $A_2 \times A_3 \times A_5$ where each A_p is either F_p or $\{a\}$ for some $a \in F_p$. For any $x \in V$, μ_x will be a congruence on A_x and will be equal to the kernel of a projection map onto some of the factors of A_x. So, after applying FACTOR, the resulting instance will have domains that are isomorphic to a product of one, two, or three of the sets F_2, F_3, and F_5, with the corresponding constraints reduced accordingly. □

3.2 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 $\mathcal{A}$ and that $\mathbb{R}$ is a subdirect product of $\mathbb{A}$ and $\mathbb{B}$ for some $\mathbb{B} \in \mathcal{A}$. 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:

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

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 3.2. *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 $\mathcal{A}$ 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. □

Lemma 3.2 allows us to split a domain of an instance of CSP($\mathcal{A}$) into subdirectly irreducible domains. Formally, we define the splitting reduction as in Algorithm 2.

Algorithm 2

procedure Split($\mathcal{I} = (V, \{A_x\}, \{R_{xy}\}, \boldsymbol{w}), \epsilon, f$)
  while there exists $x \in V$ such that $\mathbb{A}_x$ is not subdirectly irreducible or trivial **do**
    Replace $\mathbb{A}_x$ in $\mathcal{I}$ with an isomorphic non-trivial subdirect 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.
    $V \leftarrow (V \setminus \{x\}) \cup \{x_1, x_2\}$, where x_1 and x_2 are newly introduced variables.
    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.
    $C \leftarrow C \setminus \{\langle (x, x), R_{xx} \rangle, \langle (x, y), R_{xy} \rangle, \langle (y, x), R_{yx} \rangle\}_{y \in V \setminus \{x\}}$.
    $C \leftarrow C \cup \{\langle (x_1, x_1), 0_{A_{x_1}} \rangle, \langle (x_2, x_2), 0_{A_{x_2}} \rangle, \langle (x_1, x_2), A_x \rangle, \langle (x_2, x_1), A_x^{-1} \rangle\}$.
    $C \leftarrow C \cup \{\langle (x_1, y), (R_{xy})_1 \rangle, \langle (x_2, y), (R_{xy})_2 \rangle, \langle (y, x_1), (R_{xy})_1^{-1} \rangle, \langle (y, x_2), (R_{xy})_2^{-1} \rangle\}_{y \in V \setminus \{x\}}$.
    Remove x from the domain of $\boldsymbol{w}$ and add x_1 and x_2.
    Set $\boldsymbol{w}(x_1) = \boldsymbol{w}(x)/2$ and $\boldsymbol{w}(x_2) = \boldsymbol{w}(x)/2$.
    Remove x from the domain of f and add x_1 and x_2.
    Set $f(x_1) \in A_{x_1}$ and $f(x_2) \in A_{x_2}$ so that $(f(x_1), f(x_2)) = f(x)$.
  return $(\mathcal{I}, \epsilon/2^{|A|}, f)$.

Let $(\mathcal{I}, \epsilon, f)$ be an input of CSP($\mathcal{A}$) and let $(\mathcal{I}', \epsilon', f') = \text{Split}(\mathcal{I}, \epsilon, f)$. It is clear that, since $\mathcal{I}$ is assumed to be a 2-consistent instance of CSP($\mathcal{A}$) then the splitting reduction constructs another 2-consistent instance $\mathcal{I}'$ of CSP($\mathcal{A}$) whose domains are all subdirectly irreducible and so have size bounded by $|A|$ (and are no bigger than the domains of $\mathcal{I}$). The next lemma shows that splitting domains of an instance does not affect the primeness of the instance's domains.

Lemma 3.3. *Let $\mathcal{I}'$ be the instance of* CSP($\mathcal{A}$) *obtained by splitting a domain $\mathbb{A}_x$ of another instance $\mathcal{I}$ 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 $\mathcal{I}$ then the domains $\mathbb{A}_{x_1}$ and $\mathbb{A}_{x_2}$ are prime in $\mathcal{I}'$. If $\mathbb{A}_y$ is some other domain of $\mathcal{I}$ then $\theta_{yx} = \theta_{yx_1} \cap \theta_{yx_2}$ and so if $\mathbb{A}_y$ is prime in $\mathcal{I}$ then it remains prime in $\mathcal{I}'$.*

Proof. Let $\mathcal{I} = (V, \{A_x\}, \{R_{xy}\}, \boldsymbol{w})$ 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 $\mathcal{I}'$ from $\mathcal{I}$ 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 $\mathcal{I}$ 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 $\mathcal{I}'$ 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 $\mathcal{A}$ 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 $\mathcal{I}'$ and by symmetry, $\mathbb{A}_{x_2}$ is also prime.

A similar use of the majority polymorphism can establish the last part of this lemma. ◻

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

Lemma 3.4. *Let $(\mathcal{I}, \epsilon, f)$ be an input of* CSP($\mathcal{A}$) *and let $(\mathcal{I}', \epsilon', f') =$ Split$(\mathcal{I}, \epsilon, f)$. If $(\mathcal{I}', \epsilon', f')$ is testable with $q(\epsilon')$ queries, then $(\mathcal{I}, \epsilon, f)$ is testable with $q(O(\epsilon))$ queries.*

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

Let $\mathcal{I} = (V, \{A_x\}, \{R_{xy}\}, \boldsymbol{w})$ and $\mathcal{I}' = (V', \{A_x'\}, \{R_{xy}'\}, \boldsymbol{w}')$ 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 $x \in V$.

We can 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 $\mathcal{I}$, then f' satisfies $\mathcal{I}'$ by Lemma 3.2. Suppose that f' is $\epsilon/(2^{|A|})$-close to satisfying $\mathcal{I}'$ and let g' be a satisfying assignment for $\mathcal{I}'$ with $\mathrm{dist}_{\mathcal{I}'}(f', g') \leq \epsilon/(2^{|A|})$. Because the tuple $(g'(x_1), \ldots, g'(x_{k_x}))$ is in A_x, we can naturally define an assignment g for $\mathcal{I}$ by setting $g(x) = (g'(x_1), \ldots, g'(x_{k_x})) \in A_x$. Then g is a satisfying assignment from Lemma 3.2. Moreover,

$$\begin{aligned}\mathrm{dist}_{\mathcal{I}}(f,g) &= \sum_{x\in V:\exists i\in[k_x],g'(x_i)\neq f'(x_i)} \boldsymbol{w}(x)\\ &\leq \sum_{x\in V}\sum_{i\in[k_x]:g'(x_i)\neq f'(x_i)} 2^{|A|}\boldsymbol{w}'(x_i)\\ &= 2^{|A|}\mathrm{dist}_{\mathcal{I}'}(f',g')\leq\epsilon.\end{aligned}$$

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

Example 8. [Example 3, continued] After applying the procedure Split each of the domains of the resulting instance will be trivial or isomorphic to F_p for some $p = 2$, 3, or 5. For variables $x \neq y$, the binary constraint from A_x to A_y will either be trivial (i.e., equal to $A_x \times A_y$) or equal to the graph of an isomorphism from A_x to A_y. Since this new instance will be reduced, then for any non-trivial A_y there will be at least one x, with the latter holding. □

3.3 Isomorphism reduction

By applying the factoring reduction and then the splitting reduction to an instance of CSP($\mathcal{A}$) 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 3.5. *Let $\mathcal{I} = (V, \{A_x\}, \{R_{xy}\}, \boldsymbol{w})$ be an instance of* CSP($\mathcal{A}$) *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 $\mathcal{I} = (V, \{A_x\}, \{R_{xy}\}, \boldsymbol{w})$ be an instance of CSP($\mathcal{A}$) 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 $\mathcal{I}$, the relation $\sim$ is naturally an equivalence relation on V. The following corollary to Lemma 3.5 establishes that unless all of the domains of $\mathcal{I}$ are trivial, the relation $\sim$ is non-trivial.

Corollary 3.6. *For $\mathcal{I} = (V, \{A_x\}, \{R_{xy}\}, \boldsymbol{w})$ an instance of* CSP($\mathcal{A}$) *as in Lemma 3.5, 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 $\mathcal{I}$, we can test the input $(\mathcal{I}, \epsilon, f)$ 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.

3.3.1 Testing $\sim$-consistency. We say that the input $(\mathcal{I}, \epsilon, f)$ is *$\sim$-consistent* if, for each $x, y \in V$ with $x \sim y$, $(f(x), f(y)) \in R_{xy}$.

For a $\sim$-class $u \subseteq V$ and $b \in A_{s(u)}$, we define

$$\overline{\boldsymbol{w}}(u, b) = \sum_{y \in u : f(y) = R_{s(u)y}(b)} \boldsymbol{w}(y),$$

$$\overline{\boldsymbol{w}}(u) = \sum_{b \in A_{s(u)}} \overline{\boldsymbol{w}}(u, b), \quad \text{and}$$

$$\overline{\boldsymbol{w}}_{\mathrm{maj}}(u) = \max_{b \in A_{s(u)}} \overline{\boldsymbol{w}}(u, b).$$

Note that $\overline{\boldsymbol{w}}(u)$ is also equal to $\sum_{x \in u} \boldsymbol{w}(x)$, the sum of the weights of the variables in u. In addition, we define ϵ_u to be $(\overline{\boldsymbol{w}}(u) - \overline{\boldsymbol{w}}_{\mathrm{maj}}(u))/\overline{\boldsymbol{w}}(u)$ and observe that $\epsilon_u \leq (|A|-1)/|A|$ since $|A_{s(x)}| \leq |A|$ and so $\overline{\boldsymbol{w}}(u)$ is the sum of at most $|A|$ terms, each of which is at most $\overline{\boldsymbol{w}}_{\mathrm{maj}}(u)$. The quantity ϵ_u represents the fraction of values, by weight, of $f|_u$ that need to be altered in order to establish $\sim$-consistency of the assignment over the class u. Let f_{maj} be the assignment obtained from f in this way. That is, for $x \in V$, $f_{\mathrm{maj}}(x) = R_{s([x])x}\left(\mathrm{argmax}_{b \in A_{s([x])}} \overline{\boldsymbol{w}}([x], b)\right)$.

We need the following simple proposition to analyze our algorithm.

Proposition 3.7. *Let X be a random variable taking values in $[0, 1]$ such that $\mathbf{E}[X] \geq \epsilon$ for some $\epsilon \geq 0$. Then, $\Pr[X \geq \epsilon/2] \geq \epsilon/2$ holds.*

Proof. Let $p = \Pr[X \geq \epsilon/2]$. Then,

$$\epsilon \leq \mathbf{E}[X] \leq 1 \cdot p + \frac{\epsilon}{2}(1 - p) \leq p + \frac{\epsilon}{2}.$$

Hence, $p \geq \epsilon/2$ holds. □

In order to test $\sim$-consistency, we run Algorithm 3.

Lemma 3.8. *Algorithm 3 tests $\sim$-consistency with query complexity $O(1/\epsilon^2)$.*

Proof. It is clear that Algorithm 3 accepts if f is $\sim$-consistent and the query complexity is $O(1/\epsilon^2)$. Suppose that f is ϵ-far from $\sim$-consistency, which means that $\mathrm{dist}_{\mathcal{I}}(f, f_{\mathrm{maj}}) \geq \epsilon$. Then, we have $\mathbf{E}[\epsilon_u] = \sum_{u : \sim\text{-class}} \overline{\boldsymbol{w}}(u)\epsilon_u \geq \epsilon$, where in the calculation of the expectation, a $\sim$-class u is chosen with probability $\overline{\boldsymbol{w}}(u)$. Note that $\epsilon_u \in [0, 1]$ for every $\sim$-class u and so we can apply Lemma 3.7, to conclude that we sample a $\sim$-class u with $\epsilon_u \geq \epsilon/2$ with probability at least $\epsilon/2$. Hence, the probability that U contains a $\sim$-class u with $\epsilon_u \geq \epsilon/2$ is at least $1 - (1 - \epsilon/2)^{\Theta(1/\epsilon)} \geq 5/6$ by choosing the

Algorithm 3

procedure CONSISTENCY($\mathcal{I}, \epsilon, f$)
  Sample a set U of $\Theta(1/\epsilon)$ $\sim$-classes of $\mathcal{I}$. In each sampling, u is chosen with probability $\overline{\boldsymbol{w}}(u)$.
  for each $u \in U$ **do**
    Let S be the set obtained by sampling $\Theta(1/\epsilon)$ variables in u with replacement. In each sampling, a variable $x \in u$ is chosen with probability $\boldsymbol{w}(x)/\overline{\boldsymbol{w}}(u)$.
    if there are two variables $x, y \in S$ with $f(y) \neq R_{xy}(f(x))$ **then**
      Reject.
  Accept.

Algorithm 4

procedure ISOMORPHISM($\mathcal{I}, \epsilon, f$)
  for each $\sim$-class u **do**
    Sample a variable $x \in u$ with probability $\boldsymbol{w}(x)/\overline{\boldsymbol{w}}(u)$, and let x_u be the sampled variable.
    $V' \leftarrow V' \cup \{u\}$.
    $A'_u \leftarrow A_{s(u)}$.
    $\boldsymbol{w}'(u) \leftarrow \overline{\boldsymbol{w}}(u)$.
    $f'(u) \leftarrow R_{x_u s(u)}(f(x_u))$.
  for each pair (u, u') of $\sim$-classes **do**
    $R'_{uu'} \leftarrow R_{x_u x_{u'}}$.
  return $((V', \{A'_x\}, \{R'_{xy}\}, \boldsymbol{w}'), \epsilon/2, f')$.

hidden constant large enough. For a $\sim$-class u with $\epsilon_u \geq \epsilon/2$, the probability that we find two vertices $x, y \in u$ with $f(y) \neq R_{xy}(f(x))$ in S is at least

$$1 - (1 - \epsilon_u)^{\Theta(1/\epsilon)} - (\epsilon_u)^{\Theta(1/\epsilon)} \geq 1 - (1 - \epsilon/2)^{\Theta(1/\epsilon)} - ((|A| - 1)/|A|)^{\Theta(1/\epsilon)} \quad (1)$$

since $\epsilon_u \geq \epsilon/2$ for this class u and, as noted earlier, $\epsilon_u \leq (|A| - 1)/|A|$ for every class u. By choosing the hidden constant large enough we can ensure that (1) is at least 5/6. By combining these bounds, we obtain two vertices x, y with $f(y) \neq R_{xy}(f(x))$ with probability at least 2/3. □

3.3.2 Isomorphism reduction. Using Algorithm 3, we can reject an input $(\mathcal{I}, \epsilon, f)$ if it is far from satisfying $\sim$-consistency. In this subsection we will consider a reduction from $(\mathcal{I}, \epsilon, f)$ to another input $(\mathcal{I}', \epsilon', f')$ assuming that it has not been rejected by Algorithm 3.

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 $\mathcal{I}$ of CSP($\mathcal{A}$) is assumed to be 2-consistent, the reduction will produce another 2-consistent instance $\mathcal{I}'$ of CSP($\mathcal{A}$). As the next lemma shows, unless the domains of $\mathcal{I}$ all have size one, some of the domains of $\mathcal{I}'$ will no longer be prime.

LEMMA 3.9. *Let $(\mathcal{I}, \epsilon, f)$ be an input of CSP($\mathcal{A}$) for which domains of $\mathcal{I}$ are either trivial or prime and subdirectly irreducible and let $(\mathcal{I}', \epsilon', f') =$ ISOMORPHISM($\mathcal{I}, \epsilon, f$).*

If some domain of $\mathcal{I}$ has more than one element, then any domain of $\mathcal{I}'$ of maximal size will not be prime, unless $\mathcal{I}'$ has only one variable.

Proof. Suppose that $\mathcal{I}'$ has more than one variable. This is equivalent to there being more than one $\sim$-class for $\mathcal{I}$. Let x be a variable of $\mathcal{I}'$ with $|A_x|$ of maximal size and let y be any other variable of $\mathcal{I}'$. Note that according to the construction of $\mathcal{I}'$ from $\mathcal{I}$, both x and y are also variables of $\mathcal{I}$ with $x \nsim y$. Furthermore, $|A_x|$ has maximal size amongst all of the domains of $\mathcal{I}$ 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 $x \nsim y$. 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 $\mathcal{I}'$. □

Lemma 3.10. *Let $(\mathcal{I}, \epsilon, f)$ be an input of* CSP($\mathcal{A}$) *and suppose that f is $\epsilon/20$-close to satisfying $\sim$-consistency. Let $(\mathcal{I}', \epsilon', f') =$ Isomorphism$(\mathcal{I}, \epsilon, f)$. If $(\mathcal{I}', \epsilon', f')$ is testable with $q(\epsilon')$ queries, then $(\mathcal{I}, \epsilon, f)$ is testable with $q(O(\epsilon))$ queries.*

Proof. We show that the reduction in Algorithm 4 is a linear reduction. Let $\mathcal{I} = (V, \{A_x\}, \{R_{xy}\}, \boldsymbol{w})$ and $\mathcal{I}' = (V', \{A'_x\}, \{R'_{xy}\}, \boldsymbol{w}')$ be the original instance and the reduced instance, respectively.

Note that $|V'| \leq |V|$ and we can determine the value of $f'(u)$ by querying $f(x_u)$. Also, if f satisfies $\mathcal{I}$, then it is clear that f' satisfies $\mathcal{I}'$.

We want to show that, if f is far from satisfying $\mathcal{I}$, then f' is also far from satisfying $\mathcal{I}'$ with high probability. To this end, we first show that the following quantity is small with high probability:

$$\mathrm{dist}(f, f') := \sum_{u:\sim\text{-class}} \sum_{\substack{x \in u: \\ f'(u) \neq R_{xs(u)}(f(x))}} \boldsymbol{w}(x).$$

For a $\sim$-class u, we define

$$\mathrm{dist}_u(f, f') := 1 - \frac{\overline{\boldsymbol{w}}(u, f'(u))}{\overline{\boldsymbol{w}}(u)} = \sum_{\substack{x \in u: \\ f'(u) \neq R_{xs(u)}(f(x))}} \frac{\boldsymbol{w}(x)}{\overline{\boldsymbol{w}}(u)}.$$

Note that we have $\mathrm{dist}(f, f') = \sum_{u:\sim\text{-class}} \overline{\boldsymbol{w}}(u)\mathrm{dist}_u(f, f')$.

Then for any $\sim$-class u,

$$\begin{aligned}
\mathop{\mathbf{E}}_{x_u}[\mathrm{dist}_u(f, f')] &= \sum_{b \in A_{s(u)}} \frac{\overline{\boldsymbol{w}}(u, b)}{\overline{\boldsymbol{w}}(u)} \left(1 - \frac{\overline{\boldsymbol{w}}(u, b)}{\overline{\boldsymbol{w}}(u)}\right) \\
&\leq \frac{\overline{\boldsymbol{w}}_{\mathrm{maj}}(u)}{\overline{\boldsymbol{w}}(u)} \left(1 - \frac{\overline{\boldsymbol{w}}_{\mathrm{maj}}(u)}{\overline{\boldsymbol{w}}(u)}\right) + \left(1 - \frac{\overline{\boldsymbol{w}}_{\mathrm{maj}}(u)}{\overline{\boldsymbol{w}}(u)}\right) \cdot 1 \\
&\leq 2\left(1 - \frac{\overline{\boldsymbol{w}}_{\mathrm{maj}}(u)}{\overline{\boldsymbol{w}}(u)}\right) = 2\epsilon_u.
\end{aligned}$$

Thus, $\mathbf{E}_{\{x_u\}_{u:\sim\text{-class}}}[\mathrm{dist}(f, f')]$ is equal to

$$\mathop{\mathbf{E}}_{\{x_u\}}\Big[\sum_{u:\sim\text{-class}} \overline{\boldsymbol{w}}(u)\mathrm{dist}_u(f, f')\Big] \leq \sum_{u:\sim\text{-class}} 2\overline{\boldsymbol{w}}(u)\epsilon_u \leq \frac{\epsilon}{10}.$$

From Markov's inequality, we have $\Pr_{\{x_u\}}[\mathrm{dist}(f, f') \geq \epsilon/2] \leq 1/20$.

Algorithm 5

```
procedure ISOMORPHISM′(𝓘, ε, f)
    if CONSISTENCY(𝓘, ε/20, f) rejects then
        Reject.
    else
        return ISOMORPHISM(𝓘, ε, f)
```

Let g' be a satisfying assignment for $\mathcal{I}'$ closest to f'. We define an assignment g for $\mathcal{I}$ as $g(x) = R_{s([x])x}(g'([x]))$. It is clear that g is a satisfying assignment. Since we have $\mathrm{dist}(f, f') + \mathrm{dist}(f', g') \geq \mathrm{dist}(f, g) \geq \epsilon$, it follows that $\Pr[\mathrm{dist}(f', g') \geq \epsilon/2] \geq 19/20$.

To summarize, the isomorphism reduction is a gap-preserving local reduction with $t(n) \leq n$, $c_1 = 1/2$, and $c_2 = 1$. □

Example 9. [Example 3, continued] The $\sim$-classes of the current version of our instance will consist of domains that are pairwise isomorphic to each other via the corresponding constraint relations. After performing the Isomorphism reduction on this instance, we will end up with an instance whose constraint relations are trivial, i.e., for variables $x \neq y$, $R_{xy} = A_x \times A_y$. After further reducing this instance via the Factor reduction, we will end up with an instance whose domains all have size equal to one. □

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

Lemma 3.11. *Let $(\mathcal{I}, \epsilon, f)$ be an input of* CSP($\mathcal{A}$) *and suppose that Isomorphism′$(\mathcal{I}, \epsilon, f)$ returned another instance $(\mathcal{I}', \epsilon', f')$. If $(\mathcal{I}', \epsilon', f')$ is testable with $q(\epsilon')$ queries, then $(\mathcal{I}, \epsilon, f)$ is testable with $q(O(\epsilon))$ queries.*

Proof. Consider Algorithm 5. If f satisfies $\mathcal{I}$, then the $\sim$-consistency test always accepts, and hence we always accept with probability $2/3$ from Lemma 3.10. Suppose that f is ϵ-far from satisfying $\mathcal{I}$. If f is $\epsilon/20$-far from satisfying $\sim$-consistency, then the $\sim$-consistency test rejects with probability at least $2/3$. If f is $\epsilon/20$-close to satisfying $\sim$-consistency, then we reject with probability at least $2/3$ by Lemma 3.10. □

3.4 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($\mathcal{A}$).

Lemma 3.12. *Let $(\mathcal{I}, \epsilon, f)$ be an input of* CSP($\mathcal{A}$)*, and suppose that Shrink$(\mathcal{I}, \epsilon, f)$ returned another instance $(\mathcal{I}', \epsilon', f')$. If we can test $(\mathcal{I}', \epsilon', f')$ with $q(\epsilon')$ queries, then we can test $(\mathcal{I}, \epsilon, f)$ with $q(O(\epsilon))$ 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.*

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 3.9 that the maximal size of the domains of the output instance will be

Algorithm 6

---

procedure SHRINK($\mathcal{I}, \epsilon, f$)
  $(\mathcal{I}, \epsilon, f) \leftarrow$ FACTOR($\mathcal{I}, \epsilon, f$).
  $(\mathcal{I}, \epsilon, f) \leftarrow$ SPLIT($\mathcal{I}, \epsilon, f$).
  if ISOMORPHISM′($\mathcal{I}, \epsilon, f$) rejects **then**
    Reject.
  else
    $(\mathcal{I}, \epsilon, f) \leftarrow$ the input returned by ISOMORPHISM′.
  $(\mathcal{I}, \epsilon, f) \leftarrow$ FACTOR($\mathcal{I}, \epsilon, f$).
  return $(\mathcal{I}, \epsilon, f)$.

---

smaller than that of the original instance, as long as the output instance has more than one variable. □

THEOREM 3.13. *Let* $\mathbf{A}$ *be a structure that has majority and Maltsev polymorphisms. Then,* 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. □

4 NON CONSTANT-QUERY TESTABILITY

In this section we consider structures $\mathbf{A}$ that do not have a majority polymorphism or do not have a Maltsev polymorphism. As noted in the previous section, this is the same as the variety $\mathcal{V}(\mathrm{Alg}(\mathbf{A}))$ failing to be arithmetic. For such structures we will show that CSP($\mathbf{A}$) is not constant-query testable:

THEOREM 4.1. *If the relational structure* $\mathbf{A}$ *does not have a majority polymorphism or does not have a Maltsev polymorphism, then* CSP($\mathbf{A}$) *is not constant-query testable.*

PROOF. From [20] we know that for a structure $\mathbf{A}$, having both majority and Maltsev polymorphisms is equivalent to $\mathcal{V}(\mathrm{Alg}(\mathbf{A}))$ being congruence meet semidistributive and congruence permutable. The former and latter cases are handled by Theorems 4.7 (Section 4.1) and 4.11 (Section 4.2), respectively. □

4.1 Hardness for the non congruence meet semidistributive case

Suppose that $\mathcal{V}(\mathrm{Alg}(\mathbf{A}))$ is not congruence meet semidistributive. We define the *singleton-expansion* of $\mathbf{A}$ to be $\mathbf{A}' = (A, \Gamma \cup \{\{a\} \mid a \in A\})$. We first observe that CSP($\mathbf{A}'$) will be sublinear-query testable if CSP($\mathbf{A}$) is. Although this observation for the Boolean case was already given in Lemma 5 of [6], its proof was not published yet, and hence we provide the proof for the general case here for completeness.

LEMMA 4.2. *Let* $\mathbf{A}'$ *be the singleton-expansion of* $\mathbf{A}$*. Assume that* $\epsilon \ll \frac{1}{2|A|}$*. If* CSP($\mathbf{A}$) *is testable with* $q(n, \epsilon)$ *queries, then* CSP($\mathbf{A}'$) *is testable with* $q(O(n), O(\epsilon)) + \Theta(1/\epsilon)$ *queries.*

PROOF. Suppose we can test CSP($\mathbf{A}$) with $q(n, \epsilon)$ queries. Given an instance $\mathcal{I}' = (V', A, C', \mathbf{w}')$ of CSP($\mathbf{A}'$), $\epsilon \ll \frac{1}{2|A|}$, and a query access to an assignment $f' : V' \to A$, we want to test whether f' is a satisfying assignment or is ϵ-far from being so. For $a \in A$, define $V_a \subseteq V'$ to be the set of all variables v for which there is a unary constraint $((v), \{a\})$ in C'. We assume

$$\sum \{\boldsymbol{w}(v) \mid a \in A, v \in V_a, f'(v) \neq a\} \leq \epsilon \tag{2}$$

as otherwise we can reject f' with high probability by sampling $\Theta(1/\epsilon)$ variables uniformly at random.

Now, we define a set of variables $V = (V' \setminus \bigcup_{a \in A} V_a) \cup \{x_a\}_{a \in A}$ and define a set of constraints C by removing from C' all unary constraints and by identifying all variables in V_a with a new variable x_a for each $a \in A$. Next, we define $\boldsymbol{w} : V \to [0, 1]$ by $\boldsymbol{w}(v) = \boldsymbol{w}'(v)/(1 + 2\epsilon|A|)$ for each $v \in V' \setminus \bigcup_{a \in A} V_a$ and $\boldsymbol{w}(x_a) = 2\epsilon$ for each $a \in A$. Let $\mathcal{I} = (V, A, C, \boldsymbol{w})$ be an instance of CSP($\mathbf{A}$).

Now given an assignment $f' : V' \to A$ to the variables of $\mathcal{I}'$, define an assignment $f : V \to A$ to the variables of $\mathcal{I}$ by setting $f(v) = f'(v)$ for each $v \in V' \setminus \bigcup_{a \in A} V_a$ and $f(x_a) = a$ for each $a \in A$. Clearly, if f' satisfies $\mathcal{I}'$, then f satisfies $\mathcal{I}$. On the other hand, suppose f is ϵ-close to a satisfying assignment $\tilde{f}$ for $\mathcal{I}$. Then, we must have $\tilde{f}(x_a) = a$ for every $a \in A$ from our choice of $\boldsymbol{w}(x_a)$. Define $\tilde{f}' : V' \to A$ by setting $\tilde{f}'(v) = \tilde{f}(v)$ for every $v \in V' \setminus \bigcup_{a \in A} V_a$ and $\tilde{f}'(v) = a$ for each $a \in A$ and $v \in V_a$. Then, $\tilde{f}'$ satisfies $\mathcal{I}'$. From the assumption (2), the distance between f' and $\tilde{f}'$ is at most $\epsilon(1 + 2\epsilon|A|) + \epsilon \leq 3\epsilon$. Thus, we have a gap-preserving local reduction from CSP($\mathbf{A}'$) to CSP($\mathbf{A}$), and so, Lemma 2.1 finishes the proof. □

By adding all of the unary singleton relations to $\mathbf{A}$ to produce $\mathbf{A}'$ it follows that the variety $\mathcal{V}(\mathrm{Alg}(\mathbf{A}'))$ is idempotent and will also not be congruence meet semidistributive. This is because whether or not an algebra generates a congruence meet semidistributive variety depends solely on its idempotent term operations (see Theorem 8.1 of [24]). For such a structure, the following is known:

Lemma 4.3. *Let $\mathbf{A}'$ be the structure as above. Then, there is some finite algebra $\mathbb{B}$ in $\mathcal{V}(\mathrm{Alg}(\mathbf{A}'))$ and some subuniverse γ of $\mathbb{B}^3$ whose domain can be identified with $\mathbb{F}_{p^k}^{\ell}$ for some prime p and integers $k, \ell \geq 1$ such that $\gamma = \{a + b + c = 0 \mid a, b, c \in \mathbb{F}_{p^k}^{\ell}\}$.*

Proof. A combination of Theorem 4.3 and Proposition 2.1 from [18] implies that there is some finite algebra $\mathbb{B}$ in $\mathcal{V}(\mathrm{Alg}(\mathbf{A}'))$ (in fact $\mathbb{B}$ will be isomorphic to a quotient of a subalgebra of $\mathrm{Alg}(\mathbf{A}')$) that is either term equivalent to the algebra with universe $\{0, 1\}$ having no basic operations or is term equivalent to the idempotent reduct of a module over a finite ring. Theorem 2.1 of [30] provides more detail on this module: it can be taken to be the module $\mathbb{F}_{p^k}^{\ell}$ over the ring of $\ell \times \ell$ matrices over the finite field $\mathbb{F}_{p^k}$ for some prime number p and some integers $k, \ell \geq 1$. In this case, $\gamma = \{a + b + c = 0 \mid a, b, c \in \mathbb{F}_{p^k}^{\ell}\}$ is a subuniverse of $\mathbb{B}^3$.

In the first case where $\mathbb{B}$ is term equivalent to the algebra with universe $\{0, 1\}$ having no basic operations, $\gamma = \{a + b + c = 0 \mid a, b, c \in \mathbb{F}_2\}$ is a subuniverse of $\mathbb{B}^3$ since every subset of B^3 will be a subuniverse. □

We now establish a linear lower bound for $CSP((B; \gamma))$ for B and γ as in Lemma 4.3.

We first show a linear lower bound for the case that p is an arbitrary prime and $k = \ell = 1$ by extending the argument for $p = 2$ and $k = \ell = 1$ due to Ben-Sasson *et al.* [5].

To this end, we introduce some definitions. For a vertex set S in a graph, let $N^1(S)$ be the set of its *unique neighbors*, that is, vertices with exactly one neighbor in S. For $\lambda, \gamma > 0$, we say that a bipartite graph $(L, R; E)$ is a (λ, γ)*-right unique neighbor expander* if $|N^1(S)| > \lambda |S|$ holds for any $S \subseteq R$ with $|S| \leq \gamma n$. For a CSP instance $\mathcal{I} = (V, A, \mathcal{C}, \boldsymbol{w})$, we define its *primal graph* $G(\mathcal{I})$ as the bipartite graph $(V, \mathcal{C}; E)$ such that the pair $(v, C) \in V \times \mathcal{C}$ belongs to E if and only if v is in the scope of C. Now, we show the following:

Lemma 4.4. *Let p be a prime and $\mathbf{B} = (\mathbb{F}_p; \Gamma)$ be a constraint language such that Γ contains a relation $\{(a, b, c) \mid a + b + c = 0\}$. Then, testing* CSP($\mathbf{B}$) *requires a linear number of queries even when the primal graph of the input instance is restricted to be a (λ, γ)-right unique neighbor expander for some universal constant λ and γ.*

Proof. As the proof is almost identical to that for the case $p = 2$ in [5], we only highlight the difference.

Showing the hardness for the case $p = 2$ amounts to finding a subspace of $U \subseteq \mathbb{F}_2^n$ such that the basis $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m\}$ of the dual space $U^\perp$ satisfies the following properties:

- The basis is ϵ*-separating*, that is, every $\boldsymbol{x} \in \mathbb{F}_2^n$ with a unique $i \in [m]$ satisfying $\langle \boldsymbol{a}_i, \boldsymbol{x} \rangle \neq 0$ has $|\boldsymbol{x}| \geq \epsilon n$.
- The basis is (q, μ)*-local*, that is, every $\alpha \in \mathbb{F}_2^n$ that is a sum of at least μm vectors in the basis has $|\boldsymbol{\alpha}| \geq q$,

Here, $\epsilon > 0$ is the error parameter, and we want to choose $q = \Omega(n)$ and μ to be a small constant, say, $1/10$. We can reuse this argument for our case by changing the Hamming weights $|\boldsymbol{x}|$ and $|\boldsymbol{\alpha}|$ with the ℓ_0 norms $\|\boldsymbol{x}\|_0$ and $\|\boldsymbol{\alpha}\|_0$, that is, the numbers non-zero elements in those vectors.

Ben-Sasson *et al.* [5] constructed such a subspace from a random regular bipartite graph. More specifically, given a bipartite graph $G = (L, R; E)$, they constructed a CSP instance on the variable set L with a constraint of the form $\sum_{u \in N(v)} u = 0 \pmod 2$ for each right vertex $v \in R$, where $N(v) \subseteq L$ is the set of neighbors of v. We use the same construction by regarding u as an element in $\mathbb{F}_p$ instead of $\mathbb{F}_2$.

For a vertex set S, let $N^p(S)$ be the set of neighbors of S whose neighbors in S is non-zero modulo p. For $\lambda, \gamma > 0$, we say that a bipartite graph $(L, R; E)$ is a (λ, γ)*-right p-expander* if $|N^p(S)| > \lambda |S|$ for any $S \subseteq R$ with $|S| \geq \gamma n$. Ben-Sasson *et al.* [5] showed that the obtained subspace is ϵ-separating and (q, μ)-local by using the fact that a random regular bipartite graph $(L, R; E)$ is a (λ, γ)-right 2-expander with high probability for some suitable λ and γ. Similarly, we can show that a random regular bipartite graph is a (λ, γ)-right p-expander with high probability and that the obtained subspace is ϵ-separating and (q, μ)-local. □

Next, we generalize Lemma 4.4 to the case that $k \geq 1$.

Lemma 4.5. *Let p be a prime, $k \geq 1$ be an integer, and $\mathbf{B} = (\mathbb{F}_{p^k}; \Gamma)$ be a constraint language such that Γ contains a relation $\{(a, b, c) \mid a + b + c = 0\}$. Then, testing* CSP($\mathbf{B}$) *requires a linear number of queries even when the primal graph of the input instance is restricted to be a (λ, γ)-right unique neighbor expander for some universal constant λ and γ.*

Proof. Let $\mathbf{B}' = (\mathbb{F}_p; \{\{(a, b, c) \mid a + b + c = 0\}\})$, which is hard to test even if the primal graph of the instance is a (λ, γ)-right unique neighbor expander for some

$\lambda, \gamma > 0$ by Lemma 4.4. We show a gap-preserving local reduction from CSP($\mathbf{B}'$) to CSP($\mathbf{B}$).

Given an instance $\mathcal{I}' = (V', \mathbb{F}_p, C', \boldsymbol{w}')$ of CSP($\mathbf{B}'$) such that the primal graph $G(\mathcal{I}')$ is a (λ, γ)-right unique neighbor expander, we construct an instance $\mathcal{I} = (V, \mathbb{F}_{p^k}, C, \boldsymbol{w})$, where $V = V', C = C'$ (after changing the domain from $\mathbb{F}_p$ to $\mathbb{F}_{p^k}$), and $\boldsymbol{w} = \boldsymbol{w}'$. A value in $\mathbb{F}_{p^k}$ can be identified with a vector in $\mathbb{F}_p^k$, where addition in $\mathbb{F}_{p^k}$ is coordinate-wise addition in $\mathbb{F}_p^k$. Now given an assignment $f' : V \to \mathbb{F}_p$ to the variables of $\mathcal{I}'$, define an assignment $f : V \to \mathbb{F}_{p^k}$ to the variables of $\mathcal{I}$ by setting $f(v) = (f'(v), 0, \ldots, 0) \in \mathbb{F}_p^k$ for every $v \in V$. Clearly, if f' satisfies $\mathcal{I}'$, then f satisfies $\mathcal{I}$. On the other hand, suppose f is ϵ-close to a satisfying assignment $\tilde{f}$ for $\mathcal{I}$. Let $S = \{v \in V \mid \exists i > 1 \text{ with } \tilde{f}(v)(i) \neq 0\}$. Then, if a constraint of the form $x + y + z = 0$ involves a variable in S, then we must have another variable in S in the constraint. This violates the fact that $G(\mathcal{I}')$ is (λ, γ)-right unique neighbor expander (in the regime $\epsilon < \gamma$), and hence $S = \emptyset$ holds. Then, we can naturally recover a satisfying assignment $\tilde{f}'$ for $\mathcal{I}'$ from $\tilde{f}$ by setting $\tilde{f}'(v) = \tilde{f}(v)(1)$, which is ϵ-close to f'. □

We further generalize to the case that $\ell \geq 1$. We omit the proof because it is almost identical to that of Lemma 4.5.

Lemma 4.6. *Let p be a prime, $k, \ell \geq 1$ be integers, and $\mathbf{B} = (\mathbb{F}_{p^k}^{\ell}; \Gamma)$ be a constraint language such that Γ contains a relation $\{(a, b, c) \mid a + b + c = 0\}$. Then, testing CSP($\mathbf{B}$) requires a linear number of queries.*

Theorem 4.7. *Let $\mathbf{A}$ be a relational structure such that $\mathcal{V}(\mathrm{Alg}(\mathbf{A}))$ is not congruence meet semidistributive. Then, testing CSP($\mathbf{A}$) requires a linear number of queries.*

Proof. Immediate from Lemmas 2.2, 4.2, 4.3, and 4.6. □

4.2 Hardness for the non congruence permutable case

Now, we consider the case that $\mathcal{V}(\mathrm{Alg}(\mathbf{A}))$ is not congruence permutable. We use the following well-known fact.

Lemma 4.8. *Let $\mathbf{A}$ be a finite relational structure that does not have a Maltsev polymorphism. Then, there is some finite algebra $\mathbb{B}$ in $\mathcal{V}(\mathrm{Alg}(\mathbf{A}))$ and some subuniverse γ of $\mathbb{B}^2$ such that there are elements $0, 1 \in B$ with $(0, 0), (0, 1), (1, 1) \in \gamma$ and $(1, 0) \notin \gamma$.*

Proof. Since $\mathbf{A}$ does not have a Maltsev polymorphism, then $\mathcal{V}(\mathrm{Alg}(\mathbf{A}))$ is not congruence permutable and so there is some finite algebra $\mathbb{B} \in \mathcal{V}(\mathrm{Alg}(\mathbf{A}))$ having congruences α and β such that $\alpha \circ \beta \neq \beta \circ \alpha$. We may assume that $\alpha \circ \beta \not\subseteq \beta \circ \alpha$ and so there will be elements $0, 1 \in B$ with $(0, 1) \in \alpha \circ \beta$ but $(1, 0) \notin \alpha \circ \beta$. Since $\alpha \circ \beta$ is a reflexive relation, then setting $\gamma = \alpha \circ \beta$ works. □

We now establish a super-constant lower bound for CSP($(B; \gamma)$) for B and γ as in Lemma 4.8 based on the super-constant lower bound for monotonicity testing given in [17]. We first note that it is not clear whether we can directly reduce monotonicity testing to testing CSP($(B; \gamma)$) to obtain a super-constant lower bound for the latter problem. The reason is that B may have more than two elements and γ may have satisfying assignments other than $(0, 0), (0, 1)$, and $(1, 1)$, which makes hard to preserve

ϵ-farness through the reduction. Hence, although our proof is almost identical to the one given in [17], we include it here for completeness.

Let $G = (V; E)$ be an undirected graph and let $M \subseteq E$ be a matching in G, i.e., no two edges in M have a vertex in common. Let $V(M)$ be the set of the endpoints of edges in M. A matching M is called *induced* if the subgraph induced by $V(M)$ contains only the edges of M. A bipartite graph $G = (X, Y; E)$ is called (s, t)*-Ruzsa-Szemerédi* if its edge set can be partitioned into at least s induced matchings $M_1, \ldots, M_s$, each of size at least t.

Lemma 4.9 (Theorem 16 of [17]). *There exist an* $(n^{\Omega(1/\log\log n)}, n/3 - o(n))$*-Ruzsa-Szemerédi graph* $G = (X, Y; E)$ *with* $|X| = |Y| = n$.

Lemma 4.10. *Let* $\mathbf{B} = (B; \gamma)$ *where* γ *is a binary relation such that for some* $0, 1 \in B$, $(0, 1)$, $(0, 0)$, *and* $(1, 1) \in \gamma$ *but* $(1, 0) \notin \gamma$. *Then,* $\mathrm{CSP}(\mathbf{B})$ *is not constant-query testable.*

Proof. If $\mathrm{CSP}(\mathbf{B})$ is testable with q queries, then $\mathrm{CSP}(\mathbf{B})$ is non-adaptively testable with $|B|^q$ queries. Hence, in order to show that $\mathrm{CSP}(\mathbf{B})$ is not constant-query testable, it suffices to show that $\mathrm{CSP}(\mathbf{B})$ is not constant-query testable non-adaptively.

Let $G = (X, Y; E)$ be an $(s, n/3 - o(n))$-Ruzsa-Szemerédi graph provided as in Lemma 4.9, where $s = n^{\Omega(1/\log\log n)}$. Then, we construct an instance $\mathcal{I} = (V, C, \boldsymbol{w})$ of $\mathrm{CSP}(\mathbf{B})$, where $V = X \cup Y$, $C = \{\langle (x, y), \gamma \rangle \mid (x, y) \in E\}$, and $\boldsymbol{w}(x) = 1/|V|$ for all $x \in V$.

We use Yao's principle, which states that to establish a lower bound on the complexity of a randomized test, it is enough to present an input distribution on which any deterministic test with that complexity is likely to fail. Namely, we define distributions D_P, D_N on positive (satisfying) and negative (far from satisfying) assignments, respectively. Our assignment distribution first chooses D_P or D_N with equal probability and then draws an assignment according to the chosen distribution. We show that every deterministic non-adaptive test with $q = o(\sqrt{s})$ queries has error probability larger than $1/3$ (with respect to the induced probability on assignments).

We now define the distributions D_P and D_N, as well as the auxiliary distribution $\widetilde{D}_N$. For D_P and D_N, choose a random $i \in \{1, \ldots, s\}$ uniformly. For all variables $x \in X$ and $y \in Y$ outside of matching M_i, set $f(x) = 0$ and $f(y) = 1$. For D_P, uniformly choose $f(x) = f(y) = 0$ or $f(x) = f(y) = 1$ independently for all edges $(x, y) \in M_i$. For $\widetilde{D}_N$, uniformly choose $f(x) = 1 - f(y) = 0$ or $f(x) = 1 - f(y) = 1$ independently for all $(x, y) \in M_i$.

Note that D_P is supported only on positive assignments, but $\widetilde{D}_N$ is not supported only on negative assignments. However, for n large enough, with probability more than $8/9$ at least $1/3$ of the constraints on the edges of M_i are violated when the assignment is chosen according to $\widetilde{D}_N$, making the assignment $\Omega(1)$-far from satisfying $\mathcal{I}$. Denote the latter event by F and define $D_N = \widetilde{D}_N|_F$, namely, D_N is $\widetilde{D}_N$ conditioned on the event F. Note that for $\widetilde{D}_N$, a constraint is violated only if it belongs to M_i, since the matchings are induced.

Given a deterministic non-adaptive test that makes a set V' of q queries, the probability that one or more of the edges of M_i have both endpoints in V' is at most $q^2/(4s)$ for both D_P and $\widetilde{D}_N$. This is because the matchings are disjoint, and the vertex set V' induces at most $q^2/4$ edges of G. For $q = o(\sqrt{s})$, with probability more than $1 - o(1)$, no edge of M_i has both endpoints in V'. Conditioned on any choice of i for which M_i has no such edge, the distribution of $f|_{V'}$ is identical for both $\widetilde{D}_N$ and D_P: every vertex outside of M_i is fixed to 0 if it is in X and to 1 if it is in Y, and the value of every other vertex is uniform and independent over $\{0, 1\}$. This means that the error probability under the above conditioning (with negative assignments chosen under $\widetilde{D}_N$ rather than D_N) is $1/2$.

As the probability of the condition is at least $1 - o(1)$, the overall error probability without the conditioning is at least $1/2 - o(1)$. Since negative assignments are chosen under D_N, not $\widetilde{D}_N$, the success probability is $(1/2+o(1))\cdot(\Pr[F]^{-1}) \leq (1/2+o(1))\cdot 9/8 \leq 9/16 + o(1)$. Thus, the error probability is $\geq 7/16 - o(1)$. □

Theorem 4.11. *Let* $\mathbf{A}$ *be a relational strcture such that* $\mathcal{V}(\mathrm{Alg}(\mathbf{A}))$ *is not congruence permutable. Then, testing* $\mathrm{CSP}(\mathbf{A})$ *requires a linear number of queries.*

Proof. Immediate from Lemmas 4.8 and 4.10. □

5 DISCUSSION

Theorem 1.1 characterizes the relational structures $\mathbf{A}$ on general domains for which $\mathrm{CSP}(\mathbf{A})$ is constant-query testable. Obtaining a characterization for the sublinear-query testable case is a tantalizing open problem. In [15] we succeed in settling this for a closely related problem, $\exists\mathrm{CSP}(\mathbf{A})$, whose instances may include existentially quantified variables. Our characterization makes use of the following generalization of a majority operation (see Definition 1).

Definition 3. *For a nonempty set A and $k \geq 3$, an operation $n : A^k \to A$ is a k-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.)

In [15] we establish the following trichotomy:

(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-ary near-unanimity polymorphism for some $k \geq 3$, 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.
(3) Else, testing $\exists\mathrm{CSP}(\mathbf{A})$ with one-sided error requires a linear number of queries.

The third item above was obtained by reducing the problem of testing assignments of monotone circuits to $\exists$CSPs. If we do not allow existentially quantified variables, then the number of variables blows up polynomially, in the reduction, and a linear lower bound for monotone circuits does not imply a linear lower bound for CSPs.

The above trichotomy for $\exists$CSPs is in terms of the number of queries needed to test with one-sided error. Obtaining a similar trichotomy for two-sided error testers is also an interesting open problem. Again the obstacle is that we reduce from the problem of testing assignments of monotone circuits. It is not clear whether this problem is hard also for two-sided error testers.

REFERENCES

[1] Kirby A Baker and Alden F Pixley. 1975. Polynomial interpolation and the Chinese Remainder Theorem for algebraic systems. *Mathematische Zeitschrift* 143, 2 (1975), 165–174.

[2] L Barto. 2014. The collapse of the bounded width hierarchy. *Journal of Logic and Computation* 26, 3 (2014), 923–943.

[3] Libor Barto. 2015. The constraint satisfaction problem and universal algebra. *The Bulletin of Symbolic Logic* 21 (9 2015), 319–337. Issue 03.

[4] Libor Barto and Marcin Kozik. 2014. Constraint Satisfaction Problems Solvable by Local Consistency Methods. *J. ACM* 61, 1 (2014), 3.

[5] Eli Ben-Sasson, Prahladh Harsha, and Sofya Raskhodnikova. 2005. Some 3CNF Properties Are Hard to Test. *SIAM J. Comput.* 35, 1 (2005), 1–21.

[6] Arnab Bhattacharyya and Yuichi Yoshida. 2013. An Algebraic Characterization of Testable Boolean CSPs. In *Proceedings of the 40th International Colloquium on Automata, Languages, and Programming (ICALP)*. 123–134.

[7] A. Bulatov. 2017. A dichotomy theorem for nonuniform CSPs. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. to appear.

[8] Andrei Bulatov and Dániel Marx. 2010. The complexity of global cardinality constraints. *Logical Methods in Computer Science* 6 (2010), 1–27.

[9] Andrei A. Bulatov. 2011. Complexity of conservative constraint satisfaction problems. *ACM Transactions on Computational Logic* 12, 4 (2011), 24.

[10] Andrei A. Bulatov. 2013. The complexity of the counting constraint satisfaction problem. *J. ACM* 60, 5 (2013), 34.

[11] Stanley Burris and H. P. Sankappanavar. 1981. *A course in universal algebra*. Graduate Texts in Mathematics, Vol. 78. Springer-Verlag, New York-Berlin. xvi+276 pages.

[12] Siu On Chan, James R. Lee, Prasad Raghavendra, and David Steurer. 2013. Approximate Constraint Satisfaction Requires Large LP Relaxations. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 350–359.

[13] Hubie Chen and Benoit Larose. 2017. Asking the Metaquestions in Constraint Tractability. *ACM Trans. Comput. Theory* 9, 3, Article 11 (Oct. 2017), 27 pages. https://doi.org/10.1145/3134757

[14] Hubie Chen and Matthew Valeriote. 2015. Learnability of Solutions to Conjunctive Queries: The Full Dichotomy. In *Proceedings of the 28th Conference on Learning Theory (COLT)*. 326–337.

[15] H. Chen, M. Valeriote, and Y. Yoshida. 2016. Testing Assignments to Constraint Satisfaction Problems. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 525–534.

[16] Martin E. Dyer and David Richerby. 2013. An Effective Dichotomy for the Counting Constraint Satisfaction Problem. *SIAM J. Comput.* 42, 3 (2013), 1245–1274.

[17] Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, Ronitt Rubinfeld, and Alex Samorodnitsky. 2002. Monotonicity testing over general poset domains. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*. 474–483.

[18] Ralph Freese and Matthew A. Valeriote. 2009. On the complexity of some Maltsev conditions. *Internat. J. Algebra Comput.* 19, 1 (2009), 41–77.

[19] Oded Goldreich (Ed.). 2011. *Property Testing*. Lecture Notes in Computer Science, Vol. 6390. Springer Berlin Heidelberg, Berlin, Heidelberg.

[20] David Hobby and Ralph McKenzie. 1988. *The structure of finite algebras*. Contemporary Mathematics, Vol. 76. American Mathematical Society, Providence, RI. xii+203 pages. Revised edition: 1996.

[21] Katarzyna Idziak and Paweł M. Idziak. 1988. Decidability problem for finite Heyting algebras. *Journal of Symbolic Logic* 53, 3 (1988), 729–735.

[22] P. Idziak, P. Markovic, R. McKenzie, M. Valeriote, and R. Willard. 2010. Tractability and learnability arising from algebras with few subpowers. *SIAM J. Comput.* 39, 7 (2010), 3023–3037.

[23] Peter Jeavons, David Cohen, and Martin C Cooper. 1998. Constraints, consistency and closure. *Artificial Intelligence* 101, 1-2 (1998), 251–265.

[24] Keith A. Kearnes and Emil W. Kiss. 2013. The shape of congruence lattices. *Mem. Amer. Math. Soc.* 222, 1046 (2013), viii+169. https://doi.org/10.1090/S0065-9266-2012-00667-8

[25] Subhash Khot. 2002. On the power of unique 2-prover 1-round games. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*. 767–775.

[26] Ilan Newman. 2010. Property Testing of Massively Parametrized Problems - A Survey. In *Property Testing*. Vol. 6390. Chapter 8, 142–157.
[27] Prasad Raghavendra. 2008. Optimal algorithms and inapproximability results for every CSP?. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*. 245–254.
[28] Dana Ron. 2010. Algorithmic and analysis techniques in property testing. *Foundations and Trends® in Theoretical Computer Science* 5 (2010), 73–205.
[29] Ronitt Rubinfeld and Asaf Shapira. 2011. Sublinear Time Algorithms. *SIAM Journal on Discrete Mathematics* 25, 4 (2011), 1562–1588.
[30] Á Szendrei. 1987. Idempotent algebras with restrictions on subalgebras. *Acta scientiarum mathematicarum* 51 (1987), 251–268.
[31] Johan Thapper and Stanislav Zivny. 2012. The Power of Linear Programming for Valued CSPs. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 669–678.
[32] Johan Thapper and Stanislav Zivny. 2013. The complexity of finite-valued CSPs. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing Conference (STOC)*. 695–704.
[33] Johan Thapper and Stanislav Zivny. 2015. Sherali-Adams Relaxations for Valued CSPs. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming (ICALP)*. 1058–1069.
[34] Yuichi Yoshida. 2014. Testing list H-homomorphisms. *Computational complexity* 25, 4 (2014), 1–37.
[35] D. Zhuk. 2017. The proof of the CSP dichotomy conjecture. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. to appear.