

Testable and Untestable Classes of First-Order Formulae

Charles Jordan^{a,1,*}, Thomas Zeugmann^{a,2}

^a*Division of Computer Science, Hokkaido University, N-14, W-9, Sapporo 060-0814, Japan*

Abstract

In property testing, the goal is to distinguish structures that have some desired property from those that are *far* from having the property, based on only a small, random sample of the structure. We focus on the classification of first-order sentences according to their testability. This classification was initiated by Alon *et al.* [2], who showed that graph properties expressible with prefix $\exists^*\forall^*$ are testable but that there is an untestable graph property expressible with quantifier prefix $\forall^*\exists^*$. The main results of the present paper are as follows. We prove that all (relational) properties expressible with quantifier prefix $\exists^*\forall\exists^*$ (Ackermann's class with equality) are testable and also extend the positive result of Alon *et al.* [2] to relational structures using a recent result by Austin and Tao [8]. Finally, we simplify the untestable property of Alon *et al.* [2] and show that prefixes $\forall^3\exists$, $\forall^2\exists\forall$, $\forall\exists\forall^2$ and $\forall\exists\forall\exists$ can express untestable graph properties when equality is allowed.

Keywords: Property testing, logic, randomized algorithms, Ackermann's class, Ramsey's class

1. Introduction

In property testing, we take a small random sample of a large structure and wish to determine if the structure has some desired property or if it is far from having the desired property. The hope is that we can gain efficiency in return for not deciding the problem exactly. We focus on the classification problem for testability, where the goal is to determine exactly which prefix-vocabulary fragments of first-order logic are testable and which are not. This problem was first considered by Alon *et al.* [2], who focused on quantifier alternations (as opposed to quantifiers) and undirected, loop-free graphs.

More concretely, we would like to know the minimum number of universal quantifiers, as well as of existential quantifiers, required to express an untestable property. In addition, we would like to know the minimum total number of quantifiers needed to express an untestable property, as it is not *prima facie* necessary that one can achieve these two

*Corresponding author

Email addresses: skip@ist.hokudai.ac.jp (Charles Jordan), thomas@ist.hokudai.ac.jp (Thomas Zeugmann)

¹Supported by a Grant-in-Aid for JSPS Fellows under Grant No. 2100195209.

²Supported by MEXT Grant-in-Aid for Scientific Research on Priority Areas under Grant No. 21013001.

minima simultaneously. Previous work by Alon *et al.* [2] implies upper bounds of twelve universal, five existential and seventeen total quantifiers, and it is natural to ask if these bounds can be improved. The following is an informal summary of our results addressing this question.

The minimum numbers of quantifiers sufficient to express an untestable property in a first-order relational language are

1. *Two universal quantifiers;*
2. *One existential quantifier;*
3. *Three³ quantifiers in total.*

In addition, the first two minima can be achieved in the vocabulary of directed graphs (i.e., one binary relation).

If we consider the pattern of quantifiers instead of focusing exclusively on counting them, our results show that one can express an untestable (graph) property with prefixes $\forall^3\exists$, $\forall^2\exists\forall$, $\forall\exists\forall^2$, and $\forall\exists\forall\exists$. It is trivial to show the testability of formulae with only existential quantifiers, but it is natural to consider the remaining prefixes of length four.

In Subsection 4.1, we show that all formulae containing at most one universal quantifier are testable, even without a restriction to graph properties or four quantifiers. This class is well-known as Ackermann’s class with equality, and covers four of the remaining prefixes of length four. In Subsection 4.2, we extend the result of Alon *et al.* [2] from graphs to relational structures and show that all formulae of the form $\exists^*\forall^*$ are testable. This covers three additional prefixes of length four.

We have recently shown that one can express untestable graph properties with prefix $\forall\exists\forall$ (improving on prefixes $\forall^2\exists\forall$, $\forall\exists\forall^2$, and $\forall\exists\forall\exists$ from this paper), see Jordan and Zeugmann [27]. This prefix was first studied by Kahr, Moore and Wang [28].

The testability of the remaining prefixes, i.e., $\exists\forall^2\exists$ and $\forall^2\exists^2$, remains open. These prefixes are variations of the Gödel class (prefixes containing at least $\forall^2\exists$).

The paper is structured as follows. In Subsection 1.1, we outline the history of testing, focusing on results that influence our approach. We state the main results of this paper in Subsection 1.3. Definitions and notation are in Section 2. In Section 3, we prove several basic results that are needed in later sections. The main results are in Section 4 (testable classes) and Section 5 (untestable classes).

1.1. History of Testing

We begin with a brief history and overview of property testing. There is a recent introduction to graph property testing by Goldreich [21], and two recent surveys by Ron, one focusing on connections with learning theory [43] and one focusing on the algorithmic techniques [44] used in testability. There are also earlier surveys, including those by Fischer [15] and Ron [42].

Property testing is a form of approximation where we trade accuracy for efficiency. Probabilistic machines seem to have been first formalized by de Leeuw *et al.* [32], who showed that such machines cannot compute uncomputable properties under reasonable assumptions. However, they mention the possibility that probabilistic machines could

³In the present paper, we only prove that the minimum is either three or four. During the reviewing process, we found an untestable property with prefix $\forall\exists\forall$, see Jordan and Zeugmann [27].

be more *efficient* than deterministic machines, a topic which was then investigated by Gill [19]. An early example of such a result is Freivalds' [18] matrix multiplication checker.

The study of property testing itself began in program verification (see Blum *et al.* [10] as well as Rubinfeld and Sudan [45]). Goldreich *et al.* [22] first considered the testability of graph properties and showed the existence of testable NP-complete properties. An approach using incidence lists to represent bounded-degree graphs was introduced by Goldreich and Ron [20]. Parnas and Ron [37] generalized this approach and attempted to move away from the functional representation of structures. There has been a great deal of recent work on graph property testing, see the survey by Alon and Shapira [5].

For other types of structures, Alon *et al.* [4] showed that the regular languages are testable and that there exist untestable context-free languages. Chockler and Kupferman [13] extended the positive result to the ω -regular languages.

There is also recent work on testing properties of (usually uniform) hypergraphs. Fischer *et al.* [17] defined a general model that is roughly equivalent to one of our models, namely \mathcal{T}_r based on Definition 10 below, and showed that hypergraph partition problems are testable in this framework. Very recently, Austin and Tao [8] have shown that all hereditary properties⁴ of colored, directed hypergraphs are testable in a model that is roughly equivalent to another of our models, \mathcal{T}_{mr} based on Definition 13 below.

Szemerédi's regularity lemma (see, e.g., the survey by Rödl and Schacht [41]) has been extremely influential in (dense) graph property testing and there has been a great deal of work on recent extensions (see, e.g., [23, 40, 49]) of this lemma to hypergraphs. However, as Alon *et al.* [2] noted, proofs of testability that avoid the regularity lemma often result in better query complexity. We therefore prove testability directly when we know how to.

Alon *et al.* [2] began a *logical* characterization of the testable (graph) properties, see Subsection 1.2. Alon and Shapira [6] gave a characterization of a natural subclass of the graph properties testable with one-sided error, which Rödl and Schacht [39] generalized to hypergraphs. Alon *et al.* [3] showed a combinatorial characterization of the graph properties testable with a constant number of queries. It would be particularly interesting to consider extensions of this last result to hypergraphs or relational structures.

1.2. Previous Work on the Classification

We briefly outline prior work on the classification for testability before stating our main results. We begin with monadic first-order logic. Löwenheim [34] proved that satisfiability is decidable⁵ for monadic first-order logic, and McNaughton and Papert [35] showed that it (with ordering and some arithmetic) characterizes the star-free regular languages. The testability of this class is then implied by a result of Alon *et al.* [4]. Using instead Büchi's [12] result that monadic second-order logic characterizes the regular languages, we get a parallel with Skolem's [47] extension of Löwenheim's result to second-order logic. Of course, we are focused on the testability of classes of first-order formulae.

Below, we use the classification notation that will be introduced formally in Definition 8. Informally, we represent classes with a triple $[\Pi, p]_e$, where Π denotes the pattern

⁴A hereditary property is one that is closed under taking induced substructures.

⁵A class is said to be *decidable* (for satisfiability) if, given an arbitrary formula from the class, one can decide if there exists a (possibly infinite) model satisfying the formula.

of quantifiers allowed, the infinite sequence p denotes the maximum number of permitted predicate symbols for each arity (we omit trailing zeros and *all* means that any number of predicate symbols with any arities are permitted), and e denotes whether $=$ is allowed.

Skolem [48] also showed that $[\forall^* \exists^*, all]$ is a reduction class⁶. An untestable graph property (essentially an encoding of graph isomorphism) was found by Alon *et al.* [2]. This property is expressible in $[\forall^* \exists^*, (0, 1)]_=$, and an examination of the proof reveals that a prefix of $\forall^{12} \exists^5$ suffices.

The class $[\exists^* \forall^*, all]_=$ was first studied in a seminal paper by Ramsey [38], who showed that it is decidable as part of a stronger result characterizing its spectrum. Alon *et al.* [2] showed that the restriction of Ramsey's class to undirected loop-free graphs (a restriction of $[\exists^* \forall^*, (0, 1)]_=$) is testable.

In Jordan and Zeugmann [27], we improved on some of the prefixes of this paper and showed that $[\forall \exists \forall, (0, 1)]_=$ is untestable.

1.3. Statement of Results

Our main results are as follows. First, we develop a framework for relational property testing including variations corresponding to the different models considered in the literature for non-uniform hypergraph⁷ testing. We use this framework to prove the following.

1. All (relational) properties expressible by formulae in Ackermann's class with equality ($[\exists^* \forall \exists^*, all]_=$) are *testable* in all of our models.
2. All (relational) properties expressible in Ramsey's class ($[\exists^* \forall^*, all]_=$) are *testable* in all of our models. This extends the result by Alon *et al.* [2] to the full class.
3. There exist graph properties expressible in the classes $[\forall^3 \exists, (0, 1)]_=$, $[\forall^2 \exists \forall, (0, 1)]_=$, $[\forall \exists \forall^2, (0, 1)]_=$ and $[\forall \exists \forall \exists, (0, 1)]_=$ which are *untestable* in all of our models. This sharpens the untestable class of Alon *et al.* [2].

The latter two results improve the results of Alon *et al.* [2], and the second result relies on an application of a strong result by Austin and Tao [8]. In the notation introduced as Definition 8 below, the current classification for testability is as follows.

- Testable classes
 1. Monadic first-order logic: $[all, (\omega)]_=$.
 2. Ackermann's class with equality: $[\exists^* \forall \exists^*, all]_=$.
 3. Ramsey's class: $[\exists^* \forall^*, all]_=$.
- Untestable classes
 1. $[\forall^3 \exists, (0, 1)]_=$.
 2. $[\forall \exists \forall, (0, 1)]_=$.

⁶A class is a *reduction class* if the satisfiability problem for first-order logic can be reduced to the satisfiability problem for the class. These classes are therefore undecidable (for satisfiability).

⁷A hypergraph is uniform if all edges have the same arity, and non-uniform if edges may have different arities.

2. Preliminaries

Instead of restricting our attention exclusively to graphs, we focus on property testing in a more general setting. We begin by defining vocabularies.

Definition 1. A *vocabulary* τ is a tuple of distinct predicate symbols R_i together with their arities a_i ,

$$\tau := (R_1^{a_1}, \dots, R_s^{a_s}).$$

Two examples (unique up to renaming) of vocabularies are $\tau_G := (E^2)$, the vocabulary of directed *graphs* and $\tau_S := (S^1)$, the vocabulary of *binary strings*. We generally use vocabulary τ (i.e., s predicate symbols R_i with arities a_i) in proofs that do not depend on the vocabulary.

Definition 2. A *structure* A of type τ is an $(s+1)$ -tuple

$$A := (U, \mathcal{R}_1^A, \dots, \mathcal{R}_s^A),$$

where U is a finite universe and each $\mathcal{R}_i^A \subseteq U^{a_i}$ is a predicate corresponding to the predicate symbol R_i of τ .

We identify U with the non-negative integers $\{0, \dots, n-1\}$ and use $n = \#(A)$ for the size of the universe of a structure A . We often omit the explicit mention of A and use n to refer to the size of structures whose identity is clear from context. The universe U of a binary string is the set of bit positions, which we will identify as $\{0, \dots, n-1\}$ from left to right. For $i \in U$, we interpret $i \in S$ as “bit i of the string is 1.” We generally omit the superscript A from the relations and include it only when we wish to explicitly distinguish the same relation in different structures.

The set of all structures of type τ and universe size n is $STRUC^n(\tau)$ and the set of all (finite) structures of type τ is $STRUC(\tau) := \bigcup_{n \geq 0} STRUC^n(\tau)$. A *property* P of structures with vocabulary τ is any subset of $STRUC(\tau)$. For a structure A with vocabulary τ , we say A *has* P if $A \in P$.

We use *language* to refer to string properties, P to denote properties and $B \setminus C$ for set difference. We refer to members of $STRUC(\tau_G)$ as *graphs*, and note that our graphs are directed and may contain loops.

A simple example of a graph property is the property of being a complete graph. This property is the set of all graphs which have full edge relations, i.e., $P_K := \bigcup_{n \geq 0} \{(U_n, \mathcal{E}^G) \mid U_n = \{0, \dots, n-1\}, \mathcal{E}^G = U_n \times U_n\}$.

2.1. Property Testing Definitions

We wish to distinguish, with high probability, between inputs that have a desired property and inputs that are *far* from having the property. We begin by defining a *distance measure* between structures. Changing the definition of distance results in a different model for relational testing. The symbol \oplus denotes exclusive-or.

Definition 3. Let $A, B \in STRUC(\tau)$ be any structures such that $\#(A) = \#(B) = n$. The *distance* between structures A and B is

$$\text{dist}(A, B) := \frac{\sum_{i=1}^s |\{\mathbf{x} \mid \mathbf{x} \in U^{a_i} \text{ and } \mathcal{R}_i^A(\mathbf{x}) \oplus \mathcal{R}_i^B(\mathbf{x})\}|}{\sum_{i=1}^s n^{a_i}}.$$

The dist distance is the fraction of assignments on which the two structures disagree. It is equivalent to the definition that would result from mapping relational structures to binary strings and using the usual definitions for testing strings. We note that Definition 3 is common in the literature on graph property testing, but that it is generally not used in testing properties of non-uniform hypergraphs (for reasons discussed before Definition 10 below). However, as we will see in Theorem 5, it results in the *weakest* notion of testability that we consider and so we will use it when proving *untestability* results.

We now give the remaining definitions for testing, and will then give alternatives to Definition 3 (cf. Definitions 10 and 13 below). When proving results, we always state explicitly which models the results hold in.

Definition 4. Let P be a property of structures with vocabulary τ and let A be such a structure with a universe of size n . Then,

$$\text{dist}(A, P) := \min_{A' \in P \cap \text{STRUC}^n(\tau)} \text{dist}(A, A') .$$

Definition 5. An ε -tester for property P is a randomized algorithm given an oracle which answers queries for the universe size and truth values of relations on desired tuples in a structure A . The tester must accept A with probability at least $2/3$ if A has P and must reject A with probability at least $2/3$ if $\text{dist}(A, P) \geq \varepsilon$.

Testers are called *oblivious* (see Alon and Shapira [6]) if they are not allowed to make decisions based on the size of the universe. More concretely, a tester in their setting is only allowed to give the oracle a natural Q , and the oracle then uniformly randomly selects Q elements of the universe of A and returns the resulting induced substructure. However, if A is of size smaller than Q , then the entire structure is returned. This is more restricted than our model, but our positive results hold even in the oblivious setting.

Some of our results hold even when the testers are restricted to one-sided error, where the following definition applies.

Definition 6. An ε -tester for P has *one-sided error* if it accepts with probability 1 if A has P and rejects with probability at least $2/3$ if $\text{dist}(A, P) \geq \varepsilon$.

Definition 7. Property P is *testable* if for every $\varepsilon > 0$ there is an ε -tester making a number of queries which is upper-bounded by a function depending only on ε .

We say that a property P is testable *with one-sided error* if the ε -testers satisfy the additional restriction of having one-sided error. Note that we allow the ε -testers to be different for each $\varepsilon > 0$. This results in uniform (where the ε -testers are computable given ε , or equivalently, where there is a single tester that also takes ε as input) and non-uniform versions of testability. Although testable properties in the literature are usually uniformly testable, see Alon and Shapira [7] for a property that is testable only with uncomputable $c(\varepsilon)$ and therefore only non-uniformly. Our results hold in both cases⁸ and so we will not distinguish between them.

⁸That is, our negative results hold for non-uniform testing and positive results for uniform testing. In the uniform case, we must restrict Lemma 10 in Section 3 to decidable properties. All properties considered in the present paper are clearly decidable.

2.2. Logical Definitions

We use a predicate logic with equality that does not contain function symbols. There are no ordering symbols such as \leq or arithmetic relations such as *PLUS*. The first-order logic of vocabulary τ is built from the atomic formulae $x_i = x_j$ and $R_i(x_1, \dots, x_{a_i})$ for variable symbols x_j and predicate symbols $R_i \in \tau$ by using the Boolean connectives and quantifiers \exists and \forall in the usual way.

Formula φ of vocabulary τ is interpreted as usual and defines property $P := \{A \mid A \in STRUC(\tau) \text{ and } A \models \varphi\}$. Lower-case Greek letters φ , ψ and γ refer to first-order formulae and x , y , and z to first-order variables. Our classification definitions are from Börger *et al.* [11] except that we omit function symbols. Essentially, we classify first-order sentences according to their pattern of quantifiers and vocabulary. The following is for completeness, where $\mathbb{N} = \{0, 1, \dots\}$ denotes the set of natural numbers.

Definition 8. A *prefix vocabulary class* is specified as $[\Pi, p]_e$, where Π is a string over the four-character alphabet $\{\exists, \forall, \exists^*, \forall^*\}$, p is a sequence over \mathbb{N} and the first infinite ordinal ω , and e is '=' or the empty string.

We often use *all* as an abbreviation for the sequence $(\omega, \omega, \omega, \dots)$. Now that we have defined the syntactic specification of a prefix vocabulary class, we define the class specified by a triple $[\Pi, p]_e$. Recall that a first-order sentence φ is in prenex normal form if it is in the form $\varphi := \pi_1 x_1 \pi_2 x_2 \dots \pi_r x_r : \psi$, with quantifiers π_i , $1 \leq i \leq r$, and quantifier-free ψ . Such a φ is a member of the prefix vocabulary class given by $[\Pi, (p_1, p_2, \dots)]_e$, where $p_i \in \mathbb{N} \cup \{\omega\}$ if

1. The string $\pi_1 \pi_2 \dots \pi_r$ is contained in the language specified by Π when Π is interpreted as a regular expression.
2. If p is not *all*, at most p_i distinct predicate symbols of arity i appear in ψ .
3. Equality ($=$) appears in ψ only if e is '='.

Here, Π is the pattern of quantifiers, p is the maximum number of predicate symbols of each arity and e determines whether or not the equality symbol is permitted.

We use the following conventions to avoid unwieldy language.

- Definition 9.**
1. A sentence is *(un)testable* if the property it defines is (un)testable.
 2. A prefix class is *testable* if every sentence in it expresses a testable property for every vocabulary in which it is evaluable.
 3. A prefix class is *untestable* if it contains an untestable sentence.

An extension of a vocabulary τ is any vocabulary formed by adding a new, distinct predicate symbol to τ . The following simple lemma justifies the intuition that we can focus on the minimal vocabulary needed in a formula and ignore vocabularies that include extraneous predicate symbols.

Lemma 1. *Let φ be a formula in the first-order logic of vocabulary τ and let τ' be any extension of τ . If φ defines a property that is testable in the context of τ , then the property of type τ' defined by φ is also testable.*

PROOF. Let φ define property P of type τ and property P' of type τ' . Assume the “new” predicate symbol in τ' is N of arity a . Let T_ε^τ be an ε -tester for P . We will show that it is also an ε -tester for P' . Assume $A' \in STRUC(\tau')$ has property P' . Removing the N predicate, the corresponding $A \in STRUC(\tau)$ has property P and so T_ε^τ accepts with probability at least $2/3$, as desired.

Assume that $\text{dist}(A', P') \geq \varepsilon$ and again let A be the structure of type τ formed by removing the N predicate from A' . By the definition of distance,

$$\begin{aligned} \text{dist}(A, P) &= \min_{B \in P} \frac{\sum_{i=1}^s |\{\mathbf{x} \mid \mathbf{x} \in U^{a_i} \text{ and } \mathcal{R}_i^A(\mathbf{x}) \oplus \mathcal{R}_i^B(\mathbf{x})\}|}{\sum_{i=1}^s n^{a_i}} \geq \\ \min_{B \in P} \frac{\sum_{i=1}^s |\{\mathbf{x} \mid \mathbf{x} \in U^{a_i} \text{ and } \mathcal{R}_i^{A'}(\mathbf{x}) \oplus \mathcal{R}_i^B(\mathbf{x})\}|}{n^a + \sum_{i=1}^s n^{a_i}} &= \text{dist}(A', P') \geq \varepsilon. \end{aligned}$$

The tester rejects such an A' with probability at least $2/3$, as desired. \square

Testable properties remain testable when the vocabulary is extended. So it suffices to consider the minimal relevant vocabulary. Simple modifications of the proof of Lemma 1 give the corresponding results for the variations considered in the next subsection.

2.3. Variations of Relational Property Testing

We now consider alternatives to Definition 3, which was presented first due to its simplicity. In Definition 3, any difference in low-arity relations is asymptotically dominated by the number of high-arity tuples. However, there are situations where this is not ideal. Consider (not necessarily admissible⁹, vertex) 3-colored graphs with the vocabulary $\tau_C := (E^2, R^1, G^1, B^1)$, where we use the binary predicate E to represent edges and the monadic predicates to represent colors. We might wish to test if the given coloring is admissible. In large graphs, this is equivalent to testing if the graph is 3-colorable and ignores the given coloring. We need a different model for our task.

Here we give two alternative definitions for the distance between structures. In testing we wish to distinguish structures that have a desired property and those that are far from the property, and so modifying the definition of distance changes the task of testing. As in Definition 3, the symbol \oplus denotes exclusive-or.

Definition 10. Let $A, B \in STRUC^n(\tau)$ be structures. Then, the r -distance is

$$\text{rdist}(A, B) := \max_{1 \leq i \leq s} \frac{|\{\mathbf{x} \mid \mathbf{x} \in U^{a_i} \text{ and } \mathcal{R}_i^A(\mathbf{x}) \oplus \mathcal{R}_i^B(\mathbf{x})\}|}{n^{a_i}}.$$

While Definition 3 gave equal weight to each *tuple* regardless of its arity, the above gives equal weight to each *relation*. Definition 10 is essentially equivalent to the model used by Fischer *et al.* [17].

However, loops (i.e., self-edges (x, x)) in graphs and other *subrelations* of relations are similar to low-arity relations. In Definition 10, these are still dominated by the “non-degenerate” tuples. Definition 13 will resolve this issue and result in a model of testability essentially equivalent to that implicit in Austin and Tao [8]. We begin by defining the syntactic notion of *subtype* before proceeding to subrelations.

⁹An admissible vertex-coloring is one that assigns distinct colors to adjacent vertices.

Definition 11. A *subtype* S of a predicate symbol with arity a is any partition of the set $\{1, \dots, a\}$.

For example, graphs have a single, binary predicate symbol E^2 which has two subtypes: $\{\{1, 2\}\}$ and $\{\{1\}, \{2\}\}$, corresponding to loops and non-loops respectively. Let $SUB(R)$ denote the set of subtypes of predicate symbol R .

Definition 12. Let $A \in STRUC(\tau)$ be a structure with vocabulary τ and universe U , and let S be a subtype of predicate symbol $R^a \in \tau$. We define the following.

- $s^U(S)$, the *tuples that belong to S* , is the set of $(x_1, \dots, x_a) \in U^a$ satisfying the following condition. For every $1 \leq j, k \leq a$, $x_j = x_k$ iff j and k are contained in the same element of S .
- The *subrelation* $s^A(S)$ of A corresponding to S is $s^A(S) := s^U(S) \cap \mathcal{R}^A$.

Returning to our example of graphs, the sets of loops and non-loops are the subrelations of the edge relation \mathcal{E} corresponding to the subtypes $\{\{1, 2\}\}$ and $\{\{1\}, \{2\}\}$ of E^2 , respectively.

We denote the symmetric difference of sets U and V by $U \Delta V$, i.e.,

$$U \Delta V := (U \setminus V) \cup (V \setminus U).$$

Definition 13. Let $A, B \in STRUC^n(\tau)$ be structures with vocabulary τ and universe size n . The *mr-distance* between A and B is

$$\text{mrdist}(A, B) := \max_{R \in \tau} \max_{S \in SUB(R)} \frac{|s^A(S) \Delta s^B(S)|}{n! / (n - |S|)!}.$$

The distance between structures is the fraction of assignments that differ in the most different subtype. As before, the distance between a structure and a property is the distance to the closest structure with the property. We let \mathcal{T} be the set of testable properties using the dist definition, \mathcal{T}_r be the set of testable properties using the rdist definition and \mathcal{T}_{mr} be the set of testable properties using the mrdist definition.

3. Basic Results

We begin by relating the three models for testability that we defined in Section 2. Theorem 5 below is quite simple, however, it justifies our focus on \mathcal{T} for untestable classes and on \mathcal{T}_{mr} for testable classes. First, the distances are related in the following simple way.

Lemma 2. *Let τ be a vocabulary and $A, B \in STRUC^n(\tau)$. Then,*

$$\text{dist}(A, B) \leq \text{rdist}(A, B) \leq \text{mrdist}(A, B).$$

PROOF. We first show $\text{dist}(A, B) \leq \text{rdist}(A, B)$. If an ε -fraction of all assignments differs and we partition the assignments, there must be a partition such that at least an

ε -fraction of the assignments differs in the partition. Let $\text{dist}(A, B) = \varepsilon$ and let α_i be the fraction of R_i -assignments that differ between the structures,

$$\alpha_i := \frac{|\{\mathbf{x} \mid \mathbf{x} \in U^{a_i} \text{ and } R_i^A(\mathbf{x}) \oplus R_i^B(\mathbf{x})\}|}{n^{a_i}}.$$

Then, $\text{rdist}(A, B) = \max_i \alpha_i$ and we can write $\text{dist}(A, B)$ in terms of the α_i ,

$$\text{dist}(A, B) = \frac{\sum_i \alpha_i n^{a_i}}{\sum_i n^{a_i}} = \varepsilon.$$

This implies that $\sum_i \alpha_i n^{a_i} = \varepsilon \sum_i n^{a_i}$, and so there must be an $\alpha_i \geq \varepsilon$.

Next, we show that $\text{rdist}(A, B) \leq \text{mrdist}(A, B)$. The proof is nearly identical to the above. If $\text{rdist}(A, B) = \varepsilon$ then there is an R_i such that an ε -fraction of the R_i -assignments differs between the structures. If we partition the R_i -assignments into the subtypes of R_i (which are disjoint), then there must be some partition such that at least an ε -fraction of the assignments in that partition differ. \square

Assume a tester distinguishes between structures A having some property P and those for which $\text{mrdist}(A, P) \geq \varepsilon$. Lemma 2 trivially implies that it also distinguishes between structures A that have P and those for which $\text{rdist}(A, P) \geq \varepsilon$. The case with rdist and dist is analogous, which proves the following.

Corollary 3. $\mathcal{T}_{mr} \subseteq \mathcal{T}_r \subseteq \mathcal{T}$.

Of course it is always desirable to show that such containments are strict. We show the separations by encoding the following language of binary strings, where \hat{u} denotes the usual reversal of string u . It is also possible to use, e.g., the untestable property that will be seen in Section 5 to prove the separations with a first-order expressible property that is closed under isomorphisms.

Theorem 4 (Alon et al. [4]). *The language $L = \{u\hat{u}v\hat{v} \mid u, v \text{ are strings over } \{0, 1\}\}$ is not testable.*

In some vocabularies, e.g., binary strings, all three definitions are equivalent. However, we will show the following.

Theorem 5. $\mathcal{T}_{mr} \subset \mathcal{T}_r \subset \mathcal{T}$.

PROOF. The inclusions are by Corollary 3 and so only the separations remain. We first show that $\mathcal{T} \setminus \mathcal{T}_r$ is not empty. It suffices to give a vocabulary τ and a property with vocabulary τ that is \mathcal{T} -testable but not \mathcal{T}_r -testable. We use the vocabulary $\tau_C := (E^2, S^1)$.

We will show $P_1 \in \mathcal{T} \setminus \mathcal{T}_r$, where $P_1 \subseteq \text{STRUC}(\tau_C)$ is the set of structures where the S assignments encode the language L of Theorem 4 above. Recall that n denotes the size of the universe and our convention is that $S(i)$ is interpreted as “bit i of the string is 1”. Therefore, A has P_1 if there is some $0 \leq k \leq n/2$ such that for all $0 \leq i < k$, $S(i)$ is true iff $S(2k - 1 - i)$ is true and for all $0 \leq j < (n - 2k)/2$, $S(2k + j)$ is true iff $S(n - 1 - j)$ is true. The property uses only the low-arity relation S ; the E relation is for “padding” to make P_1 testable under the dist definition for distance.

We first show that P_1 is in \mathcal{T} . A structure with a universe of odd size cannot have P_1 . A tester can begin by checking the parity of n and rejecting if it is odd and so we assume in the following that the size of the universe is even.

Lemma 6. *Property P_1 is testable under the dist definition for distance.*

PROOF. For any (even) n , 1^n is of the form $u\bar{v}v\bar{v}$. Changing all $S(i)$ assignments to true in any given A results in the string 1^n . This involves at most n modifications and so $\text{dist}(A, P_1) \leq \text{dist}(A, A') = O(n)/\Theta(n^2) < \varepsilon$, where the final inequality holds for sufficiently large n . Let $N(\varepsilon)$ be the smallest value of n for which it holds. The following is an ε -tester for P_1 , where the input has universe size n .

1. If $n < N(\varepsilon)$, query all assignments and output whether the input has P_1 .
2. Otherwise, accept.

If A has P_1 , we accept with zero error. If $\text{dist}(A, P_1) \geq \varepsilon$, then $n < N(\varepsilon)$. In this case we query all assignments and reject with zero error. \square Lemma 6

It remains to show that P_1 is not testable when using the rdist definition for distance. We do this by showing that it would contradict Theorem 4 above.

Lemma 7. *Property P_1 is not testable under the rdist definition for distance.*

PROOF. Suppose there exist \mathcal{T}_r -type ε -testers T^ε for all $\varepsilon > 0$. The following is a \mathcal{T} -type ε -tester for the language L of Theorem 4. Let the input be w , a binary string of length n .

1. Run T^ε and intercept all queries.
2. When a query is made for $S(i)$, return the value of $S(i)$ in w .
3. When a query is made for $E(i, j)$, return 0.
4. Output the decision of T^ε .

We run T^ε on the $A \in \text{STRUC}^n(\tau_C)$ that agrees with w on S and where all E assignments are false. If $w \in L$, then any such A has property P_1 and so our tester accepts with probability at least $2/3$.

Assume $\text{dist}(w, L) \geq \varepsilon$. Then, $\text{rdist}(A, P_1) = \text{dist}(w, L) \geq \varepsilon$ and so our tester rejects with probability at least $2/3$. These are testers for the untestable language of Theorem 4, and so P_1 is untestable under the rdist definition. \square Lemma 7

Lemmata 6 and 7, together with Corollary 3 show $\mathcal{T}_r \subset \mathcal{T}$. The separation $\mathcal{T}_{mr} \subset \mathcal{T}_r$ is shown in a similar way, using a property with sufficient “padding” to make \mathcal{T}_r testing simple but \mathcal{T}_{mr} testing would contradict Theorem 4.

For example, one can use the property P_2 of graphs in which the “loops” $E(i, i)$ encode the language from Theorem 4. That is, a graph has P_2 if there is some $0 \leq k \leq n/2$ such that for all $0 \leq i < k$, $E(i, i)$ is true iff $E(2k - 1 - i, 2k - 1 - i)$ is true and for all $0 \leq j < (n - 2k)/2$, $E(2k + j, 2k + j)$ is true iff $E(n - 1 - j, n - 1 - j)$ is true. The non-loops are used as padding to ensure \mathcal{T}_r testability while \mathcal{T}_{mr} testability would allow us to violate Theorem 4. \square Theorem 5

There exist properties that are testable in the rdist sense but not in the mrdist sense. However, the definition of subtypes and \mathcal{T}_{mr} testability allows for a simple mapping between vocabularies such that rdist-testability of certain *classes* of properties implies mrdist-testability of the same classes. For these classes, proving testability in the rdist sense is equivalent to proving it in the mrdist sense, and so it suffices to use whichever definition is more convenient.

Lemma 8 below is stated in the context of the classification problem for first-order logic but it is not difficult to prove similar results in other contexts. We use only equality and that the class is closed under adding additional predicate symbols. The main result in Subsection 4.1 will be the testability of Ackermann's class with equality, which is of the form required by the lemma.

Lemma 8. *Let $\mathcal{C} := [\Pi, all]_{=}$ be a prefix vocabulary class. Then, \mathcal{C} is testable in the rdist sense iff it is testable in the mrdist sense.*

PROOF. Recalling Theorem 5, \mathcal{T}_{mr} testability implies \mathcal{T}_r testability. We prove \mathcal{T}_r testability of such prefix classes implies \mathcal{T}_{mr} testability using Lemma 9. In the following, $\mathfrak{S}(n, k)$ is the Stirling number of the second kind.

Lemma 9. *Let $\mathcal{C} = [\Pi, (p_1, p_2, \dots)]_{=}$ be a prefix vocabulary class and, furthermore, let $q_j = \sum_{i \geq j} p_i \mathfrak{S}(i, j)$. If $\mathcal{C}' = [\Pi, (q_1, q_2, \dots)]_{=}$ is \mathcal{T}_r testable, then \mathcal{C} is \mathcal{T}_{mr} testable.*

PROOF. Let $\varphi \in \mathcal{C}$ be arbitrarily fixed and assume that the predicate symbols of φ are $\{R_1^1, R_2^1, \dots, R_{p_1}^1, R_1^2, \dots\}$, where the arity of R_j^i is i . We construct a $\varphi' \in \mathcal{C}'$ and show that \mathcal{T}_r testability of φ' implies \mathcal{T}_{mr} testability of φ . In φ' we will use a distinct predicate symbol for each subtype of each R_j^i in φ . A subtype S of R_j^i such that $|S| = k$ is a partition of the integers $\{1, \dots, i\}$ into k non-empty sets and so there are $\mathfrak{S}(i, k)$ such subtypes. We therefore require a total of q_k distinct predicate symbols of arity k .

For example, we will map the “loops” in a binary predicate E to a new monadic predicate and the non-loops to a separate binary predicate. Formally, recall that s^U maps the subtypes of a predicate to the sets of tuples comprising the subtypes. For our example of a binary predicate, $(0, 1) \in s^U(\{\{1\}, \{2\}\})$ and $(0, 0) \in s^U(\{\{1, 2\}\})$. Next, we let r be a bijection from the subtypes of predicates to their new names, the predicate symbols that we will use in φ' .

We create φ' by modifying φ . Replace all occurrences of $R_j^i(x_1, \dots, x_i)$ with

$$\left(\bigvee_{S \in \text{SUB}(R_j^i)} [(x_1, \dots, x_i) \in s^U(S) \wedge r(S, R_j^i)(\mathbf{y})] \right).$$

Note that $(x_1, \dots, x_i) \in s^U(S)$ is an abbreviation for a simple conjunction, e.g., $x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge \dots$. Likewise, \mathbf{y} is an $|S|$ -ary tuple, formed by removing the duplicate components of (x_1, \dots, x_i) . The implicit mapping from (x_1, \dots, x_i) is invertible given S . To continue our example of a binary predicate E , we would replace all occurrences of $E(x, y)$ in φ with

$$([x = y \wedge E_1(x)] \vee [x \neq y \wedge E_2(x, y)]).$$

We assume that φ' is \mathcal{T}_r testable, and so there exists an ε -tester T^ε for it. We run this tester and intercept all queries. For a query to $r(S, R_j^i)(\mathbf{y})$, we return the value of

$R_j^i(x_1, \dots, x_i)$. This is possible because r is a bijection, and so we can retrieve S and R_j^i using its inverse. Then, we can reconstruct the full i -ary tuple (x_1, \dots, x_i) from \mathbf{y} and S .

The tester implicitly defines a map¹⁰ from structures A which we wish to test for φ to structures A' (with the same universe as A) which we can test for φ' . Given an $A \models \varphi$, the corresponding $A' \models \varphi'$ and so T^ε will accept with probability at least $2/3$.

We map each subtype S to a distinct predicate symbol with arity $|S|$. Therefore, for any structures A, B , the implicit mapping to A', B' is such that

$$\text{mrdist}(A, B) = \text{rdist}(A', B').$$

For convenience, let $P := \{B \mid B \models \varphi\}$ and $P' := \{B' \mid B' \models \varphi'\}$. For an A such that $\text{mrdist}(A, P) \geq \varepsilon$, we simulate T^ε on an A' such that $\text{rdist}(A', P') \geq \varepsilon$. The tester T^ε rejects with probability at least $2/3$, as desired. \square Lemma 9

Proving \mathcal{T}_r testability for $[\Pi, \text{all}] =$ implies proving it for all (q_1, \dots) that are “images” of some (p_1, \dots) and so Lemma 9 is stronger than required. \square Lemma 8

The proofs in Subsection 4.2 and Section 5 will follow the proofs by Alon *et al.* [2], and in particular rely on a generalization of their notion of *indistinguishability* to relational structures, which we define as follows.

Definition 14. Let $P_1, P_2 \subseteq \text{STRUC}(\tau)$ be properties with vocabulary τ that are closed under isomorphisms. We say that P_1 and P_2 are *indistinguishable* if for every $\varepsilon > 0$ there exists an $N := N(\varepsilon) \in \mathbb{N}$ such that the following holds for all $n > N$. For every $A \in \text{STRUC}^n(\tau)$, if A has property P_1 , then $\text{mrdist}(A, P_2) < \varepsilon$ and if A has P_2 , then $\text{mrdist}(A, P_1) < \varepsilon$.

Alon *et al.* [2] introduced the concept of indistinguishability and showed that it preserves testability of graph properties. This is true even after the above extension to relational properties.

Lemma 10. Let $P_1, P_2 \subseteq \text{STRUC}(\tau)$ be *indistinguishable* properties with vocabulary τ . Property P_1 is testable iff P_2 is testable.

The proof by Alon *et al.* [2] extends without difficulty.

4. Testable Classes

Next, we show that two well-known classes of first-order logic are *testable*. We begin with Ackermann’s class with equality, the set of first-order sentences that have at most one universal quantifier (and any number of existential quantifiers), before proceeding to Ramsey’s class in Subsection 4.2.

¹⁰Explicitly, map A to an A' with the same universe size, where $\mathbf{y} \in r(S, R_j^i)$ in A' if $(x_1, \dots, x_i) \in R_j^i$ in A . Note that we have not yet defined the assignments of tuples \mathbf{y} with duplicate components. By construction, the assignments of these tuples do not affect φ' and so any reasonable convention will do. For example, for any predicate symbol Q of φ' and any tuple \mathbf{z} that has at least one duplicate component, we define $\mathbf{z} \notin Q$. The resulting map is injective but not necessarily surjective.

4.1. Ackermann's Class with Equality

In this section we show that Ackermann's class with equality ($[\exists^*\forall\exists^*, all]_=$) is testable. We begin by reviewing the history of this class, which has a number of nice properties.

Ackermann's class was first considered (without equality) by Ackermann [1], who showed that the satisfiability problem for the class is decidable and that it has the finite model property¹¹. Kolaitis and Vardi [30] showed the satisfiability problem for Ackermann's class with equality is complete for NEXPTIME and that a 0-1 law holds for existential second-order logic¹² where the first-order part belongs to $[\exists^*\forall\exists^*, all]_=$. Lewis [33] proved that satisfiability for Ackermann's class *without* equality is complete for (deterministic) EXPTIME. Grädel [24] showed that satisfiability for Ackermann's class without equality is complete for EXPTIME even with the addition of arbitrarily-many function symbols.

If we allow equality and a unary function symbol, the result is Shelah's class, which Shelah [46] proved decidable. Shelah's class is a decidable class that does not have the finite model property, and it would be interesting to determine if it is testable. This would require extending relational testing to allow function symbols.

Ackermann's class with equality has been studied in other settings as well. For example, Fermüller and Salzer [14] used an extension of resolution to decide an extension of Ackermann's class with equality using automated theorem provers.

The main goal of this subsection is Theorem 11 below. Recalling Theorem 5, this also implies that such properties are testable in the *dist* and *rdist* senses. If the vocabulary consists of a single relation, the *rdist* and *dist* definitions are equivalent to the dense hypergraph model. We therefore obtain the corresponding results in the dense hypergraph and dense graph models as special cases.

We denote the set of monadic predicate symbols in a vocabulary τ by $M := \{R_i \mid R_i \in \tau \text{ and } a_i = 1\}$. The set of assignments of the symbols in M for an element in a universe is called the *color* of the element and there are $2^{|M|}$ possible colors. We define $\text{Col}(A, c)$ to be the set of colors that occur at least c times in A .

Theorem 11. *All formulae in $[\exists^*\forall\exists^*, all]_=$ define properties that are in \mathcal{T}_{mr} with one-sided error.*

PROOF. Recall that Ackermann's class with equality is $[\exists^*\forall\exists^*, all]_=$ and, therefore, it suffices to show the testability of property P of type $\tau = (R_1^{a_1}, \dots, R_s^{a_s})$ defined by formula $\varphi := \exists x_1 \dots \exists x_a \forall y \exists z_1 \dots \exists z_b : \psi$, where ψ is quantifier-free. Note that a is the number of leading existential quantifiers and b is the number of trailing existential quantifiers. We can trivially test any φ that has only finitely-many models with a constant number of queries and zero error, and so it suffices to assume that φ has infinitely-many models.

¹¹A class is said to have the *finite model property* if every satisfiable formula in the class has a finite model. Classes without this property have *infinity axioms*, i.e., sentences with only infinite models.

¹²A class C of first-order logic has an associated 0-1 law if all existential *second-order* sentences $\varphi := \exists C_1 \dots C_a \psi$, where ψ is a first-order sentence in C , have the property that the limit as $n \rightarrow \infty$ of the probability that a random structure of size n satisfies φ exists and is either 0 or 1. Recall that the focus is on existential *second-order* because all of first-order admits a 0-1 law, see the references in Kolaitis and Vardi [31].

The class $[\exists^* \forall \exists^*, all]_{=}$ is of the form required by Lemma 8 above, and so it is *mrdist-testable* iff it is *rdist-testable*. It therefore suffices to show that P is testable in the *rdist* sense. We will show that the following is an ε -tester in the *rdist* sense for P on input $A \in STRUC^n(\tau)$. Here, $k := k(\tau, \varepsilon)$ is the number of elements queried and $N := N(\varphi, \tau, \varepsilon)$ is a constant, both of which are determined below. Note the actual number of *queries* in Step 2 is not exactly k , but rather a constant multiple of it depending on τ . Finally, we explicitly give $\kappa := \kappa(\varphi, \tau)$ below.

1. If $n < N$, query all of A and decide exactly whether A has P .
2. Uniformly and independently choose k members of the universe of A and query all monadic predicates on the members in this sample. Let B be the observed substructure.
3. Search over all $A' \in STRUC^\kappa(\tau)$. Accept if an A' is found such that $A' \models \varphi$ and $\text{Col}(B, a+1) \subseteq \text{Col}(A', a+1)$.
4. Otherwise, reject.

We will show that the tester accepts (with probability 1) if $A \models \varphi$ and rejects with probability $2/3$ if $\text{rdist}(A, P) > \varepsilon$. We first show that if $A \models \varphi$, then the tester is guaranteed to accept. Then, we will show in Lemma 13 that with probability at least $2/3$, we get a “good” sample in Step 2. A sample is “good” if it contains at least $(a+1)$ -many distinct representatives of each color that occurs on at least an $\varepsilon/(2 \cdot 2^{|M|})$ fraction of the elements of A . We then show that the tester is correct if it obtains a good sample, and therefore rejects with probability at least $2/3$ if $\text{rdist}(A, P) > \varepsilon$.

We will now show that if $A \models \varphi$, the tester will accept with probability 1. We begin with Lemma 12.

Lemma 12. *Let A be a model of φ such that $\#(A) > N$ and let*

$$\kappa := a + 3b \left(a + 2^{\sum_{i=1}^s \sum_{j=1}^{a_i} \binom{a_i}{j} a^{a_i-j}} \right) + 2^{|M|}(a+1).$$

Then, there is an $A' \models \varphi$ such that $\#(A') = \kappa$ and $\text{Col}(A, a+1) \subseteq \text{Col}(A', a+1)$.

PROOF. Assume that $N > \kappa$. The structure A is a model of φ , and so there exists at least one tuple of a elements (u_1, \dots, u_a) such that φ is satisfied when the existential quantifiers bind u_i to x_i . We consider the x_i and the substructure induced by them to be fixed, and refer to this substructure as A_x .

There are at most $\kappa_2 := a + 2^{\sum_{i=1}^s \sum_{j=1}^{a_i} \binom{a_i}{j} a^{a_i-j}}$ many *distinct* structures constructed by adding an element labeled y to A_x when we include the structures where the label y is simply placed on one of the x_i . We let $v \leq \kappa_2$ be the number of such structures that occur in A and assume there is an enumeration of them.

For each of these v substructures there exist b elements, w_1, \dots, w_b , such that when we label w_i with z_i , the structure induced by $(x_1, \dots, x_a, y, z_1, \dots, z_b)$ models ψ . We construct $A_{i,j}$ for $1 \leq i \leq v$ and $1 \leq j \leq b$ such that $A_{i,j}$ is a copy of the w_1, \dots, w_b used for the j -th structure. We connect each $A_{i,j}$ to A_x in the same way as in A , modifying assignments on tuples $(A_x \cup A_{i,j})^{a_k}$.

For each w_h in $A_{i,j}$, we consider the case where y is bound to w_h . By construction the substructure induced by (x_1, \dots, x_a, y) occurs in A . We assume it is the g -th structure

and use the elements of $A_{i+1 \bmod 3, g}$ to construct a structure satisfying ψ . We modify the assignments of tuples as needed to create a structure identical to that in A satisfying ψ . Note that by construction all of these assignments are of tuples that contain w_h and at least one element from $A_{i+1 \bmod 3, g}$. The resulting structure, which we call A_1 , is a model of φ . Before this step we have not modified any assignments “spanning” the “rows” $A_{i, j}$ of A_1 and so there are no assignments that we modify more than once.

However, there may be some color from $\text{Col}(A, a+1)$ that does not appear $a+1$ times in A_1 . We therefore add a new block, denoted A_e , of at most $2^{|M|}(a+1)$ elements which consists of $a+1$ copies of each color from $\text{Col}(A, a+1)$. Each of these colors occurred at least $a+1$ times in A , and so for each such color C , there is an element q in A with color C such that q is *not* part of A_x . If the substructure induced by (A_x, q) in A is the j -th structure in our enumeration, then we do the following for each member p of A_e that has the same color as q . First, we make the substructure induced by (A_x, p) identical to that induced by (A_x, q) in A . Next, we make the substructure induced by $(p, A_{1, j})$ identical to that induced by q and the corresponding z_i in A . All of these modifications are on tuples containing a $p \in A_e$ and so we do not modify any tuples more than once. We call this structure A_2 .

Finally, so far we only have an upper-bound on the size of A_2 while the lemma states it to be *exactly* of size κ . We therefore pad in the following simple way¹³. We know that $N > \kappa > 2^{|M|}a$ and so there is a color that occurs at least $a+1$ times in A . If $\#(A_2) < \kappa$, we simply make an additional $\kappa - \#(A_2)$ many copies of this color in A_e and modify the assignments of tuples containing these new elements in the same manner as above. The resulting A' has size κ and satisfies the requirements of the lemma. \square Lemma 12

For any sample B of A , it is true that $\text{Col}(B, a+1) \subseteq \text{Col}(A, a+1)$. If $A \models \varphi$, then Lemma 12 implies that our tester will find an A' satisfying the conditions of Step 3 and will therefore accept. This holds for any sample B and so the tester will accept such A with probability 1.

Next, assume that $\text{rdist}(A, P) \geq \varepsilon$. In this case we must show that the tester rejects with probability at least $2/3$. First, we show that the tester obtains a “good” sample with probability at least $2/3$.

Lemma 13. *There are constants k and N such that, with probability at least $2/3$, the tester obtains a sample that contains at least $(a+1)$ -many distinct representatives of each color in $\text{Col}(A, \varepsilon n / (2 \cdot 2^{|M|}))$.*

PROOF. The probability that any particular query misses a fixed color that occurs on at least an $\varepsilon / (2 \cdot 2^{|M|})$ fraction of A is at most $(1 - \varepsilon / (2 \cdot 2^{|M|}))$. Moreover, the probability that we miss such a fixed color after k_1 independent queries is at most $(1 - \varepsilon / (2 \cdot 2^{|M|}))^{k_1}$. There are at most $2^{|M|}$ such colors, and so the probability that a sample of k_1 elements fails to contain at least one representative of all such colors is at most

$$p_1 := 2^{|M|} \left(1 - \frac{\varepsilon/2}{2^{|M|}} \right)^{k_1}.$$

The $|M|$ is a constant, and we choose k_1 such that $(a+1)p_1$ is at most $1/6$.

¹³One could instead change the tester to search structures with size at most κ .

Let $k := (a + 1)k_1$. We will make k independent queries, and consider the total sample as $(a + 1)$ separate samples of size k_1 . All of these smaller samples will contain at least one representative of every color in $\text{Col}(A, \varepsilon n / (2 \cdot 2^{|M|}))$ with probability at least $1 - (a + 1)p_1 \geq 5/6$. However, there is the possibility that some of these smaller samples contain elements in common. We will choose N such that for $n > N$, the probability that any particular element in the universe of A is chosen more than once is at most $1/6$. In particular, if $\#(A) = n$ and we define $x^y := x(x - 1) \cdots (x - (y - 1)) = x! / (x - y)!$, then the probability that some element is queried more than once is

$$p_2 := 1 - \frac{n!}{n^{(a+1)k_1} (n - (a+1)k_1)!} = 1 - \frac{n^{(a+1)k_1}}{n^{(a+1)k_1}}.$$

The sample size $(a + 1)k_1$ is a constant, and so we can choose N such that $p_2 \leq 1/6$ for $n > N$.

The probability that the tester obtains a sample that contains at least $(a + 1)$ -many distinct representatives of each color in $\text{Col}(A, \varepsilon n / (2 \cdot 2^{|M|}))$ is at least

$$1 - (a + 1)p_1 - p_2 \geq 2/3.$$

□ Lemma 13

Our goal is to show that if $\text{rdist}(A, P) \geq \varepsilon$, then we reject with probability at least $2/3$. It is easier to show the contrapositive: if the tester accepts with probability strictly greater than $1/3$, then $\text{rdist}(A, P) < \varepsilon$.

If we accept a structure A with probability strictly greater than $1/3$, then we must accept it when we obtain a good sample. We construct a B such that $B \models \varphi$ and $\text{rdist}(A, B) < \varepsilon$ from the A' that the tester must find to accept. We begin with Lemma 14, which we will use to “grow” smaller models.

Lemma 14. *Let $\varphi := \exists x_1 \dots \exists x_a \forall y \exists z_1 \dots \exists z_b : \psi$ be a formula with vocabulary τ , where ψ is quantifier-free and $A \in \text{STRUC}(\tau)$ be such that $A \models \varphi$. Additionally, let $B \in \text{STRUC}(\tau)$ be any structure containing A as an induced substructure such that $\#(B) = \#(A) + 1$. If the additional element of B has a color that occurs at least $a + 1$ times in A , then we can construct a $B' \models \varphi$ by modifying at most a constant number of non-monic assignments in B .*

PROOF. Structure B contains an induced copy of A and one additional element, which we will denote by q . By assumption, A is a model of φ and therefore contains an a -tuple (u_1, \dots, u_a) such that the formula is satisfied when x_i is bound to u_i . In addition, there are at least $a + 1$ elements in A that have the same color as q . Therefore, there is at least one such element p that is not one of the u_i . We will make q equivalent to p without modifying any monadic assignments.

We begin by modifying the assignments as needed to make the structure induced by (x_1, \dots, x_a, q) identical to that induced by (x_1, \dots, x_a, p) . This requires at most $\sum_{i=1}^s \sum_{j=1}^{a_i} \binom{a_i}{j} a^{a_i-j} = O(1)$ modifications, all of which are non-monic. There must be (v_1, \dots, v_b) in A such that ψ is satisfied when z_i is bound to v_i and y to p . We modify the assignments needed to make the structure induced by (q, v_1, \dots, v_b) identical

to that induced by $(p, v_1, \dots, v_b)^{14}$. This requires at most $\sum_{i=1}^s \sum_{j=1}^{a_i} \binom{a_i}{j} b^{a_i-j} = O(1)$ modifications, all of which are non-monadic. The result has $\#(A) + 1$ elements, models φ and was constructed from B by making a constant number of modifications to non-monadic assignments. \square Lemma 14

Let A be the structure that the tester is running on and A' be the structure found in Step 3 of the tester. As mentioned above, we will construct a $B \models \varphi$ from A' such that $B \models \varphi$ and $\text{rdist}(A, B) < \varepsilon$.

Note that there must exist at least one color in $\text{Col}(A, \varepsilon n / (2 \cdot 2^{|M|}))$ and assume that N is large enough that $\varepsilon n / (2 \cdot 2^{|M|}) \geq a + 1$. We first make a constant sized portion of A identical to A' . This requires at most $O(1)$ -many modifications to each relation. All colors in $\text{Col}(A, \varepsilon n / (2 \cdot 2^{|M|}))$ occur at least $a + 1$ times in A' , allowing us to recursively apply Lemma 14 and add the elements of A that have colors in $\text{Col}(A, \varepsilon n / (2 \cdot 2^{|M|}))$. This entails making $O(1)$ -many modifications to non-monadic relations (and none to monadic relations) at each step, for a total of $O(n)$ modifications to the non-monadic relations.

Finally, we consider the elements of A that have colors occurring at most $\varepsilon n / (2 \cdot 2^{|M|})$ times. There are at most $2^{|M|}$ such colors and at most $\varepsilon n / 2$ elements with these colors. We change the monadic assignments on such elements as required to give them colors contained in $\text{Col}(A, \varepsilon n / (2 \cdot 2^{|M|}))$. This requires at most $\varepsilon n / 2$ modifications to each of the monadic assignments. We again recursively apply Lemma 14 to A , making $O(1)$ modifications to non-monadic assignments at each step. The resulting structure is B and is such that $B \models \varphi$.

Finally, we show that $\text{rdist}(A, B) < \varepsilon$. If R_i is a monadic relation, then the i -th term of the maximum in the definition of rdist (cf. Definition 10 above) is at most $\varepsilon / 2 + o(1)$. If R_i has arity at least two, then the i -th term of the maximum is $O(n) / \Omega(n^2) = o(1)$. All $o(1)$ terms can be made arbitrarily small by choosing $N(\varphi, \tau, \varepsilon)$ appropriately and so we can assume that all terms are strictly less than ε . The maximum is then strictly less than ε and so $\text{rdist}(A, B) < \varepsilon$ as desired. \square Theorem 11

4.2. Ramsey's Class

In this section we revisit a result of Alon *et al.* [2] in the light of recent work by Austin and Tao [8]. The main result is the testability of the full Ramsey's class (i.e., removing the restriction to undirected loop-free graphs). As we did for Ackermann's class with equality in Subsection 4.1, we begin by reviewing the history and properties of the class, denoted $[\exists^* \forall^*, \text{all}]_{=}$.

Ramsey's class is also known as the Bernays-Schönfinkel-Ramsey class. Bernays and Schönfinkel [9] proved the finite model property and that satisfiability is decidable for the class *without* equality. Ramsey [38] extended these results to the class with equality as part of a stronger result. Lewis [33] showed that satisfiability is NEXPTIME-complete for Ramsey's class and Kolaitis and Vardi [29] proved that a 0-1 law holds for existential second-order logic where the first-order part belongs to $[\exists^* \forall^*, \text{all}]_{=}$. Omodeo and Policriti [36] have recently shown that the class is semidecidable for set theory.

The main goal of this subsection is Theorem 15 below. Recalling Theorem 5, this also implies testability in the \mathcal{T} and \mathcal{T}_r senses. The proof of Theorem 15 follows the proof by Alon *et al.* [2], and relies on a reduction to a strong result by Austin and Tao [8].

¹⁴The case where $v_i = p$ can be handled by replacing v_i with q in (q, v_1, \dots, v_b) .

An outline of the proof is as follows. First, we show that all sentences in $[\exists^*\forall^*, all]_{=}$ define properties which are indistinguishable from instances of a generalized colorability problem. Next, we note that all such problems are hereditary and therefore testable when mapped to the setting defined by Austin and Tao [8]. Finally, we show that this implies testability under our definitions, giving the following.

Theorem 15. *All sentences in $[\exists^*\forall^*, all]_{=}$ define properties in \mathcal{T}_{mr} .*

We begin the proof of Theorem 15 by defining a generalized colorability problem, as did Alon *et al.* [2].

For any fixed set F of structures with vocabulary τ , some positive number of colors c , and functions that assign a color between 1 and c to each element of each structure in F , we define the F -colorability problem as follows. A structure $A \in STRUC(\tau)$ is F -colorable if there exists some (not necessarily proper) c -coloring of A such that A does not contain any induced substructures isomorphic to a member of F . We let P_F be the set of structures that are F -colorable.

For example, we can consider the case of graphs and let F contain c copies of K_2 . We enumerate these copies in some fashion from 1 to c , and for copy i , color both vertices with i . The resulting problem is of course the usual (k - or equivalently) c -colorability. The following is a straightforward generalization of the proof by Alon *et al.* [2].

Lemma 16. *Let φ be any first-order sentence in the class $[\exists^*\forall^*, all]_{=}$. There exists an instance of the F -colorability problem that is indistinguishable from P , the property defined by φ .*

PROOF. Let $\varepsilon > 0$ be arbitrary and $\varphi := \exists x_1 \dots \exists x_t \forall y_1 \dots \forall y_u : \psi$ be any first-order formula with quantifier-free ψ and vocabulary τ . We note, as did Alon *et al.* [2], that we can restrict our attention to formulae ψ where it is sufficient to consider only cases where the variables are bound to distinct elements. This is because, given any ψ' , we can construct a ψ satisfying this restriction that is equivalent on structures with at least $t + u$ elements, and the smaller structures do not matter in the context of indistinguishability.

Let $P = \{A \mid A \in STRUC(\tau), A \models \varphi\}$ be the property defined by φ . We now define an instance of F -colorability that we will show to be indistinguishable from P . We denote our c colors by the elements of

$$\{(0, 0)\} \cup \{(a, b) \mid 1 \leq a \leq \pi_1, 1 \leq b \leq \pi_2, a, b \in \mathbb{N}\}.$$

Here, π_1 is the number of distinct structures of vocabulary τ with exactly t elements, $\pi_1 := 2^{\sum_{i=1}^s t^{a_i}}$. Similarly, we denote by π_2 the number of ways it is possible to “connect” or “add” a single element to some existing, fixed t -element structure of vocabulary τ , i.e., $\pi_2 := 2^{\sum_{i=1}^s \sum_{j=1}^{a_i-1} \binom{a_i}{j} t^{a_i-j}}$. We will use fixed enumerations of these π_1 structures with t elements and π_2 ways of connecting an additional element to a fixed t element structure.

We impose on the coloring of the structure the following restrictions. Each can be expressed by prohibiting finite sets of colored induced substructures.

- (1) The color $(0, 0)$ may be used at most t times. Therefore, we prohibit all $(t + 1)$ -element structures that are colored completely with $(0, 0)$.¹⁵

¹⁵Note that introducing a constraint guaranteeing the existence of t such elements cannot be done by forbidding finite sets of structures, and would result in a non-hereditary property.

- (2) The graph must be colored using only $\{(0,0)\} \cup \{(a,b) \mid 1 \leq b \leq \pi_2\}$ for some fixed $a \in \{1, \dots, \pi_1\}$. Therefore, we prohibit all two-element structures colored $((a,b), (a',b'))$ with $a \neq a'$.
- (3) We now consider some fixed coloring of a u -element structure V , whose universe we identify with $\{v_1, \dots, v_u\}$. We assume that this coloring satisfies the previous restriction and that color $(0,0)$ does not appear. We must decide whether to prohibit this structure. In order to do so, we first take the fixed a guaranteed by the previous restriction, and consider the t -element structure E , whose universe we identify with $\{e_1, \dots, e_t\}$, that is the a^{th} structure in our enumeration of t element structures. We connect each v_i to E in the following way. If v_i is colored (a,b) , we use the b^{th} way of connecting an additional element to a t -element structure in our enumeration. We denote the resulting $(t+u)$ -element structure as M and allow (do not prohibit) V iff M is a model of ψ when we replace x_i with u_i and y_j with v_j .

We now show that the resulting F -colorability problem is indistinguishable from P . Recall the definition of indistinguishability (Definition 14) and assume that we are given an $A \models \varphi$. Color the t vertices existentially bound to the x_i with $(0,0)$. Then, we can color all remaining vertices v_i with (a,b) , where a corresponds to the substructure induced by $\{x_1, \dots, x_t\}$ in our enumeration of t -element structures, and b corresponds to the connection between v_i and $\{x_1, \dots, x_t\}$. It is easy to see that this coloring satisfies the restrictions of our F -colorability problem. We have not made any modifications to the structure and so $\text{mrdist}(A, P_F) = 0$ (i.e., $A \in P_F$).

Next, we assume that we are given a structure with a coloring that satisfies our restrictions. We will show that we can obtain a model of φ by making only a small number of modifications. First, if there are less than t elements colored $(0,0)$, we arbitrarily choose additional elements to color $(0,0)$ so that there are exactly t such elements. We will denote these t elements with $\{e_1, \dots, e_t\}$. Restriction (2) guarantees that all colors which are not $(0,0)$ share the same first component. Let a be this shared component. We make the structure induced by $\{e_1, \dots, e_t\}$ identical to the a^{th} structure in our enumeration of t -element structures, requiring at most $\sum_{i=1}^s t^{a_i} = O(1)$ modifications. Next, for each element v_i that is colored (a,b) with $a, b \neq 0$, we modify the connections between v_i and $\{e_1, \dots, e_t\}$ in order to make these connections identical to the b^{th} way of making such connections in our enumeration. This requires at most

$$(n-t) \sum_{i=1}^r \sum_{j=1}^{a_i-1} \left[\binom{a_i}{j} t^{a_i-j} \right] = O(n)$$

additional modifications, all of which are to non-monadic subrelations. Binding x_i to e_i , the resulting structure is a model of φ . We made at most $O(1)$ modifications to monadic subrelations and $O(n)$ modifications to non-monadic subrelations, and so $\text{mrdist}(A, P) \leq \max\{O(1)/n, O(n)/\Omega(n^2)\} = o(1) < \varepsilon$, where the inequality holds for sufficiently large n .

Therefore, all such properties P are indistinguishable from instances of F -colorability, as desired. \square

Recall that a *hereditary property* of relational structures is one which is closed under taking induced substructures. F -colorability is clearly a hereditary property; if A is

F -colorable, then so are its induced substructures. However, the definitions of Austin and Tao [8] are significantly different from ours and so we explicitly reduce the following translation in our setting to their result.

Theorem 17 (Translation of Austin and Tao). *Let P be a hereditary property of relational structures which is closed under isomorphisms. Then, property P is testable in the sense of \mathcal{T}_{mr} with one-sided error.*

Theorem 17 can be viewed as the latest in a series of generalizations of Alon *et al.* [2] Corollary 6.3, i.e., the testability of colorability problems for undirected loop-free graphs. The first such generalization was by Fischer [16], who extended the result to more general colorability problems with counting restrictions. This was followed by Alon and Shapira [6] who extended it to hereditary graph properties. Ishigami [25] extended the testability result to hereditary partite uniform hypergraph properties, and Rödl and Schacht [39] extended it to hereditary uniform hypergraph properties. These generalizations are closely related to extensions of Szemerédi’s Regularity Lemma and the Removal Lemma, see the references in Subsection 1.1.

Before reducing Theorem 17 to its statement in [8], we first briefly introduce their definitions. All of the definitions in Subsection 4.2.1 are from Austin and Tao [8], although we omit definitions which are not necessary for our purposes.

4.2.1. Framework of Austin and Tao

We begin by introducing their analogue of vocabularies: *finite palettes*.

Definition 15. A *finite palette* K is a sequence $K := (K_j)_{j=0}^{\infty}$ of finite sets, of which all but finitely-many are singletons. The singletons are called *points* and denoted pt . A point is called *trailing* if it occurs after all non-points.

We will write $K = (K_0, \dots, K_k)$, omitting trailing points and call k the *order* of K . We use the elements of K_j to *color* the j -ary edges in hypergraphs.

Definition 16. A *vertex set* V is any set which is at most countable. If V, W are vertex sets, then a *morphism* f from W to V is any injective map $f: W \rightarrow V$ and the set of such morphisms is denoted $\text{Inj}(W, V)$. For $N \in \mathbb{N}$, we denote the set $\{1, \dots, N\}$ by $[N]$.

Of course, $[N]$ is a vertex set. Our structures are finite so we are mostly interested in *finite* vertex sets. Next, we define the analogue of relational structures.

Definition 17. Let V be a vertex set and K be a finite palette. A *K -colored hypergraph* G on V is a sequence $G := (G_j)_{j=0}^{\infty}$, where each $G_j: \text{Inj}([j], V) \rightarrow K_j$ is a function. Let $K^{(V)}$ be the set of K -colored hypergraphs on V .

Only finitely many of the K_j are not points, and so only finitely many G_j are non-trivial. The G_j assign colors from K_j to the morphisms in $\text{Inj}([j], V)$. In our relational setting, this set of morphisms corresponds to the set of j -ary tuples (x_1, \dots, x_j) with pairwise distinct components.

Before defining hereditary K -properties, we need one last technical definition.

Definition 18. Let V, W be vertex sets and $f \in \text{Inj}(W, V)$ be a morphism from W to V . The *pullback map* $K^{(f)}: K^{(V)} \rightarrow K^{(W)}$ is

$$\left(K^{(f)}(G)\right)_j(g) := G_j(f \circ g),$$

for all $G = (G_j)_{j=0}^\infty \in K^{(V)}$, $j \geq 0$ and $g \in \text{Inj}([j], W)$. If $W \subseteq V$ and $f \in \text{Inj}(W, V)$ is the identity map on W , we abbreviate

$$G \upharpoonright_W := K^{(f)}.$$

Abusing notation, the pullback map $K^{(f)}$ maps K -colored hypergraphs on V to those on W , by assigning the color of $f \circ g$ to g , for all tuples g . Note that $G \upharpoonright_W$ is equivalent to the induced subhypergraph on W . For notational clarity, we reserve P for properties of relational structures and use \mathcal{P} to denote properties of hypergraphs.

Definition 19. Let $K = (K_j)_{j=0}^\infty$ be a finite palette. A *hereditary K -property* \mathcal{P} is an assignment $\mathcal{P}: V \mapsto \mathcal{P}^{(V)}$ of a collection $\mathcal{P}^{(V)} \subseteq K^{(V)}$ of K -colored hypergraphs for every finite vertex set V such that

$$K^{(f)}(\mathcal{P}^{(V)}) \subseteq \mathcal{P}^{(W)}$$

for every morphism $f \in \text{Inj}(W, V)$ between finite vertex sets.

Finally, we state the definition of (one-sided error) testability used by Austin and Tao [8]. Here, for a vertex set V and $c \in \mathbb{N}$, we write $\binom{V}{c} := \{V' \mid V' \subseteq V, |V'| = c\}$ to denote the set of subsets of V with exactly c elements.

Definition 20. Let K be a finite palette with order $k \geq 0$ and \mathcal{P} be a hereditary K -property. Property \mathcal{P} is *testable with one-sided error* if for every $\varepsilon > 0$, there exists $N \geq 1$ and $\delta > 0$ satisfying the following. For all vertex sets V with $|V| \geq N$, if $G \in K^{(V)}$ satisfies

$$\frac{1}{\left|\binom{V}{N}\right|} \left| \left\{ W \mid W \in \binom{V}{N}, G \upharpoonright_W \in \mathcal{P}^{(W)} \right\} \right| \geq 1 - \delta, \quad (1)$$

then there exists a $G' \in \mathcal{P}^{(V)}$ satisfying

$$\frac{1}{\left|\binom{V}{k}\right|} \left| \left\{ W \mid W \in \binom{V}{k}, G \upharpoonright_W \neq G' \upharpoonright_W \right\} \right| \leq \varepsilon. \quad (2)$$

To see that this is a variant of testability, it is easiest to consider the contrapositive. If there is a G' satisfying (2), then G is not ε -far from \mathcal{P} , using the implicit distance measure based on the fraction of differing induced subhypergraphs of size k . If there is no such G' (i.e., G is ε -far from \mathcal{P}) and \mathcal{P} is testable, then (1) must not hold. That is, there are *many* induced subhypergraphs of size N that do not have \mathcal{P} . The definition is for hereditary \mathcal{P} , and so if G has \mathcal{P} , then so do all induced subhypergraphs. This allows the construction of testers.

Finally, we can state one of the main results of Austin and Tao [8].

Theorem 18 (Austin and Tao [8]). *Let K be a finite palette and let \mathcal{P} be a hereditary K -property. Then, \mathcal{P} is testable with one-sided error.*

In the following subsection we will map our vocabularies, structures and properties to this setting. We will then show that hereditary properties in our setting correspond to hereditary properties (in the sense of Definition 19 above) here, and that testability in the sense of this section (Definition 20) implies testability of the original relational properties. That is, we explicitly reduce our translation (Theorem 17) to Theorem 18.

4.2.2. Reducing Theorem 17 to Theorem 18

We begin by mapping vocabulary $\tau = \{R_1^{a_1}, \dots, R_s^{a_s}\}$ to a finite palette $K_\tau = (K_i)_{i=0}^\infty$. We use the color of a “tuple” to represent the set of assignments on it. The difference between the set of j -ary tuples over a finite universe U and $\text{Inj}([j], U)$ is that the latter does not permit repeated components. If $S \in \text{SUB}(R_i^{a_i})$ has $|S| < a_i$, then the corresponding subrelation consists of tuples with repeated components. We treat such S as relations with arity $|S|$ and no repeated components. Recall that $\mathfrak{S}(n, k)$ is the Stirling number of the second kind.

For $a \geq 1$, let $P_a := \{R_i^{a_i} \mid R_i^{a_i} \in \tau, a_i = a\}$ be the set of predicate symbols with arity a . We now define palette K . Let $K_0 := \text{pt}$ and $K_i := \left[2^{\sum_j |P_j| \mathfrak{S}(j, i)}\right]$. There are finitely-many predicate symbols and so only finitely-many $K_i \neq \text{pt}$.

Let $S_a := \{S_a^i \mid S_a^i \in \text{SUB}(R_i^{a_i}), |S_a^i| = a, 1 \leq i \leq s\}$ be the set of subtypes with cardinality a for all $a \geq 1$. Now, $2^{|S_a|} = |K_a|$ and we have exactly enough colors to encode the set of assignments of the a -ary subtypes on a -ary tuples.

We will now define a map h from relational structures A on universe U to hypergraphs $G_A \in K^{(U)}$. For any $S_a^i \in S_a$, there is a bijection

$$r(S_a^i): s^U(S_a^i) \rightarrow \{(x_1, \dots, x_a) \mid x_i \in U, x_i \neq x_j \text{ for } i \neq j\}$$

from $s^U(S_a^i)$ to the a -ary tuples without duplicate components, formed by removing the duplicate components. That is, $r(S_a^i)$ maps (x_1, \dots, x_{a_i}) to $(x_{i_1}, \dots, x_{i_a})$ where $1 \leq i_1 < i_2 < \dots < i_a \leq a_i$. We can now define $G_A = h(A)$.

For $j > 0$, we define $G_j: \text{Inj}([j], U) \rightarrow K_j$ as follows. Assign to $f \in \text{Inj}([j], U)$ the color encoding the set of assignments of the subtypes S_j on $(f(1), \dots, f(j))$, using the inverses $(r(S_j^i))^{-1}$ to get assignments for subtypes of high-arity relations. For $j = 0$, $\text{Inj}([j], U) = \emptyset$ and $K_0 = \text{pt}$ and we can use a trivial map.

Of course, we extend the map to properties in the obvious way. If P is a property of relational structures, we let $\mathcal{P}^{(U)} := \{h(A) \mid A \in P\}$. Formally, we define $\mathcal{P}(U) := \mathcal{P}^{(U)}$, but there is a small technical point. We have identified finite universes with subsets of the naturals, allowing us to call $\text{STRUC}(\tau)$ a *set*. However, Definition 16 in this section allows a vertex set to be *any* finite set and Definition 19 requires hereditary hypergraph properties to be closed under bijections between vertex sets. To remedy this, for each finite vertex set W , we fix a¹⁶ bijection $g^W: W \rightarrow \{0, \dots, |W| - 1\}$. We then define $\mathcal{P} := h(P)$ formally as

$$\mathcal{P}(W) := \begin{cases} \mathcal{P}^{(W)}, & \text{if } W = \{0, 1, \dots, |W| - 1\}; \\ K^{(g^W)}(\mathcal{P}^{\{0, \dots, |W| - 1\}}), & \text{otherwise.} \end{cases}$$

¹⁶Our properties are closed under isomorphisms, so any fixed bijection is acceptable.

Hereditary relational properties are mapped to hereditary hypergraph properties, which are testable in the sense of this section (Definition 20 above) by Theorem 18.

Lemma 19. *If P is a hereditary property of relational structures, then $h(P)$ is a hereditary property of hypergraphs.*

PROOF. Let P be a hereditary property of relational structures with vocabulary τ . Assume that $\mathcal{P} := h(P)$ is not a hereditary K -property. Then, by Definition 19 above, there exist finite vertex sets V and W , and a morphism $f' \in \text{Inj}(W, V)$ such that

$$K^{(f)}(\mathcal{P}^{(V)}) \not\subseteq \mathcal{P}^{(W)}. \quad (3)$$

Since f' exists, $\text{Inj}(W, V)$ cannot be the empty set and so $|V| \geq |W|$. Let $U_V := \{0, \dots, |V| - 1\}$ and $U_W := \{0, \dots, |W| - 1\}$. By the definition of \mathcal{P} , we can fix bijections $g^V: V \rightarrow U_V$ and $g^W: W \rightarrow U_W$ such that $\mathcal{P}^{(V)} = K^{(g^V)}(\mathcal{P}^{(U_V)})$ and $\mathcal{P}^{(W)} = K^{(g^W)}(\mathcal{P}^{(U_W)})$. By the definition of $\mathcal{P} = h(P)$, this implies

$$K^{(f)}\left(K^{(g^V)}\left(\mathcal{P}^{(U_V)}\right)\right) \not\subseteq K^{(g^W)}\left(\mathcal{P}^{(U_W)}\right).$$

Bijections are invertible, and so this implies

$$K^{(g^V \circ f' \circ (g^W)^{-1})}\left(\mathcal{P}^{(U_V)}\right) \not\subseteq \mathcal{P}^{(U_W)}.$$

Rename $f' := g^V \circ f' \circ (g^W)^{-1}$ and note $f' \in \text{Inj}(U_W, U_V)$. Let $A' \in \mathcal{P}^{(U_V)}$ be such that $K^{(f')}(A') \not\subseteq \mathcal{P}^{(U_W)}$.

We defined \mathcal{P} as $h(P)$ for a hereditary property P of relational structures. Property P is closed under isomorphisms, and so there is an $A := h^{-1}(A') \in P \cap \text{STRUC}^{|U_V|}(\tau)$ such that the $|U_W|$ -element substructure induced by $\{a \mid a = f'(u) \text{ for some } u \in U_w\}$ does not have P . This contradicts the hereditariness of P and so \mathcal{P} must be hereditary in the sense of this section (Definition 19). \square

We mapped hereditary relational properties to hereditary hypergraph properties, which are testable by Theorem 18. We will show this implies testability of the original properties.

Definition 21. Let $A, B \in \text{STRUC}^n(\tau)$ be structures with vocabulary τ and universe $U := \{0, \dots, n-1\}$ of size n , $k := \max_i a_i$ be the maximum arity of the predicate symbols, and $h: \text{STRUC}^n(\tau) \rightarrow K^{(U)}$ be the map defined above. The h -distance between A and B is

$$\text{hdist}(A, B) := \frac{1}{\left| \binom{U}{k} \right|} \left| \left\{ W \mid W \in \binom{U}{k}, h(A) \upharpoonright_W \neq h(B) \upharpoonright_W \right\} \right|.$$

We now relate the two distances with the following simple lemma.

Lemma 20. *Let $A, B \in \text{STRUC}^n(\tau)$ be relational structures with vocabulary τ and size n . Then, $\text{hdist}(A, B) \geq \text{mrdist}(A, B)$.*

PROOF. Assume that $\text{mrdist}(A, B) = \varepsilon$. Then, there exists a predicate symbol $R_i^{a_i} \in \tau$ and subtype $S \in \text{SUB}(R_i^{a_i})$ such that $|s^A(S) \Delta s^B(S)| / (n! / (n - |S|)!) = \varepsilon$. Let $k := \max_i a_i$ and let the universe of both structures be $U_n := \{0, \dots, n - 1\}$.

Consider a random permutation of the universe (i.e., a bijection $r: U_n \rightarrow U_n$) chosen uniformly from the set of such permutations. The probability that the substructures induced on $\{r(0), \dots, r(k - 1)\}$ differ in A and B is $\text{hdist}(A, B)$. The probability that the tuple of the first $|S|$ elements, i.e. $(r(0), \dots, r(|S| - 1))$, differ in $s^A(S)$ and $s^B(S)$ is ε and so $\text{hdist}(A, B) \geq \varepsilon$. \square

Equality is obtained when $|S| = k$. It is possible to show that the two distances differ by at most a constant factor, and so the corresponding notions of testability are essentially equivalent. However, Lemma 20 suffices for our purposes.

Lemma 21. *Let $P \subseteq \text{STRUC}(\tau)$ be a property of relational structures which is mapped by h to a property of hypergraphs that is testable with one-sided error. Then, P is testable with one-sided error.*

PROOF. Let $\mathcal{P} := h(P)$ be the hypergraph property which P is mapped to. We show that the following is an ε -tester for P with one-sided error. Let $N \geq 1$, $\delta > 0$ be the constants of Definition 20 above for ε . Assume that we are testing a structure $A \in \text{STRUC}^n(\tau)$ and recall that $U = \{0, \dots, n - 1\}$.

1. If $\#(A) \leq N$, query the entire structure and decide exactly whether $A \in P$.
2. Otherwise, repeat the following $q(\delta)$ times.
 - (a) Uniformly select N elements and query the induced substructure.
 - (b) If it has P , continue. Otherwise, reject.
3. Accept if all of the induced substructures had P .

If $A \in P$, then all induced substructures have P because P is hereditary and the tester accepts with probability 1. Next, assume $\text{mrdist}(A, P) > \varepsilon$. We use Definition 20 above to show the tester will find a witness for $A \notin P$ with probability at least $2/3$. By Lemma 20, $\text{hdist}(A, P) \geq \text{mrdist}(A, P) > \varepsilon$. We assumed $h(P)$ is hereditary, and so (by Theorem 18) it is testable in the sense of Definition 20. The probability that a uniformly chosen N -element substructure does not have P is at least δ . We use $q(\delta)$ to amplify the success probability from δ to $2/3$. \square

This completes the proof of the testability of Ramsey's class (Theorem 15). All properties expressible in Ramsey's class are indistinguishable from instances of F -colorability. Indistinguishability preserves testability and so it sufficed to show that these instances are testable. All instances of F -colorability are hereditary relational properties, which are testable by Theorem 17, which we reduced to the statement by Austin and Tao [8].

5. Untestable Classes

We now shift our attention to untestable classes. As mentioned in Subsection 1.2, Alon *et al.* [2] proved that there exists an untestable property of undirected, loop-free graphs expressible with quantifier prefix $\forall^{12}\exists^5$. We simplify their untestable example and thereby show that the classes $[\forall^3\exists, (0, 1)]_=$, $[\forall^2\exists\forall, (0, 1)]_=$, $[\forall\exists\forall^2, (0, 1)]_=$ and

$[\forall\exists\forall\exists, (0, 1)]_=$ are untestable. The focus on graphs is justified by recalling that monadic first-order logic is testable. We begin by briefly reviewing other properties of these classes.

The class $[\forall^3\exists, (0, 1)]$ (usually without equality) is well-known in the literature. It is trivial to prove that this class does not have the finite model property. In addition, Kolaitis and Vardi [30] showed that a 0-1 law does not hold for second-order existential logic when the first order part is in this class (even without equality). However, it is an essentially finite class (i.e., it can only express a finite number of properties) and therefore decidable.

The other classes of this section contain the well-known class $[\forall\exists\forall, (0, 1)]$, which is a subclass of the Kahr-Moore-Wang [28] class. We have recently shown that there are untestable graph properties in this class, see Jordan and Zeugmann [27]. It is again trivial to prove that $[\forall\exists\forall, (0, 1)]$ does not have the finite model property. Vedø [50] showed that a 0-1 law does not hold for second-order existential logic when the first-order part is in this class (again, even without equality). However, as above, all classes in this section are essentially finite and therefore decidable.

We will begin by defining property P , which is essentially the graph isomorphism problem for undirected loop-free graphs encoded in directed graphs that may contain loops. We will begin by showing in Lemma 23 that P is indistinguishable from property P_f (cf. Definition 23 below) which is expressible in any of the prefix vocabulary classes mentioned in Theorem 22 below. We will then show that P is not testable. Indistinguishability preserves testability and so this implies that P_f is also untestable, which will suffice to show the following theorem.

Theorem 22. *The following prefix classes are not testable:*

1. $[\forall\exists\forall\exists, (0, 1)]_=$
2. $[\forall\exists\forall^2, (0, 1)]_=$
3. $[\forall^2\exists\forall, (0, 1)]_=$
4. $[\forall^3\exists, (0, 1)]_=$

We define property P as follows. First, a graph that has property P must consist of an even number of vertices, of which exactly half have loops. The subgraph induced by the vertices with loops must be isomorphic to that induced by the vertices without loops, ignoring all loops, and there must be no edges connecting the vertices with loops to those without loops. Finally, all edges must be *undirected* (i.e., an edge from x to y implies an edge from y to x). We refer to such undirected edges as *paired edges*.

Definition 22. A graph $G \in \mathcal{G}^n$ has P iff the following conditions are satisfied:

1. For some s , $n = 2s$.
2. There are exactly s vertices x satisfying $E(x, x)$. We will refer to the set of such vertices as H_1 and to the remaining s vertices as H_2 .
3. The substructure induced by H_1 is isomorphic to that induced by H_2 when all loops are removed. That is, there is a bijection f from H_1 to H_2 such that for distinct $x, y \in H_1$, it is true that $G \models E(x, y)$ iff $G \models E(f(x), f(y))$.
4. There are no edges between H_1 and H_2 .
5. All edges are paired.

Graph isomorphism is not directly expressible in first-order logic, and so we use the following encoding where the bijection f is made explicit by adding n edges between H_1 and H_2 . This of course reduces the complexity from the level of finding an isomorphism to the level of checking a given one, in order to achieve first-order expressivity. However, it maintains hardness for testability: essentially, our samples are too small to see any part of the given isomorphism.

Definition 23. A graph $G \in \mathcal{G}^n$ has P_f iff the following conditions are satisfied:

1. For every vertex x , if $E(x, x)$ then there is an edge from x to exactly one y such that $\neg E(y, y)$.
2. For every vertex x , if $\neg E(x, x)$ then there is an edge from x to exactly one y such that $E(y, y)$.
3. For all vertices x and y , $E(x, y)$ iff $E(y, x)$.
4. For all vertices x_1, x_2, x_3, x_4 that are pairwise distinct, if $E(x_1, x_1)$, $\neg E(x_2, x_2)$, $E(x_3, x_3)$, $\neg E(x_4, x_4)$, $E(x_1, x_2)$ and $E(x_3, x_4)$, then $E(x_1, x_3)$ iff $E(x_2, x_4)$.

Expressing Conditions 1 and 2 as “there is at most one such y ” and “there is at least one such y ,” P_f can be expressed in each of the classes $[\forall\exists\forall\exists, (0, 1)]_=$, $[\forall\exists\forall^2, (0, 1)]_=$, $[\forall^2\exists\forall, (0, 1)]_=$ and $[\forall^3\exists, (0, 1)]_=$.

For example, in the class $[\forall^3\exists, (0, 1)]_=$, we can express P_f by

$$\begin{aligned} \forall x_1 \forall x_3 \forall x_4 \exists x_2 : [& \\ & ((E(x_1, x_1) \leftrightarrow \neg E(x_2, x_2)) \wedge E(x_1, x_2)) \quad \wedge \\ & [((E(x_1, x_1) \leftrightarrow \neg E(x_3, x_3)) \wedge (E(x_3, x_3) \leftrightarrow E(x_4, x_4))) \wedge \\ & \quad E(x_1, x_3) \wedge E(x_1, x_4)) \rightarrow x_3 = x_4] \quad \wedge \\ & (E(x_1, x_3) \rightarrow E(x_3, x_1)) \quad \wedge \\ & ([E(x_1, x_1) \wedge E(x_3, x_3) \wedge x_1 \neq x_3 \wedge \neg E(x_4, x_4) \wedge E(x_3, x_4)] \rightarrow \\ & \quad (\neg E(x_2, x_2) \wedge E(x_1, x_2) \wedge (E(x_1, x_3) \leftrightarrow E(x_2, x_4)))))] . \end{aligned}$$

To express P_f with prefixes $\forall^2\exists\forall$ and $\forall\exists\forall^2$, it suffices to reorder the quantifiers (keeping x_2 existential and x_1 first). The prefix $\forall\exists\forall\exists$ requires a few additional modifications.

The two properties P and P_f differ only in the edges which make the isomorphism explicit in P_f but are forbidden in P . There are at most n such edges, none of which are loops. This suffices to prove the following.

Lemma 23. *Properties P and P_f are indistinguishable.*

PROOF. Let $\varepsilon > 0$ be arbitrary and let $N_\varepsilon = \varepsilon^{-1}$. Assume that G is a structure that has property P and that $\#(G) > N_\varepsilon$. We will show that $\text{mrdist}(G, P_f) < \varepsilon$.

Structure G has P and so there is a bijection f satisfying Condition 3 of Definition 22. For all $x \in H_1$, we add the edges $E(x, f(x))$ and $E(f(x), x)$ and call the result G' . Property P_f differs from P only in that the isomorphism is made explicit by the edges connecting loops and non-loops, and so G' has P_f . Indeed, it satisfies Conditions 1 and 2 of Definition 23 because G had no edges between loops and non-loops and we have

connected each to exactly one of the other, following the bijection f . Next, G' satisfies Condition 3 of Definition 23 because G satisfied Condition 5 of Definition 22 and we added only paired edges. Finally, G' satisfies Condition 4 of Definition 23 because the edges between loops and non-loops follow the isomorphism f from Condition 3 of Definition 22.

We have added exactly n (directed) edges, none of which are loops and so we have $\text{mrdist}(G, P) \leq \text{mrdist}(G, G') = 0 + n/n^2 < \varepsilon$, where the inequality holds for $n > N_\varepsilon$. The converse is analogous; given a G that has property P_f , we simply remove the n edges between loops and non-loops after using them to construct the isomorphism f . \square

Properties P and P_f are indistinguishable. We saw in Section 3 that testability is preserved by indistinguishability (cf. Theorem 10) and thus showing that P is not testable suffices to prove that P_f is not testable (and therefore Theorem 22). The proof closely follows that of Alon *et al.* [2]. The crucial lemma is the following, a combination of Lemmata 7.3 and 7.4 from Alon *et al.* [2]. We use $\text{count}_H(T)$ to refer to the number of times that a graph T occurs as an induced subgraph in H . A *bipartite graph* is a graph where we can partition the vertices into two sets H_1 and H_2 such that there are no edges “internal” to the partitions. That is, for all $x_1, y_1 \in H_1$ and $x_2, y_2 \in H_2$, $\neg E(x_1, y_1)$ and $\neg E(x_2, y_2)$. See Jordan and Zeugmann [26] for an explicit proof of the following, which is somewhat technical and long.

Lemma 24 (Alon *et al.* [2]). *There exists a constant $\varepsilon' > 0$ such that for every $D \in \mathbb{N}$, there exist two undirected bipartite graphs $H = H(D)$ and $H' = H'(D)$, and a number t satisfying the following conditions.*

1. Both H and H' have a bipartition into classes U_1 and U_2 , each of size t .
2. In both H and H' , for all subgraphs X with size $t/3 \leq \#(X) \leq t$, there are more than $t^2/18$ undirected edges between X and the remaining part of the graph.
3. The minimum degree of both H and H' is at least $t/3$.
4. $\text{dist}(H, H') \geq \varepsilon'$.
5. For all D -element graphs T , $\text{count}_H(T) = \text{count}_{H'}(T)$.

It is worth noting that the above is for undirected, loop-free graphs. However, bipartite graphs never have loops and “undirected” in our setting results in paired edges. It is easy to show that if two structures agree on the counts for all size D induced subgraphs, they agree on the counts for all induced subgraphs of size at most D . This is done by applying the following lemma inductively.

Lemma 25. *Let H and H' be two graphs, both of size s , and let $2 < D \leq s$. If for every graph T of size D , $\text{count}_H(T) = \text{count}_{H'}(T)$, then for every graph T' of size $D - 1$, $\text{count}_H(T') = \text{count}_{H'}(T')$.*

PROOF. Assume H and H' satisfy the initial conditions of Lemma 25, but that there exists a T' of size $D - 1$ such that $\text{count}_H(T') \neq \text{count}_{H'}(T')$. Let $C = \{T \mid \#(T) = D \text{ and } T \text{ contains } T' \text{ as an induced subgraph}\}$.

Note that $\sum_{T \in C} \text{count}_H(T) \text{count}_T(T') = \text{count}_H(T')(s - D + 1)$ and likewise for $\sum_{T \in C} \text{count}_{H'}(T) \text{count}_T(T')$. We have assumed that H and H' satisfy $\text{count}_H(T) = \text{count}_{H'}(T)$ for $T \in C$, but $\text{count}_H(T') \neq \text{count}_{H'}(T')$, giving a contradiction and the Lemma follows. \square

Lemma 26. *Property P is not testable.*

PROOF. Assume that P is testable. Then, there exists an ε -tester for

$$\varepsilon := \min \{ \varepsilon' / 8, 1/144 \},$$

where ε' is the constant from Lemma 24 above. We can assume without loss of generality that the tester queries all edges in a random sample of $D := D(\varepsilon)$ vertices.

Consider the graph G which contains two copies of the $H = H(D)$ from Lemma 24, where one of the copies is marked by loops on each vertex and there are no edges between the copies. This graph has property P , and so the tester must accept it with probability at least $2/3$. Next, consider the graph G' which contains one copy of H marked by loops and one copy of H' , again where there are no edges between the two (induced) subgraphs. Graph G' is such that $\text{dist}(G', P) \geq \varepsilon$ (cf. Lemma 27 above) and so it must be rejected with probability at least $2/3$. Both G and G' consist of two bipartite graphs, each of which has a bipartition into two classes of size t , and so $\#(G) = \#(G') = 4t$.

However, G and G' both contain exactly the same number of each induced subgraph with D vertices. This is because both have loops on exactly half of the vertices and the two halves are not connected by any edges. Some of the D vertices must be in the first copy of H and the others in the second H (resp. H'). By Lemma 25 above, H and H' contain the same number of each induced subgraph with size at most D . The tester therefore obtains any fixed sample with the same probability in G and G' and is unable to distinguish between them. Hence, it is unable to accept G with probability $2/3$ and also reject G' with probability $2/3$. This completes the proof, taking into account Lemma 27 below. \square

Recall that testing is easiest under the dist definition, and so Lemma 26 also implies P is not testable under other definitions.

Lemma 27. *The graph G' is such that $\text{dist}(G', P) \geq \varepsilon$.*

PROOF. Suppose that $\text{dist}(G', P) < \varepsilon$. Then, there is an $M \in P$ such that $\text{dist}(G', M) < \varepsilon$. Let M_1 be the set of vertices with loops in M and let M_2 be the set of vertices without loops. We will refer to the subgraph induced by the vertices with loops in G' as H and to that induced by those without loops as H' . Without loss of generality, assume that $|M_1 \cap H| \geq |M_1 \cap H'|$. Then, $|M_1 \cap H| \geq t$. We let α_1 be the set $M_1 \setminus H$ and α_2 be $M_2 \setminus H'$. Note that $|\alpha_1| = |\alpha_2|$ and $|\alpha_1| \leq t$ because $|M_1 \cap H| \geq t$.

Informally, M is formed by moving the vertices α_1 from H' to H and the vertices α_2 from H to H' , and then possibly making other changes. There are three cases, which we will consider in order.

1. $|\alpha_1| = 0$.
2. $|\alpha_1| \geq t/3$.
3. $0 < |\alpha_1| < t/3$.

If $|\alpha_1| = 0$, then we can construct M from G' without exchanging vertices between H and H' , and in particular, construct H' from H (ignoring loops), by making less than $\varepsilon(4t)^2$ modifications. However, $\text{dist}(H, H') \geq \varepsilon'$ by Lemma 24 above and so this must

require at least $\varepsilon'(2t)^2$ modifications. By definition, $\varepsilon < \varepsilon'/4$ so $\varepsilon(4t)^2 < \varepsilon'(2t)^2$. The first case is therefore not possible.

Recall that $|\alpha_1| \leq t$. If $|\alpha_1| \geq t/3$, then by Condition 2 of Lemma 24 there exists at least $t^2/18$ undirected edges between α_1 and $H' \setminus \alpha_1$ and between α_2 and $H \setminus \alpha_2$. All of these edges must be removed to satisfy P because each would connect a vertex with a loop to a vertex without a loop. Therefore,

$$\text{dist}(G', M) \geq \frac{4t^2/18}{(4t)^2} = 1/72.$$

But, $\varepsilon < 1/72$ and so the second case is not possible.

Therefore, it must be that $0 < |\alpha_1| < t/3$. Here, we will show that it must be the case that α_1 and α_2 are relatively far apart. If they are not far apart, then it is possible to modify them instead of swapping them. This essentially results in the first case considered above. Condition 3 of Lemma 24 requires that each vertex has relatively high degree. These edges can be either internal to α_1 (resp. α_2) or connecting α_1 (α_2) with $H' \setminus \alpha_1$ ($H \setminus \alpha_2$). If α_1 and α_2 are relatively far apart, then we will see that this forces too many edges “outside” of α_1 (resp. α_2), resulting in a similar situation to the second case considered above.

We have assumed that $\text{dist}(G', M) < \varepsilon$ and that we can construct M from G' by making less than $\varepsilon(4t)^2$ modifications if we move α_1 to H and α_2 to H' . This entails the following modifications.

1. Removing all edges connecting α_1 to $H' \setminus \alpha_1$.
2. Removing all edges connecting α_2 to $H \setminus \alpha_2$.
3. Adding any required edges between α_1 and $H \setminus \alpha_2$.
4. Adding any required edges between α_2 and $H' \setminus \alpha_1$.
5. Changing α_1 , α_2 , $H \setminus \alpha_2$ and $H' \setminus \alpha_1$ to their final forms.

We can assume that the total number of modifications is less than $\varepsilon(4t)^2$. It must be that $\text{dist}(\alpha_1, \alpha_2)|\alpha_1|^2/(4t)^2 + \varepsilon \geq \varepsilon'/4$. If this does not hold, then we could first modify α_1 to make it identical to α_2 and then make H' identical to M_2 . Next, M_2 is identical to M_1 , which we could make identical to H . This would require less than $\varepsilon'(2t)^2$ modifications, which would violate Lemma 24. Therefore,

$$\text{dist}(\alpha_1, \alpha_2) \geq \frac{16(\varepsilon'/4 - \varepsilon)t^2}{|\alpha_1|^2}. \quad (4)$$

If both α_1 and α_2 are complete graphs then they cannot be far apart. Given that all vertices in α_1 (α_2 is analogous) have degree at least $t/3$, then there must be at least

$$|\alpha_1|(t/3 - |\alpha_1| + 1) + 2r$$

edges connecting α_1 to $H' \setminus \alpha_1$, where r is the number of edges internal to α_1 that must be omitted to satisfy (4). The simple lower bound on r , the number of edges needed for two graphs with at most r edges to be $\text{dist}(\alpha_1, \alpha_2)$ -far, that follows from $\text{dist}(\alpha_1, \alpha_2) \leq 2r/|\alpha_1|^2$ is sufficient. Finally, combining this with Inequality (4) yields

$$r \geq 8(\varepsilon'/4 - \varepsilon)t^2. \quad (5)$$

The number of edges connecting α_1 to $H' \setminus \alpha_1$ is therefore, by (5), at least

$$|\alpha_1|(t/3 - |\alpha_1| + 1) + 16(\varepsilon'/4 - \varepsilon)t^2 \geq 16(\varepsilon'/4 - \varepsilon)t^2.$$

All of these edges must be removed to move α_1 (resp. α_2), and so

$$\text{dist}(G', M) \geq \frac{16(\varepsilon'/4 - \varepsilon)t^2}{(4t)^2} = \frac{\varepsilon'}{4} - \varepsilon.$$

We have defined $\varepsilon \leq \varepsilon'/8$ and so $\text{dist}(G', M) \geq \varepsilon$, a contradiction.

The cases are exhausted and so $\text{dist}(G', P) \geq \varepsilon$ as desired. \square

6. Conclusions

We have focused on the testability of prefix-vocabulary classes of first-order logic, extending work that was initiated by Alon *et al.* [2]. Alon *et al.* [2] showed that all properties of undirected, loop-free graphs expressible in first-order sentences with quantifier pattern $\exists^* \forall^*$ are testable, while there exists an untestable property expressible with quantifier pattern $\forall^* \exists^*$. Their proof of the latter result implies upper bounds of twelve, five and seventeen for the minimum number of universal, existential and total quantifiers, respectively, sufficient to express an untestable property. One of our goals was to optimize these bounds and find the minimum number of universal and existential quantifiers, as well as quantifiers in total, sufficient to express an untestable property. Our results imply that these minima are two universal, one existential and three total quantifiers, respectively. In addition, we remove the restriction to undirected, loop-free graphs and focus on relational structures.

Our main results are as follows. First, we proved that all properties expressible in Ackermann's class with equality ($[\exists^* \forall \exists^*, all]_=$) are testable. Then, we extended the positive result of Alon *et al.* [2] from undirected, loop-free graphs to relational structures by using a result from Austin and Tao [8]. This answers a question of Fischer [16] on the testability of hypergraph properties expressible with quantifier pattern $\exists^* \forall^*$, although much of the work for this case is by Austin and Tao [8]. Finally, we simplified the untestable property of Alon *et al.* [2] and showed that there are untestable properties of directed graphs expressible with quantifier prefixes $\forall^3 \exists$, $\forall^2 \exists \forall$, $\forall \exists \forall^2$, and $\forall \exists \forall \exists$. Recently, we have shown in Jordan and Zeugmann [27] that there are untestable properties of directed graphs expressible with prefix $\forall \exists \forall$, improving on three of these prefixes.

The current classification of prefix-vocabulary classes for testability is the following.

- Testable classes
 1. Monadic first-order logic: $[all, (\omega)]_=$.
 2. Ackermann's class with equality: $[\exists^* \forall \exists^*, all]_=$.
 3. Ramsey's class: $[\exists^* \forall^*, all]_=$.
- Untestable classes
 1. $[\forall^3 \exists, (0, 1)]_=$.
 2. $[\forall \exists \forall, (0, 1)]_=$.

It is interesting to compare this classification for testability with known (complete) classifications for other properties. For example, the current classification for testability is consistent with the classifications for the finite model property (see, e.g., Chapter 6 of Börger *et al.* [11]), for docility¹⁷ (see Kolaitis and Vardi [31]) and for 0-1 laws for fragments of existential second-order logic (see Kolaitis and Vardi [31]). These classifications may be helpful in providing guidance in the classification for testability.

This similarity between classifications may indicate a deeper connection between these seemingly distinct properties. We would like to know which (if any) of the traditional classifications coincides with the classification for testability, and hope to understand the connections between testability and other properties of prefix classes.

As concrete open problems, we are especially interested in the testability of $[\forall\exists\forall, (0, 1)]$ and $[\forall^3\exists, (0, 1)]$ (without equality) and variants of the Gödel class (i.e., classes whose prefix contain at least $\forall^2\exists$). Determining the testability of these classes may suffice to complete the classification for the special case of predicate logic with equality.

There are also *many* possible variations of the classification for testability. For example, one could be more interested in classes which are *constructively* testable, i.e., where it is possible to compute an ε -tester given ε and a formula from the class. However, in the present paper we are fortunate that many of the possible classifications coincide. Namely, all of our positive results are for constructive (and therefore uniform) testability in the most-restricted model (\mathcal{T}_{mr}) that we consider, while all of the negative results hold even for non-uniform testability in the least-restricted model (\mathcal{T}).

Acknowledgments

We would like to thank an anonymous referee from LATA for significantly improving Theorem 22 in a preliminary version of this paper by eliminating one variable from each prefix, adding $\forall\exists\forall\exists$ and simplifying the example following Definition 23.

We also thank the anonymous referees for their careful reading and valuable comments.

- [1] Wilhelm Ackermann. Über die Erfüllbarkeit gewisser Zählausdrücke. *Math. Annalen*, 100:638–649, 1928.
- [2] Noga Alon, Eldar Fischer, Michael Krivelevich, and Mario Szegedy. Efficient testing of large graphs. *Combinatorica*, 20(4):451–476, 2000.
- [3] Noga Alon, Eldar Fischer, Ilan Newman, and Asaf Shapira. A combinatorial characterization of the testable graph properties: It’s all about regularity. In *STOC ’06: Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 251–260, New York, NY, USA, 2006. ACM.
- [4] Noga Alon, Michael Krivelevich, Ilan Newman, and Mario Szegedy. Regular languages are testable with a constant number of queries. *SIAM J. Comput.*, 30(6):1842–1862, 2001.
- [5] Noga Alon and Asaf Shapira. Homomorphisms in graph property testing. In M. Klazar, J. Kratochvíl, M. Loeb, J. Matoušek, R. Thomas, and P. Valtr, editors, *Topics in Discrete Mathematics*, volume 26 of *Algorithms and Combinatorics*, pages 281–313. Springer, 2006.
- [6] Noga Alon and Asaf Shapira. A characterization of the (natural) graph properties testable with one-sided error. *SIAM J. Comput.*, 37(6):1703–1727, 2008.
- [7] Noga Alon and Asaf Shapira. A separation theorem in property testing. *Combinatorica*, 28(3):261–281, 2008.
- [8] Tim Austin and Terence Tao. On the testability and repair of hereditary hypergraph properties. *Random Struct. Algorithms*, 36(4):373–463, 2010.

¹⁷A class is said to be *docile* (or decidable for finite satisfiability) if given an arbitrary formula from the class, one can decide if there exists a *finite* model of the property.

- [9] Paul Bernays and Moses Schönfinkel. Zum Entscheidungsproblem der mathematischen Logik. *Math. Annalen*, 99(1):342–372, 1928.
- [10] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *J. of Comput. Syst. Sci.*, 47(3):549–595, 1993.
- [11] Egon Börger, Erich Grädel, and Yuri Gurevich. *The Classical Decision Problem*. Springer-Verlag, 1997.
- [12] J. Richard Büchi. Weak second-order arithmetic and finite-automata. *Z. Math. Logik Grundlagen Math.*, 6:66–92, 1960.
- [13] Hana Chockler and Orna Kupferman. ω -regular languages are testable with a constant number of queries. *Theoret. Comput. Sci.*, 329(1-3):71–92, 2004.
- [14] Christian Fermüller and Gernot Salzer. Ordered paramodulation and resolution as decision procedure. In *Logic Programming and Automated Reasoning, 4th International Conference, LPAR'93, St. Petersburg, Russia, July 13-20, 1993, Proceedings*, volume 698 of *Lecture Notes in Computer Science*, pages 122–133. Springer, 1993.
- [15] Eldar Fischer. The art of uninformed decisions. *Bulletin of the European Association for Theoretical Computer Science*, 75:97–126, October 2001. Columns: Computational Complexity.
- [16] Eldar Fischer. Testing graphs for colorability properties. *Random Struct. Algorithms*, 26(3):289–309, 2005.
- [17] Eldar Fischer, Arie Matsliah, and Asaf Shapira. Approximate hypergraph partitioning and applications. *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2007*, pages 579–589, 2007.
- [18] Rūsiņš Freivalds. Fast probabilistic algorithms. In *Mathematical Foundations of Computer Science 1979, Proceedings, 8th Symposium, Olomouc, Czechoslovakia, September 3-7, 1979*, volume 74 of *Lecture Notes in Computer Science*, pages 57–69. Springer-Verlag, 1979.
- [19] John Gill. Computational complexity of probabilistic Turing machines. *SIAM J. Comput.*, 6(4):675–695, 1977.
- [20] O. Goldreich and D. Ron. Property testing in bounded degree graphs. *Algorithmica*, 32:302–343, 2002.
- [21] Oded Goldreich. Introduction to testing graph properties. Technical Report TR10-082, Electronic Colloquium on Computational Complexity (ECCC), May 2010.
- [22] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998.
- [23] W. T. Gowers. Hypergraph regularity and the multidimensional Szemerédi theorem. *Ann. of Math. (2)*, 166(3):897–946, 2007.
- [24] Erich Grädel. Satisfiability of formulae with one \forall is decidable in exponential time. *Arch. Math. Logic*, 29:256–276, 1990.
- [25] Yoshiyasu Ishigami. Removal lemma for infinitely-many forbidden hypergraphs and property testing. Preprint, available on arXiv.org as arXiv:math/0612669v2, 2008.
- [26] Charles Jordan and Thomas Zeugmann. Contributions to the classification for testability: Four universal and one existential quantifier. Technical Report TCS-TR-A-09-39, Hokkaido University, Division of Computer Science, November 2009.
- [27] Charles Jordan and Thomas Zeugmann. Untestable properties in the Kahr-Moore-Wang class. In Lev Beklemishev and Ruy de Queiroz, editors, *Logic, Language, Information and Computation, 18th International Workshop, WoLLIC 2011, Philadelphia, USA, May 2011, Proceedings*, volume 6642 of *Lecture Notes in Computer Science*, pages 176–186. Springer, 2011.
- [28] A. S. Kahr, Edward F. Moore, and Hao Wang. Entscheidungsproblem reduced to the $\forall\exists\forall$ case. *Proc. Nat. Acad. Sci. U.S.A.*, 48:365–377, 1962.
- [29] P. Kolaitis and M. Vardi. The decision problem for the probabilities of higher-order properties. In *STOC '87: Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, pages 425–435, New York, NY, USA, 1987. ACM.
- [30] Phokion G. Kolaitis and Moshe Y. Vardi. 0-1 laws and decision problems for fragments of second-order logic. *Inf. Comput.*, 87(1-2):302–338, 1990.
- [31] Phokion G. Kolaitis and Moshe Y. Vardi. 0-1 laws for fragments of existential second-order logic: A survey. In Mogens Nielsen and Branislav Rován, editors, *Mathematical Foundations of Computer Science 2000, 25th International Symposium, MFCS 2000, Bratislava, Slovakia, August/September 2000, Proceedings*, volume 1893 of *Lecture Notes in Computer Science*, pages 84–98. Springer, 2000.
- [32] K. de Leeuw, E. F. Moore, C. E. Shannon, and N. Shapiro. Computability by probabilistic machines. In C.E. Shannon and J. McCarthy, editors, *Automata Studies*, pages 183–212. Princeton University Press, Princeton, NJ, 1956.

- [33] Harry R. Lewis. Complexity results for classes of quantificational formulas. *J. Comput. Syst. Sci.*, 21(3):317–353, 1980.
- [34] Leopold Löwenheim. Über Möglichkeiten im Relativkalkül. *Math. Annalen*, 76:447–470, 1915.
- [35] Robert McNaughton and Seymour Papert. *Counter-Free Automata*. M.I.T. Press, 1971.
- [36] Eugenio Omodeo and Alberto Policriti. The Bernays-Schönfinkel-Ramsey class for set theory: Semidecidability. *J. Symbolic Logic*, 75(2):459–480, June 2010.
- [37] Michal Parnas and Dana Ron. Testing the diameter of graphs. *Random Struct. Algorithms*, 20(2):165–183, 2002.
- [38] F. P. Ramsey. On a problem of formal logic. *Proc. London Math. Soc. (2)*, 30:264–286, 1930.
- [39] Vojtěch Rödl and Mathias Schacht. Generalizations of the removal lemma. *Combinatorica*, 29(4):467–501, 2009.
- [40] Vojtěch Rödl and Mathias Schacht. Regular partitions of hypergraphs: Regularity lemmas. *Combin. Probab. Comput.*, 16:833–885, 2007.
- [41] Vojtěch Rödl and Mathias Schacht. Regularity lemmas for graphs. Preprint, available at http://www.math.uni-hamburg.de/home/schacht/preprints/reg_survey.pdf, 2010.
- [42] Dana Ron. Property testing. In Sanguthevar Rajasekaran, Panos M. Pardalos, John H. Reif, and José Rolim, editors, *Handbook of Randomized Computing*, volume II, chapter 15, pages 597–649. Kluwer Academic Publishers, 2001.
- [43] Dana Ron. Property testing: A learning theory perspective. *Found. Trends Mach. Learn.*, 1(3):307–402, 2008.
- [44] Dana Ron. Algorithmic and analysis techniques in property testing. *Found. Trends Theor. Comput. Sci.*, 5(2):73–205, 2009.
- [45] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996.
- [46] Saharon Shelah. Decidability of a portion of the predicate calculus. *Israel J. Math.*, 28(1-2):32–44, 1977.
- [47] Th. Skolem. Untersuchungen über die Axiome des Klassenkalküls und über Produktations und Summationsprobleme, welche gewisse Klassen von Aussagen betreffen. *Videnskapsselskapets skrifter, I. Matematisk-naturvidenskabelig klasse*, (3):37–71, 1919.
- [48] Th. Skolem. Logisch-kombinatorische Untersuchungen über die Erfüllbarkeit oder Beweisbarkeit mathematischer Sätze nebst einem Theorem über dichte Mengen. *Videnskapsselskapets skrifter, I. Matematisk-naturvidenskabelig klasse*, (4):1–26, 1920.
- [49] Terence Tao. A variant of the hypergraph removal lemma. *J. Comb. Theory Ser. A*, 113(7):1257–1280, 2006.
- [50] Anne Vedø. Asymptotic probabilities for second-order existential Kahr-Moore-Wang sentences. *J. Symbolic Logic*, 62(1):304–319, March 1997.