

Untestable Properties Expressible with Four First-Order Quantifiers^{*}

Charles Jordan^{**} and Thomas Zeugmann^{***}

Division of Computer Science
Hokkaido University, N-14, W-9, Sapporo 060-0814, Japan
{skip,thomas}@ist.hokudai.ac.jp

Abstract. In property testing, the goal is to distinguish between structures that have some desired property and those that are *far* from having the property, after examining only a small, random sample of the structure. We focus on the classification of first-order sentences based on their quantifier prefixes and vocabulary into testable and untestable classes. This classification was initiated by Alon *et al.* [1], who showed that graph properties expressible with quantifier patterns $\exists^*\forall^*$ are testable but that there is an untestable graph property expressible with quantifier pattern $\forall^*\exists^*$. In the present paper, their untestable example is simplified. In particular, it is shown that there is an untestable graph property expressible with each of the following quantifier patterns: $\forall\exists\forall\exists$, $\forall\exists\forall^2$, $\forall^2\exists\forall$ and $\forall^3\exists$.

Key words: property testing, logic

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 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 classes of first-order logic are testable and which are not.

Property testing was first introduced in the context of program verification (cf. Rubinfeld and Sudan [11] and Blum *et al.* [4]). The study of combinatorial property testing was initiated by Goldreich *et al.* [8], who focused on graphs. Alon *et al.* [1] first considered the classification problem for testability, although they restricted their attention to undirected, loop-free graphs. They showed that all such first-order sentences¹ with quantifier prefixes of the form $\exists^*\forall^*$ express testable properties. They also showed that there exists an untestable property

^{*} An earlier version with additional proofs is available as [9]. We would like to thank an anonymous referee for significant improvements to Theorem 2.

^{**} Supported by a Grant-in-Aid for JSPS Fellows under Grant No. 2100195209.

^{***} Supported by MEXT Grand-in-Aid for Scientific Research on Priority Areas under Grant No. 21013001.

¹ We assume throughout that all sentences are in prenex normal form.

expressible with the prefix $\forall^*\exists^*$. The example they give is (essentially) an encoding of graph isomorphism that can be expressed with a quantifier prefix of the form $\forall^{12}\exists^5$.

In studying the classification problem, it is necessary to determine the *minimum* number of quantifiers needed to express untestable properties. Additionally, the first-order theory of graphs is not restricted to undirected, loop-free graphs. Here, we show that there exists an untestable property of directed graphs that is expressible in first-order sentences with prefixes $\forall\exists\forall\exists$, $\forall\exists\forall^2$, $\forall^2\exists\forall$, and $\forall^3\exists$, when equality ($=$) is allowed (see Theorem 2 for a more formal statement). That is, with prefixes of these patterns and when equality is allowed, four first-order quantifiers suffice to express an untestable graph property. The proof is a modification of the proof by Alon *et al.* [1], which is made possible by the presence of directed edges and loops.

The results in Jordan and Zeugmann [10] show that one universal quantifier is not sufficient to express an untestable property (regardless of the vocabulary), and so it would be interesting to determine the status of the remaining prefixes with two universal quantifiers.

A class of first-order logic has the *finite model property* if every satisfiable formula in the class has a finite model. Classes without the finite model property contain *infinity axioms*, i.e. satisfiable formulas without finite models. The current classification for testability closely resembles the classification for the finite model property. It would be particularly interesting to determine the testability of the “minimal” classes with infinity axioms $[\forall^3\exists, (0, 1)]$ and $[\forall\exists\forall, (0, 1)]^2$.

2 Preliminaries

In property testing, the goal is always to distinguish structures that have some property from those that are far from having the property. We are particularly interested in properties that are first-order definable, and so we begin by defining our logic. Enderton [7] provides a more detailed introduction.

The atomic terms are the (countable) variable symbols x_i . There are no function or constant symbols, and so the terms are exactly the atomic terms. The atomic formulas are $E(x_i, x_j)$ and $x_i = x_j$, for any two variable symbols x_i and x_j . The formulas are built from the atomic formulas using the usual Boolean connectives (i.e., \wedge , \vee , \rightarrow , \leftrightarrow), negation (\neg) and first-order quantifiers (\forall , \exists) in the canonical way. The *well-formed* formulas or *sentences* are the formulas which contain no free variables. We have no further use for formulas that are not well-formed, and so we will refer to the well-formed formulas simply as *formulas*.

Our logic contains a special equality predicate symbol ($=$) which is always interpreted as true equality (i.e., $x_i = x_j$ is true iff the two symbols x_i and x_j refer to the same object). It also contains a single binary predicate symbol, which we have given the name E . Of course, the name of this symbol is not important; any fixed, unique name could have been chosen.

² These classes do not permit equality and so Theorem 2 does not immediately imply that the first is untestable.

A structure is an object that allows us to interpret a sentence in our logic. It consists of a finite universe U over which the variable symbols are allowed to range, and a binary relation E corresponding to the symbol E in our logic. Any such object can be considered as a (directed) graph, so from now on we refer to these structures as graphs. See Diestel [6] for an introduction to graph theory.

Definition 1. A graph $A = (U^A, E^A)$ is a pair consisting of a finite set of vertices U^A and a binary edge relation $E^A \subseteq U^A \times U^A$.

The natural numbers are denoted by $\mathbb{N} := \{0, 1, \dots\}$. For any set U we write $|U|$ to denote the cardinality of U and generally identify U with the set $\{0, \dots, |U| - 1\}$. We denote the set of graphs on exactly n vertices by \mathcal{G}^n and the set of all graphs by $\mathcal{G} := \cup_{n \in \mathbb{N}} \mathcal{G}^n$. The *size* of the universe of a graph $A = (U^A, E^A)$ is denoted by $\#(A)$ and defined as $\#(A) := |U^A|$.

A *property* P is any subset of \mathcal{G} . Sentences are interpreted in the usual way, and so we can decide $A \models \varphi$ for any fixed graph A and first-order sentence φ . Each sentence φ therefore defines a property, namely the set of its models,

$$P_\varphi := \{A \mid A \in \mathcal{G} \text{ and } A \models \varphi\}.$$

The properties that we use in the proof of Theorem 2 involve encodings of isomorphisms. Graphs $A = (U^A, E^A)$ and $B = (U^B, E^B)$ are *isomorphic* if there is a bijection $f : U^A \rightarrow U^B$ such that for all $(x, y) \in U^A \times U^A$, $(x, y) \in E^A$ iff $(f(x), f(y)) \in E^B$. We say that a property P is *closed under isomorphisms* if for all isomorphic $A, B \in \mathcal{G}$, it is true that $A \in P$ iff $B \in P$. All properties expressible in our logic are closed under isomorphisms.

The goal in property testing is to distinguish between structures that have properties and structures that are *far* from having them. This requires a distance measure, which we define next. In the following, \oplus denotes exclusive-or and E^A and E^B are the edge relations of A and B , respectively.

Definition 2. Let $n \in \mathbb{N}$ and let U be any universe such that $|U| = n$. Furthermore, let $A = (U, E^A)$ and $B = (U, E^B)$ be any two graphs with universe U . The distance between A and B is

$$\text{dist}(A, B) := \frac{|\{(x_1, x_2) \mid x_1, x_2 \in U \text{ and } E^A(x_1, x_2) \oplus E^B(x_1, x_2)\}|}{n^2}.$$

Note that by definition, $\#(A) = \#(B) = n$. The *dist* distance is the fraction of edges on which the two graphs disagree. This is the *dense graph* model introduced by Goldreich *et al.* [8] and is essentially based on the adjacency matrix representation. The *dist* distance generalizes to properties in the following way.

Definition 3. Let $P \subseteq \mathcal{G}$ be a property of graphs and let $A \in \mathcal{G}^n$ be a graph with n vertices. Then,

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

We are now able to define property testing itself. The following definitions are typical, but we will also mention several variations.

Definition 4. An ε -tester for property P is a randomized algorithm given an oracle which answers queries for the universe size and queries for the existence of edges connecting given nodes in a graph A . The tester must accept with probability at least $2/3$ if A has P and must reject with probability at least $2/3$ if $\text{dist}(A, P) \geq \varepsilon$.

Definition 5. A property P is testable if for every $\varepsilon > 0$ there is an ε -tester for P making a number of queries which is bounded from above by a function depending only on ε .

We allow different ε -testers for each $\varepsilon > 0$ and our definitions are therefore non-uniform. The uniform case is strictly more difficult (see, e.g., Alon and Shapira [3]). We are interested in proving untestability, and our results hold even in the non-uniform case. In *oblivious* testing (see Alon and Shapira [2]), the testers are not given access to the size of the universe. Again, our results hold in the more general case where the testers may make decisions based on the size of the universe. In a similar way, the number of loops in a graph is asymptotically insignificant compared to the number of possible non-loops. Modifying the definition of distance to account for this makes testing strictly more difficult (see Jordan and Zeugmann [10]) and so we use the more general definition above.

However, the (possible) loops seem to affect the notion of *indistinguishability* defined by Alon *et al.* [1]. We use the following modification of Definition 2.

Definition 6. Let $n \in \mathbb{N}$ and let U be any universe such that $|U| = n$. Furthermore, let $A = (U, E^A)$ and $B = (U, E^B)$ be any two graphs with universe U . For notational convenience, let

$$d_1(A, B) := \frac{|\{x \mid x \in U \text{ and } E^A(x, x) \oplus E^B(x, x)\}|}{n}, \text{ and}$$

$$d_2(A, B) := \frac{|\{(x_1, x_2) \mid x_1, x_2 \in U, x_1 \neq x_2, \text{ and } E^A(x_1, x_2) \oplus E^B(x_1, x_2)\}|}{n^2}.$$

The mr-distance between A and B is

$$\text{mrdist}(A, B) := \max \{d_1(A, B), d_2(A, B)\}.$$

Again, note that $\#(A) = \#(B) = n$. Although the number of loops is asymptotically insignificant, a tester can easily restrict its queries to the form (x, x) and distinguish between graphs that differ only in loops. Definition 6 is a special case of a definition from Jordan and Zeugmann [10], and *mrdist* abbreviates “maximum relational distance.” We use the following simple variation of indistinguishability for graphs that may contain loops.

Definition 7. Two properties P and Q of graphs are indistinguishable if they are closed under isomorphisms and for every $\varepsilon > 0$ there exists an N_ε such that for any graph A with universe of size $n \geq N_\varepsilon$, if A has P then $\text{mrdist}(A, Q) \leq \varepsilon$ and if A has Q then $\text{mrdist}(A, P) \leq \varepsilon$.

The important fact to note is that indistinguishability preserves testability. The proof of the following theorem is analogous to that given in Alon *et al.* [1].

Theorem 1. *If P and Q are indistinguishable, then P is testable if and only if Q is testable.*

Our classification definitions are from Börger *et al.* [5] except that we omit function symbols. We omit a detailed discussion, but the following is for completeness. Let Π be a string over the four-character alphabet $\{\exists, \exists^*, \forall, \forall^*\}$. Then $[\Pi, (0, 1)]_=$ is the set of sentences in prenex normal form which satisfy the following conditions:

1. The quantifier prefix is contained in the language specified by the regular expression Π .
2. There are zero (0) monadic predicate symbols.
3. In addition to the equality predicate ($=$), there is at most one (1) binary predicate symbol.
4. There are no other predicate symbols.

That is, $[\Pi, (0, 1)]_=$ is the set of sentences in the logic that we have defined above whose quantifier prefixes in prenex normal form match Π .

3 An Untestable Property

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 1 that P is indistinguishable from property P_f (cf. Definition 9) which is expressible in any of the prefix vocabulary classes mentioned in Theorem 2. 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 2. *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 8. *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 .

Definition 9. 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 pairwise distinct vertices x_1, x_2, x_3, x_4 , 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

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

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 1. *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 8. 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 9 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 9 because G satisfied Condition 5 of Definition 8 and we added only paired edges. Finally, G' satisfies Condition 4 of Definition 9 because the edges between loops and non-loops follow the isomorphism f from Condition 3 of Definition 8.

We have added exactly n (directed) edges, none of which are loops and so $\text{mrdist}(G, P) \leq \text{mrdist}(G, G') = 0 + n/n^2 < \varepsilon$, 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. Testability is preserved by indistinguishability (cf. Theorem 1) and thus showing that P is not testable suffices to prove that P_f is not testable (and therefore Theorem 2). The proof closely follows that of Alon *et al.* [1]. The crucial lemma is the following, a combination of Lemmata 7.3 and 7.4 from Alon *et al.* [1]. 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 [9] for an explicit proof of Lemma 2, which is technical and long.

Lemma 2 (Alon *et al.* [1]). *There exists a constant $\varepsilon' > 0$ such that for every $D \in \mathbb{N}$, there exist two undirected bipartite graphs $H = H(D)$ and $H' = H'(D)$ 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 Lemma 3 inductively.

Lemma 3. *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 3, 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 4. *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 2. 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 2, 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 5) 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 3, 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 5 below. \square

Recall that testing is easiest under the dist definition, and so Lemma 4 also implies P is not testable under other definitions.

Lemma 5. *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 2 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 2 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 2 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 2. Therefore,

$$\text{dist}(\alpha_1, \alpha_2) \geq \frac{16(\varepsilon'/4 - \varepsilon)t^2}{|\alpha_1|^2}. \quad (1)$$

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 (1). 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 (1) yields

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

The number of edges connecting α_1 to $H' \setminus \alpha_1$ is therefore, by (2), 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

4 Conclusion

Property testing is an application of induction (in the philosophy of science sense), in which we obtain a small, random sample of a structure and seek to determine whether the structure has a desired property or is far from having the property. We have considered the *classification problem* for testability, wherein we classify the prefix vocabulary classes of first-order logic according to their testability. In particular, we simplified the untestable property, expressible with quantifier prefix $\forall^{12}\exists^5$, from Alon *et al.* [1] for the case of directed graphs which may contain loops. This implies that there exists an untestable property expressible with quantifier prefixes $\forall\exists\forall\exists$, $\forall\exists\forall^2$, $\forall^2\exists\forall$ and $\forall^3\exists$.

In the classification problem for testability, it is necessary to determine the *minimal* untestable classes. Informally, we seek the untestable properties that are *easiest* to express. Jordan and Zeugmann [10] showed that classes with (at most) one universal quantifier are testable, and so there are at least two minimal untestable classes which have either two or three universal quantifiers.

The current classification for testability closely resembles the classification for the finite model property (see, e.g., Section 6.5 of Börger *et al.* [5]). The “minimal” classes without this property (i.e., those with infinity axioms) are $[\forall^3\exists, (0, 1)]$ and $[\forall\exists\forall, (0, 1)]$, while the case of $[\forall^2\exists, (0, 1)]_{=}$ is apparently open. It would be particularly interesting to determine the testability of these classes.

Acknowledgments We would like to thank an anonymous referee for significantly improving Theorem 2 by eliminating one variable from each prefix, adding $\forall\exists\forall\exists$, and also for simplifying the example following Definition 9.

References

- [1] N. Alon, E. Fischer, M. Krivelevich, and M. Szegedy. Efficient testing of large graphs. *Combinatorica*, 20(4):451–476, 2000.
- [2] N. Alon and A. Shapira. A characterization of the (natural) graph properties testable with one-sided error. *SIAM J. Comput.*, 37(6):1703–1727, 2008.
- [3] N. Alon and A. Shapira. A separation theorem in property testing. *Combinatorica*, 28(3):261–281, 2008.
- [4] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. *J. of Comput. Syst. Sci.*, 47(3):549–595, 1993.
- [5] E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem*. Springer-Verlag, 1997.
- [6] R. Diestel. *Graph Theory*. Springer, third edition, 2006.
- [7] H. B. Enderton. *A Mathematical Introduction to Logic*. Academic Press, second edition, 2000.
- [8] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998.
- [9] C. Jordan and T. 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.
- [10] C. Jordan and T. Zeugmann. Relational properties expressible with one universal quantifier are testable. In O. Watanabe and T. Zeugmann, editors, *Stochastic Algorithms: Foundations and Applications, 5th International Symposium, SAGA 2009, Sapporo, Japan, October 2009, Proceedings*, volume 5792 of *Lecture Notes in Computer Science*, pages 141 – 155. Springer, 2009.
- [11] R. Rubinfeld and M. Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996.