# Untestable Properties in the Kahr-Moore-Wang Class

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.** Property testing is a kind of randomized approximation in which one takes a small, random sample of a structure and wishes to determine whether the structure satisfies some property or is far from satisfying the property. We focus on the testability of classes of first-order expressible properties, and in particular, on the classification of prefix-vocabulary classes for testability. The main result is the untestability of $[\forall\exists\forall, (0, 1)]_{=}$. This is a well-known class and minimal for untestability. We discuss what is currently known about the classification for testability and briefly compare it to other classifications.
**Keywords:** property testing, logic, randomized algorithms

## 1 Introduction

In property testing, we take a random sample of some large structure and wish to distinguish between the case that it has some desired property and the case that it is far from having the property. We focus on the testability of first-order expressible properties, and in particular on the classification of prefix-vocabulary classes of first-order logic for testability.

Rubinfeld and Sudan [21] and Blum *et al.* [4] introduced the notion of property testing in the context of formal verification. The basic idea was soon extended by Goldreich *et al.* [11], who represented graphs as binary functions and focused on testing graph properties. We omit a detailed history of testing, see the recent introduction to testing graph properties by Goldreich [10], two recent surveys by Ron [19, 20], and older surveys by Fischer [7] and Ron [18].

We are particularly interested in the testability of properties expressible in subclasses of first-order logic, and review relevant work in Subsection 1.1.

Here, we show that there exist untestable graph properties expressible with quantifier prefix $\forall\exists\forall$ when equality is allowed (see Section 3 for a formal statement). Taking into account the related work described in Subsection 1.1 and using the notation of Definition 7, the current classification for testability is the following.

* Supported by a Grant-in-Aid for JSPS Fellows under Grant No. 2100195209.
** Supported by MEXT Grant-in-Aid for Scientific Research on Priority Areas under Grant No. 21013001.

- 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. $[\forall\exists\forall, (0,1)]_=$.

We are especially interested in determining the testability of variants of the Gödel class (i.e., classes whose prefix contains at least $\forall^2\exists$) as this would suffice to complete the classification for the special case of predicate logic with equality. The above classification is consistent with several other well-known classifications, such as that for the finite model property (see, e.g., Chapter 6 of Börger *et al.* [5], for docility (or finite satisfiability, see Kolaitis and Vardi [16]) and for associated 0-1 laws for fragments of existential second-order logic (see Kolaitis and Vardi [16]). It would be interesting to know if the classification for testability coincides with one of these classifications.

The rest of the paper is organized as follows. First, we review related work in Subsection 1.1. Definitions and notation are in Section 2. The main result is presented in Section 3.

### 1.1 Related Work

Alon *et al.* [3] proved that the regular languages are testable, implying that monadic first-order is testable given the well-known results of Büchi [6] or McNaughton and Papert [17]. Alon *et al.* [2] were the first to directly consider the classification problem for testability, restricted to properties of undirected, loop-free graphs. They proved the testability of all such properties expressible by prenex sentences with quantifier prefix $\exists^*\forall^*$, and also showed that there exists an untestable property expressible with quantifier prefix $\forall^*\exists^*$ (examining the proof shows that $\forall^{12}\exists^5$ suffices).

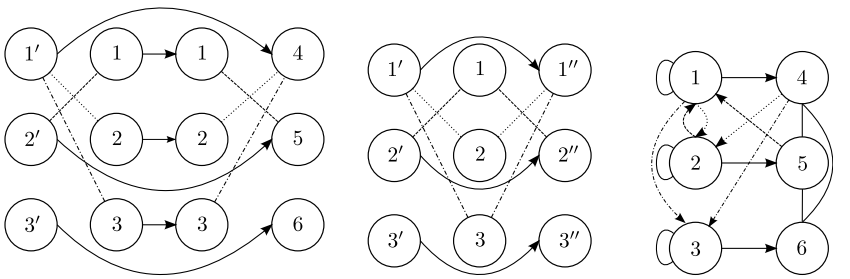
Both of these are (restrictions of) well-known classes. Jordan and Zeugmann extended [13] the positive result to the full Ramsey's class ($[\exists^*\forall^*, all]_=$), proved [12] the testability of Ackermann's class with equality ($[\exists^*\forall\exists^*, all]_=$), and sharpened [14] the negative result to prefixes $\forall^3\exists$, $\forall^2\exists\forall$, $\forall\exists\forall\exists$ and $\forall\exists\forall^2$ (on directed graphs with equality). This paper sharpens these last three prefixes and proves that $[\forall\exists\forall, (0,1)]_=$ is a minimal prefix class for untestability. This particular class is the restriction of the Kahr-Moore-Wang [15] class (plus equality) to directed graphs.

It is easy to show that $[\forall\exists\forall, (0,1)]$ (even without equality) has infinity axioms[1]. Vedø [22] 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).

The current paper sharpens some (prefixes $\forall^2\exists\forall$, $\forall\exists\forall\exists$, $\forall\exists\forall^2$) of the results of Jordan and Zeugmann [14] and so we briefly outline the improvement that allows us to minimize the prefix considered. The untestable property considered in [14]

[1] An infinity axiom is a sentence that has only infinite models.

is closely related to the untestable property of Alon *et al.* [2], but modified to minimize the number of quantifiers used. These properties are essentially first-order expressible versions of checking an explicitly given isomorphism between two graphs[2]. In fact, restricting the properties to checking an explicitly given isomorphism between undirected, bipartite graphs (see Figure 1(a)) maintains hardness for testing. However, graph isomorphism seems to require one to discuss at least four vertices simultaneously (because one wishes to state that an edge is present iff its image is present and the edges are disjoint in general).




(a) A graph with $P_b$ (b) A graph with $P_e$ (c) A graph with $P_f$

**Fig. 1.** Properties $P_b$, $P_e$ and $P_f$

Sharing one of the partitions (see Figure 1(b)) would seem to remove the need for four quantifiers. The resulting property is perhaps closer to a variant of *function* isomorphism, e.g., for functions $f, g\colon [n] \to \{0,1\}^n$ where the bit $i$ of $f(j)$ is 1 if there is an edge from $j$ in the leftmost partition to $i$ in the middle partition and likewise for $g(j)$ and the right partition. This property is not first-order expressible, but there is a somewhat tedious first-order encoding that is similar (see Figure 1(c) and the formula in Section 3).

The connection with function isomorphism allows us to leverage recent work on the testability of (Boolean) function isomorphism and use recent ideas and techniques from Alon and Blais [1] to prove Lemma 1.

## 2 Preliminaries

The goal in property testing is always to distinguish structures that have some property from those that are far from having the property. Here, we focus on first-order expressible properties of directed graphs and so we begin with the necessary definitions.

**Definition 1.** *A graph is an ordered pair $G = (V, E)$, where $V$ is a finite set and $E \subseteq V \times V$ a binary relation defined on $V$.*

We generally identify $V$ with the first $n$ naturals $[n] := \{1, \ldots, n\}$ and call $\#(G) := |V| = n$ the size of a graph $G$. Let $\mathcal{G}^n$ be the set of graphs of size $n$ and

[2] Graph isomorphism is generally hard for testing, see, e.g., Fischer and Matsliah [8].

$\mathcal{G} := \cup_{n \geq 0} \mathcal{G}^n$ be the set of all (finite) graphs. Note that our graphs are directed and may contain loops.

A property $P \subseteq \mathcal{G}$ of graphs is any set of graphs. We are particularly interested in first-order expressible properties. Our logic is a basic first-order predicate logic with equality. There are no function or constant symbols. We focus on first-order properties of graphs, and so the only predicate symbol (besides the special symbol $=$) is the binary edge symbol $E$.

A sentence $\varphi$ defines a property in the natural way,

$$P_\varphi := \{G \mid G \in \mathcal{G}, G \models \varphi\} .$$

We require a distance between graphs and properties, which we define in the following way. We denote the symmetric difference of sets by $\triangle$ and let $E^A$ and $E^B$ be the edge predicates of $A$ and $B$, respectively.

**Definition 2.** *Let $A = (V, E^A)$ and $B = (V, E^B)$ be two graphs defined such that $|V| = n$. The distance between $A$ and $B$ is*

$$\operatorname{dist}(A, B) := |E^A \triangle E^B|/n^2 .$$

The distance generalizes to properties in the obvious way, $\operatorname{dist}(A, P) := \min_{B \in P} \operatorname{dist}(A, B)$. Definition 2 results in a typical model of testing based on the dense graph model introduced by Goldreich *et al.* [11]. We now proceed to the remaining testing definitions.

**Definition 3.** *An* $\varepsilon$-tester *for property $P$ is a randomized algorithm that makes queries for the existence of edges 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 $\operatorname{dist}(A, P) \geq \varepsilon$.*

**Definition 4.** *Property $P$ is called* testable *if there is some function $c(\varepsilon)$ and for every $\varepsilon > 0$, an $\varepsilon$-tester for $P$ such that the tester makes at most $c(\varepsilon)$ queries.*

Note that the query complexity is bounded by a function that does not depend on the size of the graphs. We allow different $\varepsilon$-testers for each $\varepsilon > 0$ and so this is a non-uniform model. However, we are focused on proving untestability and our results hold even in the non-uniform case.

Next, we will define *indistinguishability*, a relation on properties introduced by Alon *et al.* [2] that preserves testability. However, testers can focus on *loops* and distinguish between structures that have an asymptotically small difference (because the number of loops is asymptotically dominated by the number of non-loops). We therefore begin with an alternative definition of distance (using this in place of Definition 2 makes testing (strictly) more difficult, but our result holds even when we use Definition 2). In the following, $\oplus$ denotes exclusive-or.

**Definition 5.** *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*

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

$$d_2(A,B) := \frac{|\{(x_1,x_2) \mid x_1,x_2 \in U,\ x_1 \neq x_2,\ \text{and } E^A(x_1,x_2) \oplus E^B(x_1,x_2)\}|}{n(n-1)} .$$

*The* mr-distance *between $A$ and $B$ is*

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

**Definition 6.** *Two properties $P$ and $Q$ of graphs are* indistinguishable *if they are closed under isomorphisms and for every $\varepsilon > 0$ there exists an $N_\varepsilon$ such that for any graph $A$ with universe of size $n \geq N_\varepsilon$, if $A$ has $P$ then* $\mathrm{mrdist}(A,Q) \leq \varepsilon$ *and if $A$ has $Q$ then* $\mathrm{mrdist}(A,P) \leq \varepsilon$.

An important property of indistinguishability is that it preserves testability. The proof of the following is analogous to that given in Alon *et al.* [2].

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

In fact, as the proof constructs an $\varepsilon$-tester for $P$ by iterating an $\varepsilon/2$-tester for $Q$ three times, one can also relate the query complexities of $P$ and $Q$. Many proofs of hardness for testability rely on Yao's Principle [23], an interpretation of von Neumann's minimax theorem for randomized computation. For completeness, we state the version that we use.

**Principle 1 (Yao's Principle)** *If there is an $\varepsilon \in (0,1)$ and a distribution over $\mathcal{G}^n$ such that all deterministic testers with complexity $c$ have an error-rate greater than $1/3$ for property $P$, then property $P$ is not testable with complexity $c$.*

The definition of "testable" is of course our usual one involving random testers. In general, one seeks to show that for sufficiently large $n$ and some increasing function $c := c(n)$, there is a distribution of inputs such that all deterministic testers with complexity $c$ have error-rates greater than $1/3$.

Finally, we briefly define the notation we use to specify prefix-vocabulary classes. See Börger *et al.* [5] for details and related material.

**Definition 7.** *Let $\Pi$ be a string over the four-character alphabet $\{\exists, \forall, \exists^*, \forall^*\}$. 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 regular language given by $\Pi$ (for technical reasons, one usually treats $\exists$ and $\forall$ as matching the relevant quantifier and also the empty string).*
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 prenex sentences in the logic defined above whose quantifier prefixes match $\Pi$. If the second component of the specification is *all*, then conditions two and three are removed (any number of predicate symbols with any arities are acceptable).

## 3 An Untestable Property

Our goal in this section is Theorem 2.

**Theorem 2.** *The prefix class* $[\forall\exists\forall, (0,1)]_=$ *is not testable.*

We begin by outlining the proof. First, we define $P_f$, a property expressible in the class $[\forall\exists\forall, (0,1)]_=$ which, as described in Subsection 1.1, is in some sense a somewhat tedious but first-order expressible variant of checking (explicit) isomorphism of undirected bipartite graphs in tripartite graphs. We then define a variant $P_2$, in which the isomorphism is not explicitly given and we must test whether there exists some suitable isomorphism. Although this increases the complexity of deciding the problem from checking an isomorphism to finding one, it does not change hardness for testing. We show that $P_2$ and $P_f$ are indistinguishable and so $P_2$ is testable iff $P_f$ is testable. Finally, we prove directly that $P_2$ is untestable, even with $o(\sqrt{n})$ queries, using an argument based on a recent proof by Alon and Blais [1].

*Proof (Theorem 2).* We begin by defining $P_f$. Formally, it is the set of graphs satisfying the following conjunction of four clauses (see Figure 1(c) for an example).

$$\begin{aligned}
\forall x\exists y\forall z: \quad & \{\,((\neg E(x,x) \wedge \neg E(z,z) \wedge x \neq z) \rightarrow E(x,z)) \\
\wedge \quad & (E(x,x) \rightarrow (E(x,y) \wedge \neg E(y,y) \wedge [(\neg E(z,z) \wedge E(x,z)) \rightarrow y = z])) \\
\wedge \quad & (\neg E(x,x) \rightarrow (E(y,x) \wedge E(y,y) \wedge [(E(z,z) \wedge E(z,x)) \rightarrow y = z])) \\
\wedge \quad & ((E(x,x) \wedge E(z,z)) \rightarrow [\neg E(y,y) \wedge E(x,y) \wedge (E(x,z) \leftrightarrow E(y,z))])\,\}
\end{aligned}$$

A graph satisfies this formula if the following conditions are all satisfied.

1. The nodes without loops form a complete subgraph.
2. For every node $x$ with a loop, there is exactly one $y$ without a loop such that there is an edge from $x$ to $y$.
3. For every node $y$ without a loop, there is exactly one $x$ with a loop such that there is an edge from $x$ to $y$.
4. For all nodes $x, z$ with loops, and $y$ the unique node without a loop such that $E(x,y)$, it holds that $E(x,z)$ iff $E(y,z)$.

Property $P_2$ below is similar to $P_f$, except that the isomorphism is not explicitly given.

**Definition 8.** *A graph* $G = (V,E)$ *has* $P_2$ *if it satisfies the following conditions.*

1. *There is a partition*[3] $V_1, V_2 \subseteq V$ *such that* $|V_1| = |V_2|$*, there are loops* $(E(x,x))$ *on all* $x \in V_1$ *and no loops* $(\neg E(x,x))$ *for all* $x \in V_2$*.*
2. *The nodes without loops form a complete subgraph.*
3. *There are no edges from a node with a loop to a node without a loop.*

[3] $V_1, V_2$ partition $V$ if $V_1 \cap V_2 = \emptyset$ and $V_1 \cup V_2 = V$.

4. *There exists a bijection* $b\colon V_1 \to V_2$ *such that if* $x, z$ *have loops, then* $E(x,z)$ *iff* $E(b(x),z)$.

It is not difficult to show that properties $P_f$ and $P_2$ are indistinguishable.

**Claim 1** *Properties* $P_f$ *and* $P_2$ *are indistinguishable.*

*Proof (Claim 1).* Let $\varepsilon > 0$ be arbitrary and let $N_\varepsilon = \varepsilon^{-1}$. Assume that $G$ has property $P_2$ and that $\#(G) > N_\varepsilon$. We will show that $\mathrm{mrdist}(G, P_f) < \varepsilon$.

Graph $G$ has $P_2$ and so there is a bijection satisfying Condition 4 of Definition 8. We therefore add the edges $E(i, b(i))$ making the isomorphism (from $V_1$ to $V_2$) explicit. The resulting graph $G_f$ has $P_f$.

We have made exactly $n/2$ modifications, all to non-loops, and $n - 1 \geq N_\varepsilon$, so $\mathrm{mrdist}(G, P_f) \leq \mathrm{mrdist}(G, G_f) = 1/2(n-1) < \varepsilon$.

The converse is analogous; given a $G$ that has $P_f$, simply remove the $n/2$ edges from loops to non-loops after using them to construct a suitable bijection $b$.
□ Claim 1

Properties $P_f$ and $P_2$ are indistinguishable and so (by Theorem 1), it suffices to show that $P_2$ is is untestable. Lemma 1 below is stronger than necessary, and actually implies a $\Omega(\sqrt{n})$ lower bound for testing $P_f$ per the discussion following Theorem 1.
□ Theorem 2

**Lemma 1.** *Fix* $0 < \varepsilon < 1/2$. *Any* $\varepsilon$*-tester for* $P_2$ *must perform* $\Omega(\sqrt{n})$ *queries.*

*Proof (Lemma 1).* The proof is via Yao's Principle (cf. Principle 1), and so we define two distributions, $D_{\mathrm{no}}$ and $D_{\mathrm{yes}}$ and show that all deterministic testers have an error-rate greater than $1/3$ for property $P_2$ when the input is chosen randomly from $D_{\mathrm{no}}$ with probability $1/2$ and from $D_{\mathrm{yes}}$ with probability $1/2$.

In the following, we consider a distribution over graphs of sufficiently large size $2n$, and an arbitrary fixed partition of the vertices into $V_1$ and $V_2$ such that $|V_1| = |V_2| = n$ (for example, let the vertices be the integers $V := [2n]$, $V_1 := [n]$ and $V_2 := V \backslash V_1$).

We begin with $D_{\mathrm{no}}$, defined as the following distribution.

1. Place a loop on each vertex in $V_1$ and place no loops in $V_2$.
2. Place each possible edge (except loops) in $V_1 \times V_1$ and $V_2 \times V_1$ uniformly and independently with probability $1/2$.

That is, $D_{\mathrm{no}}$ is the uniform distribution of graphs (with this particular partition) satisfying the first three conditions of $P_2$.

Next, we define $D_{\mathrm{yes}}$ as the following.

1. Choose uniformly a random bijection $\pi\colon V_1 \to V_2$.
2. Place a loop on each vertex in $V_1$ and place no loops in $V_2$.
3. For each possible edge $(i, j \neq i) \in V_1 \times V_1$, uniformly and independently place *both* $(i,j)$ and $(\pi(i), j)$ with probability $1/2$ (otherwise place *neither*).

It is easy to see that $D_{\mathrm{yes}}$ generates only positive instances. Next, we show that $D_{\mathrm{no}}$ generates negative instances with high probability.

**Lemma 2.** *Fix $0 < \varepsilon < 1/2$ and let $n$ be sufficiently large. Then,*

$$\Pr_{G \sim D_{no}} [\mathrm{dist}(G, P_2) \leq \varepsilon] = o(1)\,.$$

*Proof (Lemma 2).* $D_{\mathrm{no}}$ is the uniform distribution over graphs of size $2n$ with a particular partition satisfying the first three conditions of $P_2$. Let $G_\varepsilon$ be the set of graphs $G'$ of size $2n$ satisfying these conditions and such that $\mathrm{dist}(G', P_2) \leq \varepsilon$ (regardless of partition).

Counting the number of such graphs shows

$$|G_\varepsilon| \;\leq\; \binom{2n}{n} 2^{n(n-1)} n! \sum_{i=0}^{\lceil \varepsilon 2n^2 \rceil} \binom{2n^2}{i} \;\leq\; \binom{2n}{n} 2^{n(n-1)} n! 2^{H(\varepsilon)2n^2}\,,$$

where $H(\varepsilon) := -\varepsilon \log \varepsilon - (1-\varepsilon)\log(1-\varepsilon)$ is the binary entropy function (cf. Lemma 16.19 in Flum and Grohe [9] for the bound on the summation).

Distribution $D_{\mathrm{no}}$ produces each of $2^{n(n-1)}2^{n^2}$ graphs with equal probability, and so

$$\begin{aligned} \Pr_{G \sim D_{\mathrm{no}}} [\mathrm{dist}(G, P_2) \leq \varepsilon] \;&\leq\; \frac{|G_\varepsilon|}{2^{n(n-1)+n^2}} \;\leq\; \binom{2n}{n} n! 2^{H(\varepsilon)2n^2} / 2^{n^2} \\ &\approx\; \frac{4^n n! 2^{H(\varepsilon)2n^2}}{\sqrt{\pi n} 2^{n^2}} \;=\; o(1)\,. \end{aligned}$$

The approximation is asymptotically tight, which suffices. □ Lemma 2

We have shown that $D_{\mathrm{yes}}$ generates only positive instances and that (with high probability) $D_{\mathrm{no}}$ generates instances that are $\varepsilon$-far from $P_2$. Next, we show that (again, with high probability) the two distributions look the same to testers making only $o(\sqrt{n})$ queries.

The proof is similar to a proof by Alon and Blais [1]. We begin by defining two random processes, $P_{\mathrm{no}}$ and $P_{\mathrm{yes}}$, which answer queries from testers and generate instances according to $D_{\mathrm{no}}$ and $D_{\mathrm{yes}}$, respectively.

Process $P_{\mathrm{no}}$ is defined in the following way.

1. Choose uniformly a random bijection $\pi\colon V_1 \to V_2$.
2. Intercept all queries from the tester and respond as follows.
   (a) To queries $E(i,i)$ with $i \in V_1$: respond 1.
   (b) To queries $E(i,i)$ with $i \in V_2$: respond 0.
   (c) To queries $E(i,j)$ with $i \in V_1$ and $j \in V_2$: respond 0.
   (d) To queries $E(i,j)$ with $i \neq j \in V_1$: quit if we have queried $E(\pi(i), j)$, otherwise respond 1 or 0 randomly with probability 1/2 in each case.
   (e) To queries $E(i,j)$ with $i \in V_2$ and $j \in V_1$: quit if we have queried $E(\pi^{-1}(i), j)$, otherwise respond 1 or 0 randomly with probability 1/2 in each case.
3. When the process has quit or the tester has finished its queries, complete the generated instance in the following way. First, fix the edges that were queried according to our answer. Next, place loops on each vertex in $V_1$, no loops in $V_2$ and no edges from $V_1$ to $V_2$. Place each remaining possible edge, place it (uniformly, independently) with probability 1/2, *ignoring* $\pi$.

We define $P_{\text{yes}}$ in the same way, except for the final step. When $P_{\text{yes}}$ quits or the tester finishes, it fixes the edges that were queried according to its answers, and *also* fixes the corresponding edges (when relevant) according to $\pi$. More precisely, for each fixed $E(i,j)$ with $i \neq j \in V_1$, we also fix $E(\pi(i),j)$ and for fixed $E(i,j)$ with $i \in V_2, j \in V_1$, we also fix $E(\pi^{-1}(i),j)$, in both cases the same as our response to $E(i,j)$ (not randomly). The remaining edges are placed as in $P_{\text{no}}$.

Note that $P_{\text{no}}$ generates instances according to $D_{\text{no}}$ and $P_{\text{yes}}$ generates instances according to $D_{\text{yes}}$. In addition, $P_{\text{yes}}$ and $P_{\text{no}}$ behave identically until they quit or answer all queries. In particular, if a tester does not cause the process to quit, the distribution of responses of its queries is identical for the two processes. We show that, with high probability, a tester that makes $o(\sqrt{n})$ queries does not cause either process to quit.

**Lemma 3.** *Let $T$ be a deterministic tester which makes $o(\sqrt{n})$ queries, and let $T$ interact with $P_{yes}$ or $P_{no}$. In both cases,*

$$\Pr[T \text{ causes the process to quit}] = o(1)\,.$$

*Proof (Lemma 3).* The condition causing the process to quit is identical in $P_{\text{yes}}$ and $P_{\text{no}}$. The probability that any pair of queries $E(i,j)$ and $E(i',j')$ cause the process to quit is at most

$$\Pr[i' = \pi(i) \text{ or } i = \pi(i')] \leq \frac{(n-1)!}{n!} = 1/n\,.$$

The tester makes at most $o(\sqrt{n})$ queries and so

$$\Pr[T \text{ causes the process to quit}] \leq o(\sqrt{n})^2 O(1/n) = o(1)\,.$$

□ Lemma 3

Any deterministic tester $T$ which makes $o(\sqrt{n})$ queries can only distinguish between $D_{\text{yes}}$ and $D_{\text{no}}$ with probability $o(1)$, but it must accept $D_{\text{yes}}$ with probability $2/3$, and reject $D_{\text{no}}$ with probability $2/3 - o(1)$. It is impossible for $T$ to satisfy both conditions, and the lemma follows from Principle 1. □ Lemma 1

## 4 Conclusion

Property testing is a kind of randomized approximation, where we take 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 focused on the classification problem for testability, wherein we seek to determine exactly which prefix vocabulary classes are testable and which are not. The main result of this paper is the untestability of $[\forall\exists\forall, (0,1)]_{=}$, a sharpening of the results of [14]. This class is a minimal class for untestability.

As mentioned in Subsection 1.1, the current classification for testability closely resembles several other classifications (e.g., those for the finite model

property, docility and associated second-order 0-1 laws) and it would be interesting to determine whether it coincides with one of these. In particular, determining the testability of variants of the Gödel class would complete the classification for the special case of predicate logic with equality.

## References


[1] Alon, N., Blais, E.: Testing Boolean function isomorphism. In: Approximation, Randomization, and Combinatorial Optimization, Algorithms and Techniques, 13th International Workshop, APPROX 2010, and 14th International Workshop, RANDOM 2010, Barcelona, Spain, September 2010, Proceedings. Volume 6302 of Lecture Notes in Computer Science., Springer (2010) 394–405

[2] Alon, N., Fischer, E., Krivelevich, M., Szegedy, M.: Efficient testing of large graphs. Combinatorica **20**(4) (2000) 451–476

[3] Alon, N., Krivelevich, M., Newman, I., Szegedy, M.: Regular languages are testable with a constant number of queries. SIAM J. Comput. **30**(6) (2001) 1842–1862

[4] Blum, M., Luby, M., Rubinfeld, R.: Self-testing/correcting with applications to numerical problems. J. of Comput. Syst. Sci. **47**(3) (1993) 549–595

[5] Börger, E., Grädel, E., Gurevich, Y.: The Classical Decision Problem. Springer-Verlag (1997)

[6] Büchi, J.R.: Weak second-order arithmetic and finite-automata. Z. Math. Logik Grundlagen Math. **6** (1960) 66–92

[7] Fischer, E.: The art of uninformed decisions. Bulletin of the European Association for Theoretical Computer Science **75** (October 2001) 97–126 Columns: Computational Complexity.

[8] Fischer, E., Matsliah, A.: Testing graph isomorphism. SIAM J. Comput. **38**(1) (2008) 207–225

[9] Flum, J., Grohe, M.: Parametrized Complexity Theory. Springer (2006)

[10] Goldreich, O.: Introduction to testing graph properties. Technical Report TR10-082, Electronic Colloquium on Computational Complexity (ECCC) (May 2010)

[11] Goldreich, O., Goldwasser, S., Ron, D.: Property testing and its connection to learning and approximation. J. ACM **45**(4) (1998) 653–750

[12] Jordan, C., Zeugmann, T.: Relational properties expressible with one universal quantifier are testable. In Watanabe, O., Zeugmann, T., eds.: Stochastic Algorithms: Foundations and Applications, 5th International Symposium, SAGA 2009, Sapporo, Japan, October 2009, Proceedings. Volume 5792 of Lecture Notes in Computer Science., Springer (2009) 141–155

[13] Jordan, C., Zeugmann, T.: A note on the testability of Ramsey's class. In Kratochvíl, J., Li, A., Fiala, J., Kolman, P., eds.: Theory and Applications of Models of Computation, 7th International Conference, TAMC 2010, Prague, Czech Republic, June 2010, Proceedings. Volume 6108 of Lecture Notes in Computer Science., Springer (2010) 296–307

[14] Jordan, C., Zeugmann, T.: Untestable properties expressible with four first-order quantifiers. In Dediu, A.H., Fernau, H., Martín-Vide, C., eds.: Language and Automata Theory and Applications, 4th International Conference, LATA 2010, Trier, Germany, May 2010, Proceedings. Volume 6031 of Lecture Notes in Computer Science., Springer (2010) 333–343

[15] Kahr, A.S., Moore, E.F., Wang, H.: Entscheidungsproblem reduced to the $\forall\exists\forall$ case. Proc. Nat. Acad. Sci. U.S.A. **48** (1962) 365–377

[16] Kolaitis, P.G., Vardi, M.Y.: 0-1 laws for fragments of existential second-order logic: A survey. In Nielsen, M., Rovan, B., eds.: 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., Springer (2000) 84–98
[17] McNaughton, R., Papert, S.: Counter-Free Automata. M.I.T. Press (1971)
[18] Ron, D.: Property testing. In Rajasekaran, S., Pardalos, P.M., Reif, J.H., Rolim, J., eds.: Handbook of Randomized Computing. Volume II. Kluwer Academic Publishers (2001) 597–649
[19] Ron, D.: Property testing: A learning theory perspective. Found. Trends Mach. Learn. **1**(3) (2008) 307–402
[20] Ron, D.: Algorithmic and analysis techniques in property testing. Found. Trends Theor. Comput. Sci. **5**(2) (2009) 73–205
[21] Rubinfeld, R., Sudan, M.: Robust characterizations of polynomials with applications to program testing. SIAM J. Comput. **25**(2) (1996) 252–271
[22] Vedø, A.: Asymptotic probabilities for second-order existential Kahr-Moore-Wang sentences. J. Symbolic Logic **62**(1) (March 1997) 304–319
[23] Yao, A.C.C.: Probabilistic computations: Toward a unified measure of complexity. In: 18th Annual Symposium on Foundations of Computer Science, IEEE Computer Society (1977) 222–227