Testing Boolean Function Isomorphism

Noga Alon[1,⋆] and Eric Blais[2]

[1] Schools of Mathematics and Computer Science, Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv 69978, Israel. Email: nogaa@tau.ac.il.
[2] School of Computer Science, Carnegie Mellon University, Pittsburgh 15213, USA. Email: eblais@cs.cmu.edu.

Abstract. Two boolean functions $f, g : \{0,1\}^n \to \{0,1\}$ are *isomorphic* if they are identical up to relabeling of the input variables. We consider the problem of *testing* whether two functions are isomorphic or far from being isomorphic with as few queries as possible.
In the setting where one of the functions is known in advance, we show that the non-adaptive query complexity of the isomorphism testing problem is $\tilde{\Theta}(n)$. In fact, we show that the lower bound of $\Omega(n)$ queries for testing isomorphism to g holds for almost all functions g.
In the setting where both functions are unknown to the testing algorithm, we show that the query complexity of the isomorphism testing problem is $\tilde{\Theta}(2^{n/2})$. The bound in this result holds for both adaptive and non-adaptive testing algorithms.

1 Introduction

The field of property testing, originally introduced by Rubinfeld and Sudan [20], considers the following general problem: given a property $\mathcal{P}$, determine the minimum number q of queries required to determine with high probability whether an input has the property $\mathcal{P}$ or whether it is "far" from $\mathcal{P}$. The field has been extremely active over the last few years – see, e.g., the recent surveys [18, 19].

In this paper, we concern ourselves with property testing of boolean functions. Despite significant progress in the study of the query complexity of many properties of boolean functions (e.g., monotonicity [7, 11, 13], juntas [10, 5], having concise representations [6], halfspaces [16, 17]), our overall understanding of the testability of boolean function properties still lags behind our understanding of the testability of graph properties, whose study was initiated by Goldreich, Goldwasser, and Ron [14].

A notable example that illustrates the gap between our understanding of graph and boolean function properties is *isomorphism.* Two graphs are isomorphic if they are identical up to relabeling of the vertices, while two boolean functions are isomorphic if they are identical up to relabeling of the input variables. There are three main variants to the isomorphism testing problem. (In the following list, an "object" refers to either a graph or a boolean function.)

⋆ Research supported in part by an ERC Advanced grant, by a USA-Israeli BSF grant and by the Hermann Minkowski Minerva Center for Geometry at Tel Aviv University.

1. **Testing isomorphism to a given object $\mathcal{O}$.** The query complexity required to test isomorphism in this variant depends on the object $\mathcal{O}$; the goal for this problem is to characterize the query complexity for *every* graph or boolean function.
2. **Testing isomorphism to the hardest known object.** A less fine-grained variant of the first problem asks to determine the maximum query complexity of testing isomorphism to $\mathcal{O}$ over objects of a given size.
3. **Testing isomorphism of two unknown objects.** In this variant, the testing algorithm has query access to two unknown objects $\mathcal{O}_1$ and $\mathcal{O}_2$ and must distinguish between the cases where they are isomorphic to each other or far from isomorphic to each other.

The problem of testing graph isomorphism was first raised by Alon, Fischer, Krivelevich, and Szegedy [1] (see also [8]), who used a lower bound on testing isomorphism of two unknown graphs to give an example of a non-testable first-order graph property of a certain type. Fischer [9] studied the problem of testing isomorphism to a given graph G and characterized the query complexity of the problem in terms of a complexity measure of G. Tight asymptotic bounds on the query complexity of the problem of testing isomorphism to a known graph and testing isomorphism of two unknown graphs were then obtained by Fischer and Matsliah [12]. As a result, all three versions of the graph isomorphism testing problem are well understood.

The picture is much less complete in the setting of boolean functions. Testing isomorphism against a known function f was first studied by Fischer, Kindler, Ron, Safra, and Samorodnitsky [10]. They gave a general upper bound on the problem showing that for every function f that depends on k variables (that is, for every k-junta), the problem of testing isomorphism to f requires $\text{poly}(k/\epsilon)$ queries. Conversely, they showed that when f is a parity function on $k < o(\sqrt{n})$ variables, testing isomorphism to f requires $\tilde{\Omega}(k)$ queries. No other progress was made on the problem of testing isomorphism on boolean functions until very recently, when Blais and O'Donnell [3] showed that for every function f that "strongly" depends on k variables, testing isomorphism to f requires $\Omega(\log k)$ queries. Taken together, the results in [10, 3] give only an incomplete solution to the problem of testing isomorphism to a given boolean function and provide only weak bounds on the other two versions of the isomorphism testing problem.

Our results. We introduce new results for all three variants of the problem of testing isomorphism to boolean functions.

In the problem of testing isomorphism to a given function $g : \{0,1\}^n \to \{0,1\}$, it is easy to show that $O(\frac{n \log n}{\epsilon})$ queries always suffice to ϵ-test isomorphism to any function g. (For completeness, we give the proof of this statement in Section 3.1.) Our main result is a matching lower bound (up to a logarithmic factor) that applies for *almost all* functions g.

Theorem 1.1. *Fix $0 < \epsilon < \frac{1}{2}$. For a $1 - o(1)$ fraction of the functions $g : \{0,1\}^n \to \{0,1\}$, any non-adaptive algorithm for ϵ-testing isomorphism to g must make at least $\frac{n}{100}$ queries.*

We present the proof of Theorem 1.1 in Sections 3.2 and 3.3. The lower bound of the theorem and the aforementioned upper bound immediately give a tight bound on the query complexity of testing isomorphism to a known function:

Corollary 1.2. *The maximum possible query complexity for testing isomorphism to a known function $\{0,1\}^n \to \{0,1\}$ non-adaptively is $\tilde{\Theta}(n)$. This bound holds for testing algorithms with 1-sided and 2-sided error.*

Finally, we examine the problem of testing two unknown functions for the property of being isomorphic. A simple algorithm can ϵ-test isomorphism in this setting with $\tilde{O}(2^{n/2}/\sqrt{\epsilon})$ queries. We give a matching lower bound establishing that no other algorithm can do better.

Theorem 1.3. *The query complexity for testing isomorphism of two unknown functions in $\{0,1\}^n \to \{0,1\}$ is $\tilde{\Theta}(2^{n/2})$. This bound holds for all testing algorithms (adaptive or non-adaptive, with 1-sided or 2-sided error).*

We present the proof of Theorem 1.3 in Section 4.

Related work. Recently, Chakraborty, García-Soriano, and Matsliah [4] independently obtained results very similar to Corollary 1.2 and Theorem 1.3. In fact, their version of Corollary 1.2 contains a stronger lower bound that also applies to adaptive testing algorithms.

Furthermore, [4] also show tight bounds on the query complexity for testing isomorphism to the hardest known function within some *restricted* classes of functions. Notably, they show that $O(k \log k)$ queries are sufficient to test isomorphism to any k-juntas and that $\Omega(k)$ queries are required to test isomorphism to some k-juntas.

2 Preliminaries and Notation

Throughout the paper, f and g represent boolean functions $\{0,1\}^n \to \{0,1\}$. The *weight* of an input $x = (x_1, \ldots, x_n) \in \{0,1\}^n$ is $|x| = x_1 + \cdots + x_n$. All big O notation in this paper refers to asymptotic statements as $n \to \infty$ while the other parameters (typically, ϵ) remain constant. Tilde notation is used to hide polylogarithmic factors – for example $f = \tilde{\Theta}(n)$ if there is a positive constant c such that $f \geq \Omega(\frac{n}{\log^c n})$ and $f \leq O(n \log^c n)$.

For a permutation $\pi : [n] \to [n]$ and $x = (x_1, \ldots, x_n) \in \{0,1\}^n$, we write $\pi(x) = (x_{\pi(1)}, \ldots, x_{\pi(n)})$. The function $g_\pi : \{0,1\}^n \to \{0,1\}$ represents the function defined by $g_\pi(x) = g(\pi(x))$ for every $x \in \{0,1\}^n$. Two functions f and g are *isomorphic* if there is a permutation π such that $f = g_\pi$.

Given a set $X \subseteq \{0,1\}^n$ and a permutation π on $[n]$, we write $\pi(X) = \{\pi(x) : x \in X\}$. With some abuse of notation, we also write $f(X) \in \{0,1\}^{|X|}$ to represent the value of f over each $x \in X$, over some ordering of X. In particular, $f(X) = g(X)$ iff $f(x) = g(x)$ for every $x \in X$.

Given two random variables A, B defined on a common discrete sample space Ω, the *total variation* distance between A and B is

$$d_{TV}(A, B) = \frac{1}{2} \sum_{\omega \in \Omega} \left| \Pr[A = \omega] - \Pr[B = \omega] \right|.$$

A *property* $\mathcal{P}$ of boolean functions $\{0,1\}^n \to \{0,1\}$ is simply a subset of those functions. The distance of a function f to $\mathcal{P}$ is the minimum distance between f and g over all $g \in \mathcal{P}$, where the distance between two functions is $\mathrm{dist}(f, g) = \Pr_x[f(x) \neq g(x)] = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} \mathbf{1}[f(x) \neq g(x)]$.

A (q, ϵ)*-tester* for the property $\mathcal{P}$ is a randomized algorithm $\mathcal{T}$ that queries an unknown function f on q different inputs in $\{0,1\}^n$ and then (1) accepts f with probability at least $\frac{2}{3}$ when $f \in \mathcal{P}$, and (2) rejects f with probability at least $\frac{2}{3}$ when f is ϵ-far from $\mathcal{P}$. (If the property deals with a pair of input functions, the algorithm may query both.)

When a tester $\mathcal{T}$ chooses all its queries in advance, it is *non-adaptive*; if it uses the responses to some of its queries to decide what queries to make afterwards, it is *adaptive*. A tester that accepts functions in $\mathcal{P}$ with probability 1 (instead of $\frac{2}{3}$) has *1-sided error*, otherwise it has *2-sided error*.

The *query complexity* of a property $\mathcal{P}$ for a given $\epsilon > 0$ is the minimum value of q for which there is a (q, ϵ)-tester for $\mathcal{P}$.

3 Testing Isomorphism to a Given Function

3.1 Upper Bound

The trivial algorithm $\mathcal{T}$ for testing isomorphism to g queries the unknown function $f : \{0,1\}^n \to \{0,1\}$ on a set $Q \subseteq \{0,1\}^n$ of $\frac{n \ln n}{\epsilon}$ randomly selected inputs. The algorithm accepts f if and only if there is a permutation $\pi \in \mathcal{S}_n$ such that $f(x) = g(\pi(x))$ for every $x \in Q$.

Clearly, the trivial algorithm $\mathcal{T}$ is non-adaptive and accepts functions isomorphic to g with probability 1. The following simple proposition completes the proof of correctness of $\mathcal{T}$ by showing that it rejects functions ϵ-far from isomorphic to g with probability at least $\frac{2}{3}$.

Proposition 3.1. *Fix $\epsilon > 0$. Let $g : \{0,1\}^n \to \{0,1\}$ be any boolean function and $f : \{0,1\}^n \to \{0,1\}$ be a function ϵ-far from isomorphic to g. Then $\mathcal{T}$ accepts f with probability $o(1)$.*

Proof. For any permutation $\pi \in \mathcal{S}_n$, there are at least $\epsilon 2^n$ values of $x \in \{0,1\}^n$ for which $f(x) \neq g(\pi(x))$. The probability that none of those inputs are queried by $\mathcal{T}$ is at most $(1 - \epsilon)^{|Q|} \leq e^{-\epsilon(n \ln n/\epsilon)} = n^{-n}$. Thus, by the union bound, the probability that there is a permutation $\pi \in \mathcal{S}_n$ such that $f(x) = g(\pi(x))$ for every $x \in Q$ is at most $n!/n^n = o(1)$. $\square$

3.2 Lower Bound

We prove Theorem 1.1 in this section. The proof of this theorem combined with the upper bound of the previous section immediately yields Corollary 1.2.

The proof of Theorem 1.1 uses Yao's Minimax Principle [21]. For a fixed function g we introduce two distributions $\mathcal{F}_{\text{yes}}$ and $\mathcal{F}_{\text{no}}$ such that a function $f \sim \mathcal{F}_{\text{yes}}$ is isomorphic to g and a function $f \sim \mathcal{F}_{\text{no}}$ is ϵ-far from isomorphic to g with high probability. We then show that for most choices of g, deterministic non-adaptive testing algorithms cannot distinguish functions drawn from either of these distributions with only $\frac{n}{100}$ queries.

We define $\mathcal{F}_{\text{yes}}$ to be the uniform distribution over functions isomorphic to g. In other words, we draw a function $f \sim \mathcal{F}_{\text{yes}}$ by choosing $\pi \in \mathcal{S}_n$ uniformly at random and setting $f = g_\pi$.

A first idea for $\mathcal{F}_{\text{no}}$ may be to make it the uniform distribution over all boolean functions $\{0,1\}^n \to \{0,1\}$. This idea does not quite work, since, for example, a random function differs from g and all functions isomorphic to it on the all 0 input or the all 1 input with probability at least 3/4. However, a simple modification of this idea does work: to draw a function $f \sim \mathcal{F}_{\text{no}}$, we choose a permutation $\pi \in \mathcal{S}_n$ uniformly at random and we choose a function f_{rand} uniformly at random from all boolean functions on n variables. We then let f be the function defined by

$$f(x) = \begin{cases} f_{\text{rand}}(x) & \text{if } \frac{n}{3} \le |x| \le \frac{2n}{3}, \\ g_\pi(x) & \text{otherwise.} \end{cases}$$

With high probability, a function $f \sim \mathcal{F}_{\text{no}}$ is far from isomorphic to g.

Proposition 3.2. *Fix $0 < \epsilon < \frac{1}{2}$. For any function $g : \{0,1\}^n \to \{0,1\}$, the function $f \sim \mathcal{F}_{\text{no}}$ is ϵ-close to isomorphic to g with probability at most $o(1)$.*

Proof. Fix any permutation $\pi \in \mathcal{S}_n$. Let f_{rand} be the random function generated in the draw of $f \sim \mathcal{F}_{\text{no}}$. By the triangle inequality,

$$\text{dist}(f, g_\pi) \ge \text{dist}(f_{\text{rand}}, g_\pi) - \text{dist}(f, f_{\text{rand}}).$$

Since $\text{dist}(f, f_{\text{rand}}) \le 2\sum_{i=0}^{n/3} \binom{n}{i}/2^n \le o(1)$, to complete the proof it suffices to fix $\epsilon < \epsilon' < \frac{1}{2}$ and show that $\text{dist}(f_{\text{rand}}, g_\pi) > \epsilon'$ with high probability.

Let $\eta = 1 - 2\epsilon'$. For any $x \in \{0,1\}^n$, $f_{\text{rand}}(x) = g_\pi(x)$ with probability $\frac{1}{2}$, so $\text{E}[\text{dist}(f_{\text{rand}}, g_\pi)] = \frac{1}{2}$. By Chernoff's bound (see, e.g., Appendix A in [2]),

$$\Pr[\text{dist}(f_{\text{rand}}, g_\pi) < \epsilon'] = \Pr[\text{dist}(f_{\text{rand}}, g_\pi) < (1-\eta)\tfrac{1}{2}] \le e^{-2^n \eta^2/6} \le o(\tfrac{1}{n!}).$$

Taking the union bound over all choices of $\pi \in \mathcal{S}_n$ completes the proof. □

Let $\mathcal{T}$ be any deterministic non-adaptive algorithm that attempts to test g-isomorphism with at most $\frac{n}{100}$ queries to an unknown function f. We will show that $\mathcal{T}$ cannot reliably distinguish between the cases where f was drawn from $\mathcal{F}_{\text{yes}}$ or from $\mathcal{F}_{\text{no}}$.

Let $Q \subseteq \{0,1\}^n$ be the set of queries performed by $\mathcal{T}$ on f. We partition the queries in Q in two: the set $Q_b = \{q \in Q : \frac{n}{3} \leq |q| \leq \frac{2n}{3}\}$ of *balanced* queries, and the set $Q_u = Q \setminus Q_b$ of *unbalanced* queries.

When f is drawn from $\mathcal{F}_{\text{yes}}$ or from $\mathcal{F}_{\text{no}}$, the responses to the unbalanced queries Q_u are consistent with some function g_π isomorphic to g. Our next proposition shows that when $\mathcal{T}$ makes only $\frac{n}{100}$ queries to f, then in fact the responses to the unbalanced queries will be consistent with *many* functions isomorphic to g. More precisely, define

$$\mathit{\Pi}_g(f, Q_u) = \{\pi \in \mathcal{S}_n : g_\pi(Q_u) = f(Q_u)\}$$

to be the set of permutations π for which g_π is consistent with the responses to the queries Q_u. The following proposition shows that when the unknown function is drawn from $\mathcal{F}_{\text{yes}}$ or from $\mathcal{F}_{\text{no}}$, then with high probability the set $\mathit{\Pi}_g(f, Q_u)$ is large.

Proposition 3.3. *Let Q_u be any set of unbalanced queries and let f be a function drawn from $\mathcal{F}_{\text{yes}}$ or from $\mathcal{F}_{\text{no}}$. Then for any $0 < t < 1$,*

$$\Pr_f\left[|\mathit{\Pi}_g(f, Q_u)| < t \cdot \tfrac{n!}{2^{|Q_u|}}\right] \leq t.$$

Proof. When $f \sim \mathcal{F}_{\text{yes}}$ or $f \sim \mathcal{F}_{\text{no}}$, then $f(x) = g_\pi(x)$ for every unbalanced input x, where π is chosen uniformly at random from $\mathcal{S}_n$. So it suffices to show that $\Pr_\pi[|\mathit{\Pi}_g(g_\pi, Q_u)| < t \cdot \frac{n!}{2^{|Q_u|}}] \leq t$.

For every $r \in \{0,1\}^{|Q_u|}$, let $S_r \subseteq \mathcal{S}_n$ be the set of permutations σ for which $g_\sigma(Q_u) = r$. A set S_r is *small* if $|S_r| \leq t\frac{n!}{2^{|Q_u|}}$. The union of all small sets covers at most $2^{|Q_u|} \cdot t\frac{n!}{2^{|Q_u|}} = tn!$ permutations, so the probability that a randomly chosen permutation π belongs to a small set is at most t. □

The last proposition showed that when f is drawn from $\mathcal{F}_{\text{yes}}$ or from $\mathcal{F}_{\text{no}}$, then with high probability $\mathit{\Pi}_g(f, Q_u)$ is large; the next lemma shows that conditioned on $\mathit{\Pi}_g(f, Q_u)$ being large, the distribution on the responses to the balanced queries is nearly uniform, even when $f \sim \mathcal{F}_{\text{yes}}$. Specifically, given a function g and a set S of permutations, we define the *discrepancy of g on S* to be

$$\mathit{\Delta}_S(g) = \max_{\substack{Q_b : |Q_b| = \frac{n}{100} \\ r \in \{0,1\}^{|Q_b|}}} \left| \Pr_{\pi \in S}[g_\pi(Q_b) = r] - 2^{-\frac{n}{100}} \right|.$$

We then define the *discrepancy* of g to be

$$\mathit{\Delta}(g) = \max_{\substack{Q_u : |Q_u| = \frac{n}{100} \\ \pi : |\mathit{\Pi}_g(g_\pi, Q_u)| \geq n!/2^{n/50}}} \mathit{\Delta}_{\mathit{\Pi}_g(g_\pi, Q_u)}(g).$$

The following lemma shows that $\mathit{\Delta}(g)$ is small for almost all functions g.

Lemma 3.4. *When g is drawn uniformly at random from the set of functions $\{0,1\}^n \to \{0,1\}$,*

$$\Pr_g \left[\Delta(g) > \tfrac{1}{3} \cdot 2^{-\frac{n}{100}}\right] \le 2^{-\Omega(2^{n/25})}.$$

We prove Lemma 3.4 in the next section, but first we show how it implies Theorem 1.1.

Proof (Theorem 1.1). By Lemma 3.4, with probability at least $1 - 2^{-\Omega(2^{n/25})} = 1 - o(1)$, the discrepancy of a randomly drawn function $g : \{0,1\}^n \to \{0,1\}$ is $\Delta(g) \le \frac{1}{3}2^{-\frac{n}{100}}$. Fix g to be any function that satisfies this condition. We will show that testing isomorphism to g requires at least $\frac{n}{100}$ queries.

As discussed earlier, we complete the proof with Yao's Minimax Principle, with the distributions $\mathcal{F}_{\text{yes}}$ and $\mathcal{F}_{\text{no}}$ as defined at the beginning of the section. Let $\mathcal{T}$ be any deterministic non-adaptive algorithm that makes at most $\frac{n}{100}$ queries to the input function f, and let $Q = Q_u \cup Q_b$ represent the queries made by $\mathcal{T}$. Without loss of generality, we can assume $|Q_u| = |Q_b| = \frac{n}{100}$. (If $|Q_b| < \frac{n}{100}$, simply add extra balanced queries to Q_b; this can only help $\mathcal{T}$ determine whether f was drawn from $\mathcal{F}_{\text{yes}}$ or from $\mathcal{F}_{\text{no}}$. Similarly, adding unbalanced queries to Q_u can only help $\mathcal{T}$.)

By Proposition 3.3, the probability that $|\Pi_g(f, Q_u)| < \frac{n!}{2^{n/50}}$ is at most $\frac{1}{2^{n/100}} = o(1)$. Assume, thus, that this event does not happen. Let $\mathcal{R}_{\text{yes}}$ and $\mathcal{R}_{\text{no}}$ be the distribution of the responses to the balanced queries Q_b. Then the total variation distance between $\mathcal{R}_{\text{yes}}$ and $\mathcal{R}_{\text{no}}$ is bounded by

$$\begin{aligned} d_{TV}(\mathcal{R}_{\text{yes}}, \mathcal{R}_{\text{no}}) &= \frac{1}{2} \sum_{r \in \{0,1\}^{\frac{n}{100}}} \left| \Pr_{\pi \in \Pi_g(f, Q_u)}[g_\pi(Q_b) = r] - 2^{-\frac{n}{100}} \right| \\ &\le \tfrac{1}{2} \cdot 2^{\frac{n}{100}}\, \Delta(g) \le \frac{1}{6}. \end{aligned} \tag{1}$$

Therefore, if $\mathcal{T}$ accepts functions drawn from $\mathcal{F}_{\text{yes}}$ with probability at least $\frac{2}{3}$, (1) implies that $\mathcal{T}$ also accepts functions drawn from $\mathcal{F}_{\text{no}}$ with probability at least $\frac{2}{3} - \frac{1}{6} = \frac{1}{2}$. But by Proposition 3.2, a function drawn from $\mathcal{F}_{\text{no}}$ is ϵ-far from isomorphic to g with probability $1 - o(1)$, so $\mathcal{T}$ can't be a valid ϵ-tester for isomorphism to g. □

3.3 Proof of Lemma 3.4

The first step in the proof of Lemma 3.4 is to show that for any sufficiently small set Q of balanced queries and sufficiently large set S of permutations, the set $\{\pi(Q)\}_{\pi \in S}$ can be partitioned into a number of large pairwise disjoint sets. The proof of this claim uses the celebrated theorem of Hajnal and Szemerédi [15].

Hajnal-Szemerédi Theorem. Let G be a graph on n vertices with maximum vertex degree $\Delta(G) \le d$. Then G has a $(d+1)$-coloring in which all the color classes have size $\lfloor \frac{n}{d+1} \rfloor$ or $\lceil \frac{n}{d+1} \rceil$.

Lemma 3.5. *Let S be a set of at least $\frac{n!}{2^{n/50}}$ permutations on $[n]$, and let Q_b be a set of at most $\frac{n}{100}$ balanced queries. Then there exists a partition $S_1\dot\cup\cdots\dot\cup S_k$ of the permutations in S such that for $i = 1, 2, \ldots, k$,*

(i) $|S_i| \geq 2^{n/20}$, *and*
(ii) The sets $\{\pi(Q_b)\}_{\pi\in S_i}$ *are pairwise disjoint.*

Proof. Construct a graph G on S where two permutations σ, τ are adjacent iff there exist $u, v \in Q_b$ such that $\sigma(u) = \tau(v)$. By this construction, when T is a set of permutations that form an independent set in G, then $\{\pi(Q_b)\}_{\pi\in T}$ are pairwise disjoint.

Consider a fixed permutation $\sigma \in S$. A second permutation τ is adjacent to σ in G iff there are two vectors u, v in Q_b such that the permutation $\tau\sigma^{-1}$ maps the indices where u has value 1 to the indices where v has value 1 as well. There are $\binom{|Q_b|}{2} \leq (\frac{n}{100})^2$ ways to choose $u, v \in Q_b$ and at most $|u|!(n - |u|)!$ ways to satisfy the mapping condition, so the graph has degree at most

$$\max_{\frac{n}{3}\leq k\leq\frac{2n}{3}} \left(\tfrac{n}{100}\right)^2 \cdot k!\,(n-k)! = \left(\frac{n}{100}\right)^2 \cdot \left(\frac{n}{3}\right)! \left(\frac{2n}{3}\right)! = \left(\frac{n}{100}\right)^2 \cdot \frac{n!}{\binom{n}{n/3}} \leq \frac{n!}{2^{cn}} - 1$$

for a constant $c = 1 - H_2(\frac{1}{3}) - o(1) \geq 0.07$.[3] Therefore, by the Hajnal-Szemerédi Theorem, G can be colored with $n!/2^{0.07n}$ colors, with each color class having size at least $\frac{n!/2^{n/50}}{n!/2^{0.07n}} = 2^{n/20}$. □

Lemma 3.5 is useful because most functions g have low discrepancy on large pairwise disjoint sets.

Lemma 3.6. *Fix Q_b to be a set of $\frac{n}{100}$ balanced queries and fix $r \in \{0,1\}^{\frac{n}{100}}$. Let S be a fixed set of at least $2^{\frac{n}{20}}$ permutations such that the sets $\{\pi(Q_b)\}_{\pi\in S}$ are pairwise disjoint. Then*

$$\Pr_g\left[\left|\Pr_{\pi\in S}[g_\pi(Q_b) = r] - 2^{-\frac{n}{100}}\right| > \tfrac{1}{3}\cdot 2^{-\frac{n}{100}}\right] < 2^{-\Omega(2^{n/25})}.$$

Proof. For every function $g : \{0,1\}^n \to \{0,1\}$ and every permutation π of $[n]$, define the indicator random variable

$$X_{g,\pi} = \begin{cases} 1 & \text{if } g_\pi(Q_b) = r, \\ 0 & \text{otherwise.} \end{cases}$$

When g is chosen uniformly at random from the set of all boolean functions $\{0,1\}^n \to \{0,1\}$, $\mathrm{E}_g[X_{g,\pi}] = \Pr_g[g_\pi(Q_b) = r] = 2^{-\frac{n}{100}}$, so

$$\mathrm{E}_g\left[\Pr_{\pi\in S}[g_\pi(Q_b) = r]\right] = \frac{1}{|S|}\sum_{\pi\in S}\mathrm{E}_g[X_{g,\pi}] = 2^{-\frac{n}{100}}.$$

[3] $H_2(p)$ represents the binary entropy of p. $H_2(\frac{1}{3}) \approx 0.918$.

Furthermore, the pairwise disjointness property of S guarantees that the indicator variables $X_{g,\pi}$ are pairwise independent. Therefore, by Chernoff's bound,

$$\Pr_{g}\left[\left|\Pr_{\pi\in S}[g_\pi(Q_b)=r]-2^{-\frac{n}{100}}\right|>\tfrac{1}{3}\cdot 2^{-\frac{n}{100}}\right]<e^{-\Omega(|S|2^{-n/100})}. \qquad \square$$

The proof of Lemma 3.4 can now be completed as follows.

Proof (Lemma 3.4). Fix a permutation π and a set Q_u of $\frac{n}{100}$ unbalanced queries such that $|\Pi_g(g_\pi, Q_u)| \geq \frac{n!}{2^{n/50}}$. Let $S = \Pi_g(g_\pi, Q_u)$, and fix a set Q_b of $\frac{n}{100}$ balanced queries.

By Lemma 3.5, there exists a partition $S_1\dot\cup\cdots\dot\cup S_k$ of S such that for each part S_i, $|S_i| \geq 2^{n/20}$ and $\{\pi(Q_b)\}_{\pi\in S_i}$ are pairwise disjoint. By Lemma 3.6, for every set S_i in the partition,

$$\Pr_{g}\left[\left|\Pr_{\pi\in S_i}[g_\pi(Q_b)=r]-2^{-\frac{n}{100}}\right|>\tfrac{1}{3}\cdot 2^{-\frac{n}{100}}\right]\leq 2^{-\Omega(2^{n/25})}.$$

Taking the union bound over all $k < n!$ sets S_i, we get that

$$\Pr_{g}\left[\left|\Pr_{\pi\in S}[g_\pi(Q_b)=r]-2^{-\frac{n}{100}}\right|>\tfrac{1}{3}\cdot 2^{-\frac{n}{100}}\right]< n!\cdot 2^{-\Omega(2^{n/25})}.$$

Applying a union bound once again, this time over all $\binom{2^n}{n/100} < 2^{\frac{n^2}{100}}$ choices of Q_b and $2^{\frac{n}{100}}$ choices for r, we obtain

$$\Pr_{g}\left[\Delta_S(g)>\tfrac{1}{3}\cdot 2^{-\frac{n}{100}}\right]<2^{\frac{n^2}{100}+\frac{n}{100}}\cdot n!\cdot 2^{-\Omega(2^{n/25})}.$$

Finally, applying the union bound one last time over the $n!$ choices for π and $\binom{2^n}{n/100} \leq 2^{\frac{n^2}{100}}$ choices for Q_u, we get

$$\Pr_{g}\left[\Delta(g)>\frac{1}{3}\cdot 2^{-\frac{n}{100}}\right]<2^{\frac{2n^2}{100}+\frac{n}{100}}\cdot n!^2\cdot 2^{-\Omega(2^{n/25})}=2^{-\Omega(2^{n/25})}. \qquad \square$$

4 Testing Isomorphism of Two Unknown Functions

4.1 Upper Bound

```
ALGORITHM $\mathcal{T}$
1. Generate two sets $Q_f, Q_g \subset \{0,1\}^n$ of $2^{n/2}\sqrt{\frac{n\ln n}{\epsilon}}$ queries independently
   and uniformly at random.
2. Query $f(x)$ for every $x \in Q_f$.
3. Query $g(x)$ for every $x \in Q_g$.
4. Accept iff there exists $\pi \in \mathcal{S}_n$ such that for every element $x \in Q_f$ where
   $\pi(x) \in Q_g$, $f(x) = g(\pi(x))$.
```

The algorithm $\mathcal{T}$ is non-adaptive and makes $\tilde{O}(2^{n/2})$ queries. Clearly, it always accepts when f and g are isomorphic. The following simple argument completes the proof of correctness of the algorithm by showing that it rejects functions that are ϵ-far from isomorphic with high probability.

Proposition 4.1. *Fix $\epsilon > 0$. Let f and g be ϵ-far from isomorphic. Then $\mathcal{T}$ rejects (f, g) with probability $1 - o(1)$.*

Proof. For any permutation $\pi \in \mathcal{S}_n$, there are at least $\epsilon 2^n$ inputs $x \in \{0,1\}^n$ for which $f(x) \neq g(\pi(x))$. It is not too difficult to show that the probability that none of these inputs satisfy $x \in Q_f$ and $\pi(x) \in Q_g$ is at most

$$4\left(1 - \frac{|Q_f|}{2^n} \cdot \frac{|Q_g|}{2^n}\right)^{\epsilon 2^n} = 4\left(1 - \frac{n \ln n}{\epsilon 2^n}\right)^{\epsilon 2^n} \leq 4e^{-\frac{n \ln n}{\epsilon 2^n} \cdot \epsilon 2^n} = 4n^{-n}.$$

By the union bound, the probability that f and g are accepted by the algorithm is at most $n!/n^n = o(1)$. □

4.2 Lower Bound

The following Lemma, combined with the upper bound in the previous section, implies Theorem 1.3.

Lemma 4.2. *Any algorithm for testing two unknown functions $f, g : \{0,1\}^n \to \{0,1\}$ for the property of being isomorphic must make at least $\Omega(\frac{2^{n/2}}{n^{1/4}})$ queries to the functions.*

Proof. Let $\mathcal{T}$ be an algorithm making $o(\frac{2^{n/2}}{n^{1/4}})$ queries to f and g. We will define two distributions $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$ on pairs of functions (f, g) that are isomorphic and ϵ-far from isomorphic with probability $1 - o(1)$, respectively, and show that $\mathcal{T}$ can not determine with probability greater than $\frac{1}{2} + o(1)$ which distribution generated an input.

Let $T = \{x \in \{0,1\}^n : \frac{n}{2} - \sqrt{n} \leq |x| \leq \frac{n}{2} + \sqrt{n}\}$ consist of the elements in the middle slice of the hypercube and let M be the set of all functions from $\{0,1\}^n$ to $\{0,1\}$ that map each $x \notin T$ to 0. A pair of functions (f, g) from $\mathcal{D}_{\text{yes}}$ is drawn by the following procedure:

1. Pick $\pi \in \mathcal{S}_n$ uniformly at random.
2. Choose $f \in M$ uniformly at random.
3. Let $g = f_\pi$.

A pair of functions (f, g) is drawn from $\mathcal{D}_{\text{no}}$ by independently choosing two functions uniformly at random from M. With probability $1 - o(1)$, f is $\frac{1}{4}$-far from isomorphic to g.

We now introduce two random processes P_{yes} and P_{no} that answer the queries of $\mathcal{T}$ while generating a pair of functions (f, g) from $\mathcal{D}_{\text{yes}}$ or from $\mathcal{D}_{\text{no}}$, respectively. Without loss of generality, we can assume that the tester queries the value of f or of g only on inputs $x \in T$, since functions drawn from $\mathcal{D}_{\text{yes}}$ or from $\mathcal{D}_{\text{no}}$ always take the value 0 on the remaining inputs.

The process P_{yes} starts by choosing a permutation $\pi \in \mathcal{S}_n$ uniformly at random. It then proceeds to answer all the queries of the algorithm $\mathcal{T}$ randomly, with one exception: P_{yes} "quits" if $\mathcal{T}$ queries the value of $f(x)$ after previously having queried $g(\pi(x))$, and similarly P_{yes} quits if $\mathcal{T}$ queries $g(x)$ after having queried $f(\pi^{-1}(x))$.

When P_{yes} quits or reaches the end of the queries, it completes the generation of (f, g) by choosing f uniformly at random from all the functions that are consistent with the previously-answered queries (note: in this step, the value of $f(x)$ for every x where $g(\pi(x))$ was queried is also determined by the value that was returned to the tester) and setting $g = f_\pi$. If there are more queries that have not yet been answered because P_{yes} quit, they are answered as per the generated f and g.

The process P_{no} is defined similarly. First, it chooses a permutation $\pi \in \mathcal{S}_n$ uniformly at random. It then answers the queries of $\mathcal{T}$ randomly, with the same exception as in the P_{yes} case: if $\mathcal{T}$ queries $f(x)$ after having queried $g(\pi(x))$, or if $\mathcal{T}$ queries $g(x)$ after having queried $f(\pi^{-1}(x))$, then P_{no} "quits".

When P_{no} quits or reaches the end of the queries, it completes the definitions of f and of g independently, randomly fixing the value of $f(x)$ and $g(x)$ for every input $x \in T$ that has not been queried by $\mathcal{T}$. If P_{no} quit before answering all the queries, those queries are then answered with the values of f and g that have been fixed.

It is easy to check that P_{yes} and P_{no} generate pairs of functions from $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$, respectively. Furthermore, when P_{yes} and P_{no} do not quit, they induce the same (i.e., uniformly random) distribution on the responses. So to complete the proof of the Lemma, it suffices to show that neither process quits with probability greater than $o(1)$.

The process P_{yes} or P_{no} quits if there is a pair of inputs $x_f, x_g \in T$ such that $f(x_f)$ and $g(x_g)$ are queried by $\mathcal{T}$ and $\pi(x_f) = x_g$. For any such pair, the probability that $\pi(x_f) = x_g$ is at most $O(\frac{\sqrt{n}}{2^n})$. But the answers to the queries yield no information about π to the tester $\mathcal{T}$, so the probability that it causes P_{yes} or P_{no} to quit is at most $o(\frac{2^{n/2}}{n^{1/4}})^2 \cdot O(\frac{\sqrt{n}}{2^n}) = o(1)$. $\square$

Acknowledgements

E.B. wishes to thank Ryan O'Donnell for much valuable advice throughout the course of this research and Michael Saks for enlightening discussions.

Some of this work was completed while the authors were participating in the 2010 ITCS mini-workshop on property testing. We thank the organizers, Oded Goldreich and Yuexuan Wang, for inviting us and everyone at the Institute for Theoretical Computer Science at Tsinghua University for their wonderful hospitality.

References

1. Noga Alon, Eldar Fischer, Michael Krivelevich, and Mario Szegedy. Efficient testing of large graphs. *Combinatorica*, 20(4):451–476, 2000.
2. Noga Alon and Joel H. Spencer. *The Probabilistic Method.* Wiley, third edition, 2008.
3. Eric Blais and Ryan O'Donnell. Lower bounds for testing function isomorphism. In *Conference on Computational Complexity*, 2010.
4. Sourav Chakraborty, David García-Soriano, and Arie Matsliah. Nearly tight bounds for testing function isomorphism, 2010. Manuscript.
5. Hana Chockler and Dan Gutfreund. A lower bound for testing juntas. *Information Processing Letters*, 90(6):301–305, 2004.
6. Ilias Diakonikolas, Homin K. Lee, Kevin Matulef, Krzysztof Onak, Ronitt Rubinfeld, Rocco A. Servedio, and Andrew Wan. Testing for concise representations. In *Proc. 48th Symposium on Foundations of Computer Science*, pages 549–558, 2007.
7. Yevgeniy Dodis, Oded Goldreich, Eric Lehman, Sofya Raskhodnikova, Dana Ron, and Alex Samorodnitsky. Improved testing algorithms for monotonicity. In *Proceedings of the 3rd RANDOM conference*, pages 97–108, 1999.
8. Eldar Fischer. The art of uninformed decisions: a primer to property testing. *Bull. Eur. Assoc. for Theoretical Comp. Sci.*, 75:97–126, 2001.
9. Eldar Fischer. The difficulty of testing for isomorphism against a graph that is given in advance. *SIAM J. on Comp.*, 34(5):1147–1158, 2005.
10. Eldar Fischer, Guy Kindler, Dana Ron, Shmuel Safra, and Alex Samorodnitsky. Testing juntas. *J. Comput. Syst. Sci.*, 68(4):753–787, 2004.
11. Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, Ronitt Rubinfeld, and Alex Samorodnitsky. Monotonicity testing over general poset domains. In *STOC '02: Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 474–483, 2002.
12. Eldar Fischer and Arie Matsliah. Testing graph isomorphism. *SIAM J. Comput.*, 38(1):207–225, 2008.
13. Oded Goldreich, Shafi Goldwasser, Eric Lehman, Dana Ron, and Alex Samorodnitsky. Testing monotonicity. *Combinatorica*, 20(3):301–337, 2000.
14. Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998.
15. András Hajnal and Endre Szemerédi. Proof of a conjecture of Paul Erdős. In Paul Erdős, Alfréd Rényi, and Vera T. Sós, editors, *Combinatorial Theory and its Applications*, pages 601–623. 1969.
16. Kevin Matulef, Ryan O'Donnell, Ronitt Rubinfeld, and Rocco A. Servedio. Testing halfspaces. In *SODA '09: Proceedings of the Nineteenth Annual ACM -SIAM Symposium on Discrete Algorithms*, pages 256–264, 2009.
17. Kevin Matulef, Ryan O'Donnell, Ronitt Rubinfeld, and Rocco A. Servedio. Testing ± 1-weight halfspaces. In *RANDOM '09*, pages 646–657, 2009.
18. Dana Ron. Property testing: a learning theory perspective. *Foundations and Trends in Machine Learning*, 1(3):307–402, 2008.
19. Dana Ron. Algorithmic and analysis techniques in property testing. *Foundations and Trends in Theoretical Computer Science*, 5(2):73–205, 2009.
20. Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996.
21. Andrew Chi-Chih Yao. Probabilistic computations: Toward a unified measure of complexity. In *Proceedings of the 28th IEEE Symposium on Foundations of Computer Science*, pages 222–227, 1977.