Randomly Testing Boolean Functions

Kent Quanrud

December 1, 2020

1 Testing boolean formulae with 3 queries, and 8 letter graph CSP's

A **boolean function** is a function $f : \{0,1\}^n \to \mathbb{R}$ that takes as input a sequence of bits and outputs a single value - often another bit.[1] Clearly *any* (deterministic) program making a binary decision is a boolean function, which makes boolean functions a natural object of study. Here we explore a *testing* approach that takes a boolean function f as a black box, queries f at a limited number of inputs, and analyzes the outputs to make useful statements about f. However, the same universality of boolean functions that makes them so attractive also makes it seem rather daunting to be able to obtain concrete and useful observations. Nonetheless today we will see a few interesting things that one *can* do, at least *approximately*, by combination of *randomization* and an appropriate change of basis.

Today we will discuss a few introductory topics in *property testing*, which takes as input f and tries to decide if f has a certain property. We would only be able to do so approximately, and differentiate functions that have the property (exactly) from functions that fail to have the property for a constant fraction of the inputs. For example, we will show how to approximately test boolean functions for the following properties.

- Linearity: whether $f : \{0,1\}^n \to \{0,1\}$ satisfies $f(x+y) = f(x) + f(y)$ for all x and y.
- Dictatorship: whether $f : \{0,1\}^n \to \{-1,1\}$ is of the form $f(x) = (-1)^{x_i}$ for some $i \in [n]$.

The main goal of today's discussion is to describe the following **universal tester** for proof systems.

Theorem 1.1. *Let $L \subseteq \{0,1\}^n$. Let $p = 2^{2^n}$. Then there is a randomized algorithm $A_L : \{0,1\}^n \times \{0,1\}^p \to \{0,1\}$ that, given oracle access to $x \in \{0,1\}^n$ and $y \in \{0,1\}^p$, has the following properties.*

1. *A_L makes 3 queries to bits of x or y.*
2. *If $x \in L$, then there exists $y \in \{0,1\}^p$ such that $A_L(x,y) = 1$ always.*
3. *If $x \notin L$ then for all $y \in \{0,1\}^p$, we have*

$$\mathbf{P}[A_L(x,y) = 0] \geq .001 \min_{y \in L} \frac{\|x - y\|_0}{n}.$$

The connection to boolean functions is not at all clear from the theorem statement above. Let us briefly describe the algorithm underlying Theorem 1.1 at a high level, which will make the connection more clear. Let $N = 2^n$, and identify $\{0,1\}^n \equiv N$. For each $i \in [N]$, let $\chi_i : \{0,1\}^N \to \{-1,1\}$ be the function defined by

$$\chi_i(x) = \begin{cases} 1 & \text{if } x_0 = 0 \\ -1 & \text{if } x_i = 1 \end{cases}$$

[1] Of course one can consider functions that output more than one value - but real-valued boolean functions suffice for the current discussion. In fact this note only really requires boolean functions of the form $f : \{0,1\}^n \to \{-1,1\}$.

χ_i is called a **dictator function** and is the topic of Section 4. Identifying $L \subseteq \{0,1\}^n$ as a subset of $[N]$, let $D_L = \{\chi_i : i \in L\}$. Note that in general, Boolean functions $f : \{0,1\}^N \to \{-1,1\}$ can be expressed in $\left(2^N = 2^{2^n}\right)$-dimensional vectors. We will create a test that, given $x \in \{0,1\}^n$ and $y \in \{0,1\}^p$, simultaneously tests if $y \in D_L$ and, conditional on $y \in D_L$, if $y = \chi_x$ (where we identify x with an index in N).

Thus the theorem reduces to understanding two tests about Boolean functions. We will build these tools over the course of this note and return to this analysis at the end.

To motivate Theorem 1.1, recall that an important ingredient of the PCP theorem from previous discussions was a subroutine that took constant-size boolean functions and output graph CSP's that model them in an error preserving fashion. Let us now show how to obtain this result using the universal tester above.

Theorem 1.2. *Let $L \subset \{0,1\}^n$ be a language. Then there exists a graph CSP with graph $G_L = (V_L, E_L)$, alphabet A, and constraints $\left\{C_e \subset A^2 : e \in E_L\right\}$ with the following properties.*

1. *$A = \{0, 1, \ldots, 7\}$.*
2. *There are n vertices $X_1, \ldots, X_n \in V_L$ that only take labels in $\{0,1\} \subset A$, and satisfy the following.*
   - *If $x_1, \ldots, x_n \in \{0,1\}$ is such that $(x_1, \ldots, x_n) \in L$, then there exists an assignment $\sigma : V_L \to \{0,1\}$ such that $\sigma(X_i) = x_i$ for all i and $\textsc{unsat}(\sigma \mid G_L) = 0$.*
   - *If $x_1, \ldots, x_n \in \{0,1\}$ is such that $(x_1, \ldots, x_n) \notin L$, then for all assignments $\sigma : V_L \to \{0,1\}$ such that $\sigma(X_i) = x_i$ for all i, we have*

$$\textsc{unsat}(\sigma \mid G_L) \geq .001 \min_{y \in L} \frac{\|x - y\|_0}{n}.$$

Proof. Fix a universal tester T for the language L that takes as input $x \in \{0,1\}^n$ and $y \in \{0,1\}^p$, for $p = 2^{2^n}$. We create a vertex for each of the following.

1. For each $i \in [n]$, a vertex X_i (modeling the input bit x_i).
2. For each $i \in [p]$, a vertex Y_i (modeling the proof bit y_i).
3. The universal tester T flips a finite number of coins. For every possible outcome of coin tosses ω, we create a vertex Z_ω.

We have an alphabet $A = \{0, \ldots, 7\}$ which we identify with $\{0,1\}^3$. For every outcome of coin tosses ω, we create 4 constraints/edges involving Z_ω based on the mechanism of the tester T when the coin tosses are ω.

1. We create a self-loop at Z_ω where, given a label

$$\sigma(Z_\omega) = (\sigma_1(Z_\omega), \sigma_2(Z_\omega), \sigma_3(Z_\omega)) \in \{0,1\}^3,$$

   we satisfy the constraint iff the following conditions hold. When the coin tosses of T are ω, and the three queries return $\sigma_1(Z_\omega)$, $\sigma_2(Z_\omega)$, and $\sigma_3(Z_\omega)$, the tester accepts (x, y).

2. For $i = 1, 2, 3$, suppose the ith query of T is to the k_ith bit of x. Then we create a constraint between Z_ω and X_{k_i} that accepts iff $\sigma_i(Z_\omega) = \sigma(X_{k_i})$. Similarly, if instead the ith query is to k_ith bit of y, we make the same constraint between Z_ω and Y_k.

One can now verify that this graph CSP satisfies the conditions of the statement. In particular, because the universal tester has a rejection probability that is proportional to the distance from x to L, the fraction of unsatisfied constraints in any labeling extending x will be proportional to the distance from x to L. ∎

While this note is motivated by the PCP theorem and limited in scope to the techniques that lead to the universal tester, there are many other applications of property testing and Boolean analysis. See [3] for a booklength treatment on these topics. These notes are based on chapters 1, 2, and 7 of [3]. Property testing also extends far beyond boolean functions. We recommend Prof. Grigorescu's Spring 2021 class on *sublinear time algorithms* for more topics in this area.

2 Fourier analysis of boolean functions

A **boolean function** is a real-valued function defined over bit-strings of a fixed length; i.e.,

$$f : \{0,1\}^n \to \mathbb{R}.$$

Note that sums of boolean functions and rescaled boolean functions are again boolean functions. In particular, we can identify the set of all boolean functions with the 2^n-dimensional Euclidean vector space $\mathbb{R}^{\{0,1\}^n}$. Here the ith coordinate of the "vector" f is the value $f(i)$. Rather than the standard Euclidean inner product $\langle x, y\rangle = \sum_i x_i y_i$, it is more convenient to rescale $\langle \cdot, \cdot \rangle$ to the following inner product that we denote $\langle \cdot, \cdot \rangle_{\mathrm{b}}$:

$$\langle f, g\rangle_{\mathrm{b}} \stackrel{\text{def}}{=} \frac{1}{2^n}\langle f, g\rangle = \mathop{\mathbf{E}}_{x\sim\{0,1\}^n}[f(x)g(x)],$$

where in the RHS x is sampled uniformly from $\{0,1\}^n$. Let

$$\|f\|_{\mathrm{b}} = \sqrt{\langle f, f\rangle_{\mathrm{b}}} = \sqrt{\mathop{\mathbf{E}}_{x}\left[f^2(x)\right]}$$

denote the corresponding norm. This norm rescales the standard Euclidean norm by $2^{-n/2}$. Here are a couple helpful identities to get us started.

Lemma 2.1. *For two boolean functions $f, g : \{0,1\}^n \to \{0,1\}$, we have*

$$\|f - g\|_{\mathrm{b}}^2 = \mathop{\mathbf{P}}_{x}[f(x) \neq g(x)].$$

Lemma 2.2. *For two boolean functions $f, g : \{0,1\}^n \to \{-1,1\}$, we have*

$$\mathbf{P}[f(x) \neq g(x)] = \frac{1}{4}\|f - g\|_{\mathrm{b}}^2.$$

Lemma 2.3. *For any boolean function $f : \{0,1\}^n \to \{-1,1\}$, we have $\|f\|_{\mathrm{b}}^2 = 1$.*

We leave the proofs of the above as exercises.

So far we have expressed boolean functions in terms of their "truth tables" as vectors in $\mathbb{R}^{\{0,1\}^n}$, but of course there are many possible bases over $\mathbb{R}^{\{0,1\}^n}$ that one could work with. *Fourier analysis* is based on the following choice of basis. For each set $S \subseteq [n]$, define a boolean function $\chi_S : \{0,1\}^n \to \{-1,1\}$ by

$$\chi_S(x) = (-1)^{\sum_{i\in S} x_i} = \begin{cases} 1 & \text{if } \sum_{i\in S} x_i \text{ is even,} \\ -1 & \text{if } \sum_{i\in S} x_i \text{ is odd.} \end{cases}$$

We call χ_S the *Sth Fourier basis function*; sometimes χ_S is called the parity function over S. The Fourier basis functions have many convenient properties of which we list a few. For $S = \emptyset$, we have $\chi_\emptyset = \mathbb{1}$, the all-one's vector. For all $S, T \subseteq [n]$, we have

$$\chi_S \chi_T = \chi_{(S \triangle T)}, \tag{1}$$

where $S \triangle T = (S \cup T) \setminus (S \cap T)$ denotes the symmetric difference. We also have, for all nonempty sets $S \neq \emptyset$,

$$\mathbf{E}[\chi_S(x)] = 0.$$

The above is easy to see for singleton sets $S = \{i\}$. For general sets S, letting $i \in S$ and $S' = S - i$, we have

$$\mathbf{E}[\chi_S] \overset{(a)}{=} \mathbf{E}[\chi_{S'}(x)\chi_i(x)] \overset{(b)}{=} \mathbf{E}[\chi_{S'}(x)]\,\mathbf{E}[\chi_i(x)] \overset{(c)}{=} 0.$$

Here (a) is by (1). (b) is by independence. (c) is applies the singleton case. Finally, by combining the above observations, we have

$$\langle \chi_S, \chi_T \rangle_{\mathrm{b}} = \begin{cases} 1 & \text{if } S = T \\ 0 & \text{otherwise} \end{cases}$$

for any two sets $S, T \subset [n]$.

The last identity signifies that the set of functions $\{\chi_S : S \subseteq [n]\}$ are an orthonormal set. There are also 2^n many of them, and we are working in a 2^n-dimensional space, so in fact they form an orthornomal basis (w/r/t $\langle \cdot, \cdot \rangle_{\mathrm{b}}$). Linear algebra then dictates that *any* boolean function $f : \{0,1\}^n \to \mathbb{R}$ can be written uniquely as a linear combination of the Fourier basis functions $\{\chi_S : S \subseteq [n]\}$, and this representation is given by

$$f = \sum_{S \subseteq [n]} \langle f, \chi_S \rangle_{\mathrm{b}} \chi_S = \sum_{S \subseteq [n]} \mathop{\mathbf{E}}_{x}[f(x)\chi_S(x)]\chi_S.$$

Let $\hat{f} : 2^n \to \mathbb{R}$ denote the coordinates in this basis; i.e., $\hat{f}_S = \langle f, \chi_S \rangle_{\mathrm{b}}$ for each set $S \subseteq [n]$. The map $f \mapsto \hat{f}$ is unitary; that is, a rotation that preserves distances. Consequently for boolean functions $f, g : \{0,1\}^n \to \{0,1\}$ we have

$$\mathop{\mathbf{P}}_{x}[f(x) \neq g(x)] = \langle f - g, f - g \rangle_{\mathrm{b}} = \langle \hat{f} - \hat{g}, \hat{f} - \hat{g} \rangle = \|\hat{f} - \hat{g}\|^2,$$

where $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$ are the standard Euclidean norm. One should not underestimate the significance of this transformation. The Fourier transform gives a unitary transformation that maps boolean functions $f : \{0,1\}^n \to \{0,1\}$ into a Euclidean vector space such that the probability of two functions agreeing is captured exactly by the norm.

3 Linearity

3.1 Testing linearity

A boolean function $f : \{0,1\}^n \to \{0,1\}$ is said to be **linear (mod 2)** if

$$f(x + y) = f(x) + f(y)$$

for all $x, y \in \{0,1\}^n$, where all additions are made modulo 2. For technical reasons it is instead convenient to consider functions of the form $f : \{0,1\}^n \to \{-1,1\}$, and define such a function to be linear if

$$f(x + y) = f(x)f(y)$$

for all $x, y \in \{0,1\}^n$. Of course, by mapping 0 to 1 and 1 to -1, there is an easy 1-to-1 correspondence between our two classes of linear functions. Note that the Fourier basis functions $\chi_S : \{0,1\}^n \to \{-1,1\}$ are linear functions in the sense immediately above.

Our goal is to devise an algorithm that, given a boolean function $f : \{0,1\}^n \to \{-1,1\}$, decides if f is a linear function. Of course we can query f everywhere but this can be inefficient. We prefer to test f with only a few queries. We point out that a deterministic and exact algorithm is impossible with only a few queries, but still we will be able to show some interesting approximate and randomized guarantees.

The following simple procedure is maybe the most obvious one to try.

1. Draw $x, y \in \{0,1\}^n$ independently and uniformly at random.
2. Evaluate $f(x)$, $f(y)$, and $f(x+y)$.
3. Accept f is $f(x+y) = f(x)f(y)$.

This algorithm was analyzed by [1] as follows.

Theorem 3.1. *Let $f : \{0,1\}^n \to \{-1,1\}$. Then*

$$\min_S \mathop{\mathbf{P}}_x[f(x) \neq \chi_S(x)] \leq \mathop{\mathbf{P}}_{x,y}[f(x)f(y) \neq f(x+y)],$$

where $x, y \in \{0,1\}^n$ are distributed uniformly and independently over $\{0,1\}^n$.

Proof. For ease of notation, let $P = \mathbf{P}_{x,y}[f(x)f(y) \neq f(x+y)]$. Let $Z \in \{0,1\}$ be the indicator variable for the event that $f(x+y) \neq f(x)f(y)$. We have $P = \mathbf{E}[Z]$. We also have

$$Z = 1 - \frac{1}{4}(f(x)f(y) - f(x+y))^2 \overset{(a)}{=} 1 - \frac{1}{4}(2 - 2f(x)f(y)f(x+y)) = \frac{1}{2}(1 + f(x)f(y)f(x+y)),$$

where (a) observes that $f^2(x) = f^2(y) = f^2(x+y) = 1$. Thus

$$P = \mathbf{E}[Z] = \frac{1}{2} + \frac{1}{2}\mathop{\mathbf{E}}_x\left[f(x)\mathop{\mathbf{E}}_y[f(y)f(x+y)]\right] = \frac{1}{2} + \frac{1}{2}\langle f, h\rangle_{\mathrm{b}} \overset{(b)}{=} \frac{1}{2} + \frac{1}{2}\langle \hat{f}, \hat{h}\rangle,$$

where we define $h(x) = \mathbf{E}_y[f(y)f(x+y)]$. (b) applies the Fourier transform. *We claim that $\hat{h}_S = \hat{f}_S^2$ for all S.* Indeed, we have

$$\begin{aligned}\mathop{\mathbf{E}}_x[h(x)\chi_S(x)] &= \mathop{\mathbf{E}}_{x,y}[f(y)f(x+y)\chi_S(x)] \overset{(c)}{=} \mathop{\mathbf{E}}_{x,y}[f(y)f(x+y)\chi_S(y)\chi_S(x+y)] \\ &= \mathop{\mathbf{E}}_y[f(y)\chi_S(y)]\mathop{\mathbf{E}}_{x,y}[f(x+y)\chi_S(x+y)] = \langle f, \chi_S\rangle_{\mathrm{b}}^2.\end{aligned}$$

(c) is by linearity of χ_S. (d) observes that y and $x+y$ are independently and uniformly distributed in $\{0,1\}^n$. Plugging back in, we now have

$$P = \frac{1}{2} + \frac{1}{2}\sum_S \hat{f}_S^3 \overset{(d)}{\leq} \frac{1}{2} + \frac{1}{2}\max_S \hat{f}_S$$

(e) applies the fact that $\|\hat{f}\|^2 = 1$ for $f : \{0,1\} \to \{-1,1\}$. Rearranging, we have

$$\hat{f}_S \geq 2P - 1$$

for some set $S \subseteq [n]$. But then

$$4\mathop{\mathbf{P}}_x[f(x) \neq \chi_S] = \|f - \chi_S\|_{\mathrm{b}}^2 = \|f\|_{\mathrm{b}} + \|\chi_S\|_{\mathrm{b}} - 2\langle f, \chi_S\rangle = 2 - 2\hat{f}_S \leq 4P,$$

as desired. ∎

Note that if f is linear, then the linearity test succeeds one hundred percent of the time. But then the above theorem asserts there exists a basis function χ_S that agrees with f one hundred percent of the time. Thus we deduce the following.

Corollary 3.2. *All linear functions $f : \{0,1\}^n \to \{-1,1\}$ are of the form χ_S for some set S.*

3.2 Locally correcting for linearity

Theorem 3.3. *Let $f : \{0,1\} \to \{-1,1\}$ be ϵ-close to a basis function $\chi_S : \{0,1\} \to \{-1,1\}$. Given $x \in \{0,1\}^n$, consider the random value $f(x)f(x+y) \in \{-1,1\}$ where $y \in \{0,1\}^n$ is sampled uniformly at random. Then*

$$\mathop{\mathbf{P}}_{y}[f(y)f(x+y) = \chi_S(x)] \geq 1 - 2\epsilon.$$

Proof. $x + y$ and y are both distributed uniformly over $\{0,1\}^n$, and we have $f(y) = \chi_S(y)$ and $f(x+y) = \chi_S(x+y)$ each with probability of error $\leq \epsilon$. By the union bound, both occur with probability of error $\leq 2\epsilon$. But then we recover $\chi_S(x) = \chi_S(x+y)\chi_S(y)$. ∎

3.3 A remark on convolutions

A key component of the proof of Theorem 3.1 is the identity $\hat{h}_S = \hat{f}_S^2$ for the function $h(x) = \mathbf{E}_y[f(y)f(x+y)]$. More generally, for two boolean formulas $f, g : \{0,1\} \to \mathbb{R}$, the **convolution** of f and g, denoted $f * g$, is the function defined by

$$(f * g)(x) = \mathop{\mathbf{E}}_{y}[f(x)g(x+y)].$$

The following identity is called **Plancheral's identity** and generalizes the calculations used in Theorem 3.1.

Lemma 3.4. *Let $f, g : \{0,1\} \to \mathbb{R}$. Then $\widehat{(f*g)}_S = \hat{f}_S \hat{g}_S$ for all $S \subseteq [n]$.*

We leave the proof as an exercise.

4 Dictators

A function $f : \{0,1\}^n \to \{-1,1\}$ is a **dictator** if it is one of the singleton basis functions[2],

$$f = \chi_i \text{ for some } i \in [n].$$

The main goal in this section is to design a test for whether or not a function $f : \{0,1\}^n \to \{-1,1\}$ is a dictator function. The test we design will be a composition of two tests. First, clearly, any dictator function is linear, which gives our first test.

1. **Linearity test:** Sample $x, y \in \{0,1\}^n$ independently and reject if $f(x+y) \neq f(x)f(y)$.

Our second test will be a new one. Let $\Omega = \{0,1\}^3 \setminus \{(0,0,0),(1,1,1)\}$ be the set of triplets where not all coordinates are equal. Abusing notation, we write Ω^n to denote the triplets of vectors $x, y, z \in \{0,1\}^n$ such that for all i, $(x_i, y_i, z_i) \in \Omega$. We note that to sample a uniformly random $(x, y, z) \in \Omega$, one can independently sample, for each $i \in [n]$, three coordinates $(x_i, y_i, z_i) \in \Omega$ uniformly at random. We write $(x, y, z) \sim \Omega^n$ to denote $(x, y, z) \in \Omega^n$ sampled uniformly at random.

Observe that for any dictator function $f = \chi_i$, and $(x, y, z) \in \Omega^n$, we have $(f(x), f(y), f(z)) \in \Omega$. This gives our second test.

2. **Not-all-equal (NAE) test:** Sample $x, y, z \sim \Omega^n$. Reject f unless $(f(x), f(y), f(z)) \in \Omega$.

[2]For ease of notation, we write χ_i instead of $\chi_{\{i\}}$.

Theorem 4.1. *Let $f : \{0,1\}^n \to \{-1,1\}$ be a boolean function. Suppose f passes both the linearity and not-all-equals test with probability or error $\leq \epsilon$ for $\epsilon \leq .1$. Then there exists a coordinate $i \in [n]$ such that*

$$\mathop{\mathbf{P}}_{x}[f(x) \neq \chi_i(x)] \leq \epsilon,$$

where $x \in \{0,1\}^n$ is sampled uniformly at random.

To prove Theorem 4.1, we first require the following lemma analyzing the not-all-equal test.

Lemma 4.2. *Let $f : \{0,1\}^n \to \{-1,1\}$. Let $(x,y,z) \sim \Omega^n$. Then*

$$\mathbf{P}[(f(x), f(y), f(z)) \in \Omega] \leq \frac{7}{9} + \frac{2}{9}\sum_{i=1}^{n} \hat{f}_i^2.$$

We will prove this lemma below in Section 4.3. First, let us use it to prove Theorem 4.1.

Proof of Theorem 4.1. By the NAE test, we have $\sum_i \hat{f}_i^2 \geq 1 - 4.5\epsilon$. By the linearity test, we have that $\hat{f}_S \geq 1 - 2\epsilon$ for some S. But this set S must be a singleton $\{i\}$ because otherwise we have

$$1 = \|\hat{f}\|^2 \geq 1 - 4.5\epsilon + (1-2\epsilon)^2 > 1,$$

a contradiction. Thus $\hat{f}_i \geq 1 - 2\epsilon$ for some i. Then

$$4 \mathop{\mathbf{P}}_{x}[f(x) \neq \chi_i(x)] \overset{\text{(a)}}{=} \|f - \chi_i\|_{\mathrm{b}}^2 = \|\hat{f} - \hat{\chi}_i\|^2 \overset{\text{(b)}}{=} (\hat{f}_i - 1)^2 + 1 - \hat{f}_i^2 = 2 - 2\hat{f}_i \leq 4\epsilon,$$

as desired. (a) is by Lemma 2.2. (b) takes the Fourier transform. (c) uses the identity $\|\hat{f}\|^2 = 1$ for all $f : \{0,1\}^n \to \{-1,1\}$. ∎

The dictatorship test we have just developed requires 6 queries to f: three for the linearity test, and three for the not-all-equals test. We can reduce this to three queries at the cost of increase the error rate with a simple trick, as follows.

Theorem 4.3. *Let $n \in \mathbb{N}$. There is a 3-query test for the family of dictators $D = \{\chi_i \mid i \in [n]\}$ with the following guarantee. Given a function $f : \{0,1\}^n \to \{-1,1\}$:*

1. *If $f \in D$, then the test always accepts f.*
2. *If f is ϵ-far from any dictator function and $\epsilon \leq .2$, then the test f with probability $\geq \epsilon/4$.*

Proof. We choose *either* the linearity test or not-all-equals test, randomly selecting one of the two with equal probability. Clearly, if f is a dictator, then the test always passes. Otherwise, suppose f fails the test with probability $\leq p$. Consider the test where we run *both* tests on f; f fails this test with probability $\leq 2p$. It follows that for $p \leq .05$, f is at most $4p$-far from some dictator. ∎

4.1 Subclasses of dictators

We can extend the dictator test above to subfamilies of dictator functions as follows. For any set $S \subset [n]$, let

$$D_S = \{\chi_i : \{0,1\}^n \to \{-1,1\} \mid i \in S\}$$

be the set of dictator functions for coordinates $i \in S$. Suppose that given $f : \{0,1\}^n \to \mathbb{R}$ and $S \subset [n]$, we want to test if f is close to any dictator function in S. Consider the following.

1. With probability 1/2, run the dictatorship test from Theorem 4.1.

2. With probability 1/2, run the locally correcting protocol for linear functions for f with input string $\mathbb{1}_S$, accepting f if this protocol returns 1.

This test has the following bounds.

Theorem 4.4. *Given $S \subseteq [n]$, there is a 3-query test for the subfamily of dictators D_S with the following guarantee. Given a function $f : \{0,1\}^n \to \{-1,1\}$:*

1. *If $f \in D_S$, then the test always accepts f.*

2. *If f is ϵ-far from any function in D_S, then the test rejects f with probablity $\geq c\epsilon$ for some universal constant $c > 0$.*

Proof. The first property is immediate. Suppose $f \notin D_S$ and fails the test with probability p. Then f fails either test with probability $\leq 2p$. The first test implies that f is (cp)-far from a dictator for some universal constant $c > 0$. Because f is (cp)-far from a dictator χ_i and in particular from a linear function, the correction protocol returns $\chi_i(\mathbb{1}_S)$ with probability of error $\leq dcp$ for a universal constant $d > 0$. Since f passes that test with probability $2p$, we conclude that f is $O(p)$ close to χ_i for some $i \in \chi_i$. ∎

4.2 Noisy perturbation of boolean functions

It remains to analyze the not-all-equal test. Doing so requires analyzing boolean functions under random perturbations of their input, as follows.

For $x \in \{0,1\}^n$ and $p \in [0,1]$, let $N_p(x)$ be the distribution of random strings where each bit x_i is flipped independently with probability p. The random function N_p arose previously in the analysis of error correcting codes. For a boolean function $f : \{0,1\}^n \to \mathbb{R}$, we define the boolean function $\mathrm{T}_p f : \{0,1\}^n \to \mathbb{R}$ by

$$\big(\mathrm{T}_p f\big)(x) = \mathop{\mathbf{E}}_{y \sim N_p(x)}[f(y)].$$

Lemma 4.5. *Let $f : \{0,1\} \to \mathbb{R}$ be a boolean function. For $S \subset [n]$, $\widehat{(\mathrm{T}_p f)}_S = (1-2p)^{|S|}\hat{f}_S$.*

Proof. Since T_p and taking the Fourier transform are both linear functions, it suffices to prove the claim for $f = \chi_S$. We have

$$\begin{aligned}\big(\mathrm{T}_p \chi_S\big)(x) &= \mathop{\mathbf{E}}_{y \sim N_p(x)}[\chi_S(y)] = \prod_{i \in S} \mathop{\mathbf{E}}_{y \sim N_p(x)}[(-1)^{y_i}] = \prod_{i \in S}((1-p)(-1)^{x_i} - p(-1)^{x_i}) \\ &= \prod_{i \in S}(1-2p)(-1)^{x_i} = (1-2p)^{|S|}\chi_S(x),\end{aligned}$$

as desired. ∎

4.3 Analysis of the not-all-equals test

Finally, let us analyze the not-all-equals test and prove Lemma 4.2.

Lemma 4.2. *Let $f : \{0,1\}^n \to \{-1,1\}$. Let $(x,y,z) \sim \Omega^n$. Then*

$$\mathbf{P}[(f(x), f(y), f(z)) \in \Omega] \leq \frac{7}{9} + \frac{2}{9}\sum_{i=1}^{n} \hat{f}_i^2.$$

Proof. Let $P = \mathbf{P}[(f(x), f(y), f(z)) \in \Omega]$. Define a boolean function $\mathrm{NAE} : \{-1, 1\}^3 \to \{0, 1\}$ by setting

$$\mathrm{NAE}(a, b, c) = \begin{cases} 0 & \text{if } a = b = c, \\ 1 & \text{otherwise.} \end{cases}$$

We have

$$\begin{aligned} \mathrm{NAE}(a, b, c) &= \frac{1}{8}\Big((a-b)^2 + (a-c)^2 + (b-c)^2\Big) = \frac{1}{8}\Big(2a^2 + 2b^2 + 2c^2 - 2ab - 2ac - 2bc\Big) \\ &= \frac{3}{4} - \frac{1}{4}(ab + ac + bc). \end{aligned}$$

Thus

$$\begin{aligned} P = \mathbf{E}[\mathrm{NAE}(f(x), f(y), f(z))] &= \frac{3}{4} - \frac{1}{4}\mathbf{E}[f(x)f(y) + f(y)f(z) + f(x)f(z)] \\ &\overset{(a)}{=} \frac{3}{4} - \frac{3}{4}\mathbf{E}[f(x)f(y)] \end{aligned}$$

where (a) is by symmetry of Ω. Consider $\mathbf{E}[f(x)f(y)]$. We have

$$\begin{aligned} \mathbf{E}[f(x)f(y)] &\overset{(b)}{=} \mathop{\mathbf{E}}_{x, y \sim \mathcal{N}_{2/3}(x)}[f(x)f(y)] = \left\langle f, \mathrm{T}_{2/3} f \right\rangle_{\mathrm{b}} \overset{(c)}{=} \langle \hat{f}, \widehat{(\mathrm{T}_{2/3} f)} \rangle \\ &= \sum_S (-1/3)^{|S|} \hat{f}_S^2 \geq -\frac{1}{3}\sum_i \hat{f}_i^2 - \frac{1}{27}\sum_{\substack{|S| \geq 3 \\ |S| \text{ odd}}} \hat{f}_S^2 \\ &\overset{(d)}{\geq} -\frac{1}{3}\sum_i \hat{f}_i^2 - \frac{1}{27}\left(1 - \sum_i \hat{f}_i^2\right) = -\frac{1}{27} - \frac{8}{27}\sum_i \hat{f}_i^2. \end{aligned}$$

Here (b) observes that x is sampled uniformly from $\{0, 1\}^n$, and conditional on x, y is distributed as $\mathcal{N}_{2/3}(x)$. (c) applies the unitary Fourier transform. (d) is by Lemma 4.5. (e) is because $\sum_S \hat{f}_S^2 = \|\hat{f}\|^2 = 1$. Plugging back in, we have

$$P \leq \frac{3}{4} - \frac{3}{4}\left(-\frac{1}{27} - \frac{8}{27}\sum_i \hat{f}_i^2\right) = \frac{7}{9} + \frac{2}{9}\sum_i \hat{f}_i^2,$$

as desired. ∎

5 Universal Tester

We have arrived at the final section of this note, where we use our newly developed toolkit for analyzing Boolean function to analyze the universal tester introduced in Section 1. We remind the reader that the following theorem is a critical component of the proof of the PCP theorem by Dinur [2], as discussed in Section 1.

Theorem 1.1. *Let $L \subseteq \{0, 1\}^n$. Let $p = 2^{2^n}$. Then there is a randomized algorithm $A_L : \{0, 1\}^n \times \{0, 1\}^p \to \{0, 1\}$ that, given oracle access to $x \in \{0, 1\}^n$ and $y \in \{0, 1\}^p$, has the following properties.*

1. *A_L makes 3 queries to bits of x or y.*
2. *If $x \in L$, then there exists $y \in \{0, 1\}^p$ such that $A_L(x, y) = 1$ always.*

3. If $x \notin L$ then for all $y \in \{0,1\}^p$, we have

$$\mathbf{P}[A_L(x,y) = 0] \geq .001 \min_{y \in L} \frac{\|x - y\|_0}{n}.$$

Proof. We briefly described the algorithm in Section 1 but let us describe it anew and more precisely. Let $N = 2^n$. Identifying $\{0,1\}^n \equiv [N]$, we identify L as a subset of $[N]$. Consider the subclass of dictator functions on N bits,

$$D_L = \left\{ \chi_w : \{0,1\}^N \to \{0,1\} \,\middle|\, w \in L \right\}.$$

Alternatively, given the subclass of dictators D_L, we have a language $L \subset \{0,1\}^n$ where $x \in L$ iff $\chi_x \in D_L$. The advantage of interpreting x as the index of a dictator χ_x, and interpreting L as the subclass of dictators D_L, is that we have by now developed powerful tests for Boolean functions such as χ_x, and for families of dictators such as D_L.

Given $x \in \{0,1\}^n$, a *proof* for $x \in L$ will be the (encoding of the) xth dictator function $\chi_x : \{0,1\}^N \to \{-1,1\}$, as a length N bit string. Given input $x \in \{0,1\}^n$ and candidate proof $Y \in \{0,1\}^N$, we will test for two things.

1. We test that $Y \in D_L$, using the test from Theorem 4.4.
2. Given that $Y = \chi_w \in D_L$ for some coordinate $w \in L$, (somehow) test that $w = x$.

We need to specify how to do the second step. Given that $Y = \chi_w$ for some w, we would like to check that $w_j = x_j$ for a random coordinate $j \in [n]$. We can query x_j but we cannot directly query w_j. The dictator test in step 1 tells us that some w (probably) exists, but does not specify which w.

Fix a coordinate j. Recall the linear correction protocol from Section 3.2. Insofar as Y is close to χ_w, we can probabilistically query $\chi_w(Z)$ for our choice of input $Z \in \{0,1\}^N$. To retrieve w_j (without knowing w), we need to define an input $Z \in \{0,1\}^N$ such that $\chi_w(Z) = w_j$ for all w. To that end, we define a string $Z_j \in \{0,1\}^N$ by

$$Z_j(w) = w_j.$$

For input $Z \in \{0,1\}^n$, let $H(Y,Z) = Y(Z + A)Y(A)$ (where $A \sim \{0,1\}^N$) denote the (random) output of running the local correction procedure on Y with input Z. We reject Y unless $x_j = H(Y, Z_j)$. If we inline the correction protocol of Section 3.2, then step 2 can be written out explicitly as follows.

2. Sample $j \in [n]$ uniformly at random and sample $A \in \{0,1\}^N$ uniformly at random. Define Z_j by $Z_j(w) = w_j$ for $w \in \{0,1\}^n$. Reject x unless $x_j = Y(Z_j + A)Y(A)$.

Having now established the testing algorithm in full, fix an input $x \in \{0,1\}^n$ and $Y \in \{0,1\}^N$, and suppose the tester accepts with probability of error ϵ for $\epsilon > 0$. This means in particular that (x, Y) would pass either of the two tests alone with probability of error $\leq 2\epsilon$. We claim that x is $O(\epsilon)$-close to some point in L.

Because Y passes the first test with probability of error $\leq 2\epsilon$, we have that Y is $(C_1\epsilon)$-close to some dictator function $\chi_w \in D_L$, for some universal constant C_1. Since Y is $(C_1\epsilon)$-close to χ_w for some w, for any input Z,

$$\mathbf{P}[H(Y,Z) \neq \chi_w(Z)] \leq C_2\epsilon \tag{2}$$

for another universal constant $C_2 > 0$. Consider now the inputs Z_j that are constructed as a function of the randomly selected coordinate j. We have

$$\mathop{\mathbf{P}}_{j}\left[x_j \neq w_j\right] \overset{(a)}{\leq} \mathop{\mathbf{P}}_{j,H}\left[H(Y,Z_j) \neq w_j\right] + \mathop{\mathbf{P}}_{j,H}\left[H(Y,Z_j) \neq x_j\right] \overset{(b)}{\leq} C_2\epsilon + 2\epsilon.$$

(a) is by the union bound. The first term in (b) is by (2) and the second term is because (x, Y) passes the second test with probability of error $\leq 2\epsilon$. Thus for a universal constant $C_3 = C_2 + \epsilon$, x is $(C_3\epsilon)$-close to some $w \in L$. ■

References

[1] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. "Self-Testing/Correcting with Applications to Numerical Problems". In: *J. Comput. Syst. Sci.* 47.3 (1993), pp. 549–595. Preliminary version in STOC, 1990.

[2] Irit Dinur. "The PCP theorem by gap amplification". In: *J. ACM* 54.3 (2007), p. 12. Preliminary version in STOC, 2006.

[3] Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014. URL: http://www.cambridge.org/de/academic/subjects/computer-science/algorithmics-complexity-computer-algebra-and-computational-g/analysis-boolean-functions.