Notes on Complexity Theory Last updated: June, 2014

Lecture 22

Jonathan Katz

1 $\mathcal{NP} \subseteq \mathsf{PCP}(\mathsf{poly}, 1)$

We show here a probabilistically checkable proof for $\mathcal{NP}$ in which the verifier reads only a *constant* number of bits from the proof (and uses only polynomially many random bits). In addition to being of independent interest, this result is used as a key step in the proof of the PCP theorem itself.

To show the desired result, we will work with the $\mathcal{NP}$-complete language of *satisfiable quadratic equations*. Instances of this problem consist of a system of m quadratic equations

$$\left\{ \sum_{i,j=1}^{n} c_{i,j}^{(k)} \cdot x_i \, x_j = c^{(k)} \right\}_{k=1}^{m} \tag{1}$$

(over the field $\mathbb{F}_2$) in the n variables $x_1, \ldots, x_n$. (Note that we can assume no linear terms since $x_i = x_i \cdot x_i$ in $\mathbb{F}_2$ and the summations above include the case $i = j$.) A system of the above form is said to be *satisfiable* if there is an assignment to the $\{x_i\}$ for which every equation is satisfied.

It is obvious that this problem is in $\mathcal{NP}$. To show that it is $\mathcal{NP}$-complete we reduce an instance of 3SAT to an instance of the above. Given a 3SAT formula ϕ on n variables, using arithmetization we can express each of its clauses as a cubic equation. (One way to do this is as follows: arithmetize the literal x_j by the term $1 - x_j$ and the literal $\bar{x}_j$ by the term x_j; a clause $\ell_1 \vee \ell_2 \vee \ell_3$ is arithmetized by the product of the arithmetization of its literals. Then ask whether there is an assignment under which the arithmetization of all the clauses of ϕ equal 0.) To reduce the degree to quadratic, we introduce the "dummy" variables $\{x_{i,j}\}_{i,j=1}^{n}$ and then: (1) replace monomials of the form $x_i x_j x_k$ with a monomial of the form $x_{i,j} x_k$, and (2) introduce n^2 new equations of the form $x_{i,j} - x_i x_j = 0$.

We remark that there is no hope of reducing the degree further (unless $\mathcal{P} = \mathcal{NP}$) since a system of linear equations can be solved using standard linear algebra.

1.1 A PCP for Satisfiable Quadratic Equations: An Overview

For the remainder of these notes, we will assume a system of m equations in the n variables $\{x_i\}$, as in Eq. (1). The proof string π will be a boolean string $\pi \in \{0,1\}^{2^{n^2}}$ that we index by a binary vector $\vec{v}$ of length n^2. Equivalently, we will view π as a function $\pi : \{0,1\}^{n^2} \to \{0,1\}$. For a given system of satisfiable quadratic equations, π should be such that

$$\pi(\vec{v}) \stackrel{\text{def}}{=} \sum_{i,j=1}^{n} a_i a_j v_{i,j}$$

for some satisfying assignment $(a_1, \ldots, a_n)$, where $\vec{v} = (v_{1,1}, \ldots, v_{1,n}, \ldots, v_{n,1}, \ldots, v_{n,n})$ Note that with $(a_1, \ldots, a_n)$ fixed, π is a linear function of $\vec{v}$; i.e., π is just the dot product of the input with the fixed string $(a_{1,1}, \ldots, a_{n,n})$.

Roughly speaking, given access to a proof string π we will have the verifier check three things: (1) that the proof string encodes a linear function; i.e.,

$$\pi(\vec{v}) = \sum_{i,j=1}^{n} \lambda_{i,j}\, v_{i,j}$$

for some $\{\lambda_{i,j}\}$; (2) that the coefficients of the linear function encoded by the proof string are *consistent*; namely, that $\lambda_{i,j} = \lambda_{i,i} \cdot \lambda_{j,j}$ for all i, j; and (3) that the assignment defined by setting $a_i = \lambda_{i,i}$ is indeed a satisfying assignment. (Note that these all hold for a "good" proof π when the system of equations is satisfiable.) Because the verifier is restricted to making a very small number of queries, the verifier will be unable to verify any of the above with certainty, but it will be able to verify these conditions probabilistically. In these notes, we focus only on achieving a constant probability of rejection when the system of equations is unsatisfiable; we aim for the simplest proof and make no attempt to optimize the constants. Of course, by using a constant number of independent repetitions we can then reduce the error probability to $1/2$ (while still reading only a constant number of bits from π and using polynomially many random bits).

We discuss in turn the tests used to verify each of the properties above, and then show how they can be combined to yield the desired PCP system.

2 The Linearity Test

A function $f : \{0,1\}^N \to \{0,1\}$ is linear if there exists an $r \in \{0,1\}^N$ such that $f(x) = \langle x, r \rangle$, i.e.,

$$f(x_1 \cdots x_N) = \sum_{i=1}^{N} r_i \cdot x_i \,.$$

In this section we show how to test whether a function $\pi : \{0,1\}^N \to \{0,1\}$ is (close to) linear.

Let us first define a notion of distance for functions. Two functions $f, g : \{0,1\}^N \to \{0,1\}$ have distance δ if they disagree on a δ fraction of their inputs; that is, if $\Pr_x[f(x) \neq g(x)] = \delta$ (where x is chosen uniformly from $\{0,1\}^N$). Viewing a boolean function over $\{0,1\}^N$ as a binary string of length 2^N, two functions have distance δ if their Hamming distance is $\delta \cdot 2^N$. We say a function f is distance at least δ from linear if for all linear functions g the distance between f and g is at least δ, and define "distance δ from linear" and "distance at most δ from linear" similarly.

The following test allows a verifier, given access to π, to check whether π is "close" to linear:

- Choose random $x, y \in \{0,1\}^N$.
- Query $\pi(x), \pi(y)$, and $\pi(x+y)$.
- Accept if and only if $\pi(x) + \pi(y) = \pi(x+y)$, where addition is in $\mathbb{F}_2$ component-wise.

Note that if π is linear, then the verifier always accepts since

$$\begin{aligned} \pi(x+y) &= \sum_{i=1}^{N} r_i \cdot (x_i + y_i) \\ &= \left(\sum_{i=1}^{N} r_i\, x_i\right) + \left(\sum_{i=1}^{N} r_i\, y_i\right) = \pi(x) + \pi(y). \end{aligned}$$

The interesting part is to show that when π is "far" from linear then the verifier rejects with high probability. In the following section we prove:

Theorem 1 *If π has distance ε from linear, the linearity test rejects with probability at least ε.*

Of course, by repeating the test a constant number of times we can increase the rejection probability to any constant less than 1.

In the next section we give a proof of Theorem 1 based on Fourier analysis. (A proof that does not use Fourier analysis is also possible but, conceptually speaking, Fourier analysis is exactly the right tool for this setting.) The proof is unnecessary for understanding the rest of the PCP construction, and so the reader willing to take Theorem 1 on faith can skip directly to Section 3.

2.1 Proof of Theorem 1

The first thing we will do is view π as a function from $\{-1,1\}^N$ to $\{-1,1\}$, by mapping each bit b of the input and output to the value $(-1)^b$. Given this notational switch, the linearity test chooses random $x, y \in \{-1,1\}^N$, and accepts if and only if $\pi(x) \cdot \pi(y) \cdot \pi(x \circ y) = 1$, where "$\circ$" denotes component-wise product.

The proof relies on some basic Fourier analysis; we provide some background first. View the set of functions from $\{-1,1\}^N$ to the reals as a vector space (over the reals). This is a vector space of dimension 2^N, with one basis given by the functions $\{I_v : \{-1,1\}^N \to \mathbb{R}\}_{v \in \{-1,1\}^N}$ where

$$I_v(v') \stackrel{\text{def}}{=} \begin{cases} 1 & v' = v \\ 0 & \text{otherwise} \end{cases}.$$

To confirm that this is a basis, note that any function $\pi : \{-1,1\}^N \to \mathbb{R}$ can be expressed as:

$$\pi = \sum_{v \in \{-1,1\}^N} \pi(v) \cdot I_v.$$

We will also define an inner product $\langle \cdot, \cdot \rangle$ on this vector space, via:

$$\langle f, g \rangle \stackrel{\text{def}}{=} \frac{1}{2^N} \cdot \sum_v f(v) \cdot g(v) = \mathbf{Exp}_v[f(v) \cdot g(v)].$$

(Note that this inner product is bilinear.) We see that the basis given above is orthogonal.

The basis described above is the "standard" one. In our context, however, there is another basis that works even better: the Fourier basis $\{\chi_v\}_{S \subseteq [N]}$ where

$$\chi_S(v') = \prod_{i \in S} v'_i$$

(with the empty product when $S = \emptyset$ interpreted as a '1'). One can check that these functions are orthogonal and hence, since there are 2^N such functions, this is indeed a basis; in fact it is an *orthonormal* basis. This means that we can write any function $f : \{-1,1\} \to \mathbb{R}$ as

$$f = \sum_{S \subseteq [N]} \hat{f}(S) \cdot \chi_S$$

with $\hat{f}(S) \in \mathbb{R}$; by orthonormality, we have

$$\langle f, \chi_S \rangle = \left\langle \sum_{S \subseteq [N]} \hat{f}(S) \cdot \chi_S,\ \chi_S \right\rangle = \hat{f}(S).$$

The first hint that the Fourier basis might be useful for our purposes is the following observation: If f, g are functions from $\{-1,1\}^N$ to $\{-1,1\}$, then

$$\langle f, g \rangle = \frac{1}{2^N} \cdot \Big(\big|\{x \mid f(x) = g(x)\}\big| - \big|\{x \mid f(x) \neq g(x)\}\big| \Big) = 1 - 2 \cdot \Pr_x[f(x) \neq g(x)];$$

in other words, if f is distance δ from g, then $\langle f, g \rangle = 1 - 2\delta$. Note that each χ_S is a linear function (except that everything has been translated from $\{0,1\}$ to $\{-1,1\}$), and so the Fourier basis includes *all* linear functions. Thus, *to find the linear function closest to π we simply need to find S for which $\langle \chi_S, \pi \rangle$ is maximized.* Furthermore, *π is far from linear if and only if $\langle \chi_S, \pi \rangle$ is small for* **all** S. We will use this in the proof below.

Before turning to the proof of the linearity test, we state two facts that follow from standard linear algebra.

- The inner product $\langle f, g \rangle$ of any two functions f, g is given by the sum of the product of the coefficients of f and g in any orthonormal basis. Thus, in particular, we have

  $$\langle f, g \rangle = \sum_{S \subseteq N} \hat{f}(S) \cdot \hat{g}(S)\,.$$

  This is known as *Plancherel's theorem.*

- It follows from the above that $\langle f, f \rangle = \sum_S \hat{f}(S)^2$. If the range of f is $\{-1,1\}$, then (by definition of the inner product)

  $$\langle f, f \rangle = \frac{1}{2^N} \sum_v f(v)^2 = 1.$$

  We thus conclude that when f maps onto $\{-1,1\}$, we have $\sum_S \hat{f}(S)^2 = 1$. This is known as *Parseval's theorem.*

We can now prove the following result:

Theorem 2 *If π has distance ε from linear, the linearity test rejects with probability at least ε.*

We begin with a lemma. Amazingly, the lemma is pretty powerful although its proof involves nothing more than grinding through some algebraic manipulations.

Lemma 3 Pr[linearity test accepts π] $= \frac{1}{2} + \frac{1}{2} \cdot \sum_S \hat{\pi}(S)^3$.

Proof In our notation, the linearity test chooses random $x, y \in \{-1,1\}^N$, and accepts iff $\pi(x) \cdot \pi(y) \cdot \pi(x \circ y) = 1$. Since π is boolean (so has range $\{-1,1\}$), we have $\pi(x) \cdot \pi(y) \cdot \pi(x \circ y) \in \{-1,1\}$.

So, $I \stackrel{\text{def}}{=} \frac{1}{2} + \frac{1}{2}\pi(x) \cdot \pi(y) \cdot \pi(x \circ y)$ is an indicator random variable for the event that the linearity test accepts. Thus:

$$\begin{aligned} \Pr[\text{linearity test accepts}] &= \mathbf{Exp}_{x,y}[I] \\ &= \frac{1}{2} + \frac{1}{2} \cdot \mathbf{Exp}_{x,y}[\pi(x) \cdot \pi(y) \cdot \pi(x \circ y)]. \end{aligned} \tag{2}$$

Expanding π in terms of its Fourier coefficients gives

$$\begin{aligned} &\mathbf{Exp}_{x,y}[\pi(x) \cdot \pi(y) \cdot \pi(x \circ y)] = \\ &\quad \mathbf{Exp}_{x,y}\left[\left(\sum_S \hat{\pi}(S)\,\chi_S(x)\right) \cdot \left(\sum_{S'} \hat{\pi}(S')\,\chi_{S'}(y)\right) \cdot \left(\sum_{S''} \hat{\pi}(S'')\,\chi_{S''}(x \circ y)\right)\right] \\ &= \mathbf{Exp}_{x,y}\left[\sum_{S,S',S''} \hat{\pi}(S)\,\hat{\pi}(S')\,\hat{\pi}(S'')\,\chi_S(x)\,\chi_{S'}(y)\,\chi_{S''}(x \circ y)\right] \\ &= \sum_{S,S',S''} \hat{\pi}(S)\,\hat{\pi}(S')\,\hat{\pi}(S'') \cdot \mathbf{Exp}_{x,y}\left[\chi_S(x)\,\chi_{S'}(y)\,\chi_{S''}(x \circ y)\right]. \end{aligned} \tag{3}$$

By definition of χ_S, for any fixed S, S', S'' we have:

$$\begin{aligned} \mathbf{Exp}_{x,y}\left[\chi_S(x)\,\chi_{S'}(y)\,\chi_{S''}(x \circ y)\right] &= \mathbf{Exp}_{x,y}\left[\prod_{i \in S} x_i \cdot \prod_{i \in S'} y_i \cdot \prod_{i \in S''} x_i y_i\right] \\ &= \mathbf{Exp}_{x,y}\left[\prod_{i \in S \triangle S''} x_i \cdot \prod_{i \in S' \triangle S''} y_i\right] \\ &= \mathbf{Exp}_x\left[\prod_{i \in S \triangle S''} x_i\right] \cdot \mathbf{Exp}_y\left[\prod_{i \in S' \triangle S''} y_i\right], \end{aligned} \tag{4}$$

where $A \triangle B$ denotes the symmetric difference between sets A and B. (I.e., $i \in A \triangle B$ iff i is in exactly one of A or B.) Above, the second equality uses the fact that $x_i^2 = y_i^2 = 1$ (since $x_i, y_i \in \{-1, 1\}$), and the third equality relies on the fact that x and y are independent.

Evaluating $\mathbf{Exp}_x\left[\prod_{i \in S \triangle S''} x_i\right]$ is easy: if $S = S''$ then the product is empty and so evaluates to 1 regardless of x. On the other hand, if $S \neq S''$ then each x_i is equally likely to be 1 or -1 and so the expected value of the product is 0. We thus see from Equation (4) that $\mathbf{Exp}_{x,y}\left[\chi_S(x)\,\chi_{S'}(y)\,\chi_{S''}(x \circ y)\right] = 0$ unless $S = S' = S''$, in which case the expression evaluates to 1. Working back through Equations (3) and (2) gives the claimed result. ■

Given Lemma 3 we can prove Theorem 2 in just a few lines. We have:

$$\begin{aligned} \Pr[\text{linearity test accepts}] &= \frac{1}{2} + \frac{1}{2} \cdot \sum_S \hat{\pi}(S)^3 \\ &\leq \frac{1}{2} + \frac{1}{2} \cdot \max_S\{\hat{\pi}(S)\} \cdot \sum_S \hat{\pi}(S)^2 \\ &= \frac{1}{2} + \frac{1}{2} \cdot \max_S\{\hat{\pi}(S)\}, \end{aligned}$$

using Parseval's theorem. If π is distance ε from linear, this means that $\max_v\{\hat{\pi}(v)\} = 1 - 2\varepsilon$. We conclude that Pr[linearity test accepts] $\leq 1 - \varepsilon$, proving the theorem.

3 The Consistency Test

We return to our notation from Section 1.1, where $\vec{v}$ represents a boolean vector of length n^2 and we index it using two indices each ranging from 1 to n.

Assume π is within distance 1/48 from linear. This means there exists a unique[1] linear function f within distance 1/48 from π; we can write f as

$$f(\vec{v}) = \sum_{i,j=1}^{n} \lambda_{i,j} \cdot v_{i,j}$$

for some $\{\lambda_{i,j} \in \{0,1\}\}$. We now want a way to check that these $\{\lambda_{i,j}\}$ are *consistent*; i.e., that $\lambda_{i,j} = \lambda_{i,i} \cdot \lambda_{j,j}$ for all i, j. A useful way to view this is to put the $\{\lambda_{i,j}\}$ in an $n \times n$ matrix M; i.e.,

$$M \stackrel{\text{def}}{=} \begin{pmatrix} \lambda_{1,1} & \lambda_{1,2} & \cdots & \lambda_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{n,1} & \lambda_{n,2} & \cdots & \lambda_{n,n} \end{pmatrix}.$$

Let $\vec{\lambda} \stackrel{\text{def}}{=} (\lambda_{1,1}, \ldots, \lambda_{n,n})$ (all our vectors will be row vectors). Then consistency is equivalent to:

$$M = \vec{\lambda}^T \vec{\lambda}.$$

(Note once again that $\lambda_{i,i}^2 = \lambda_{i,i}$ since we are working in $\mathbb{F}_2$.) We first show an efficient way to test equality of matrices, and then show how the test can be implemented using access to π.

Claim 4 *Let M, M' be two unequal $n \times n$ matrices over $\mathbb{F}_2$. Then*

$$\Pr_{\vec{x},\vec{y} \in \{0,1\}^n}[\vec{x}M\vec{y}^T = \vec{x}M'\vec{y}^T] \leq \frac{3}{4}.$$

Proof Note that $\vec{x}M\vec{y}^T - \vec{x}M'\vec{y}^T = \vec{x}(M - M')\vec{y}^T$ and $M - M'$ is a non-zero matrix. So we are interested in the probability that $\vec{x}M''\vec{y}^T = 0$ for non-zero matrix M''.

The probability that $M''\vec{y}^T = \vec{0}$ is at most 1/2. Assuming this does not occur, the probability that $\vec{x}(M''\vec{y}^T) = 0$ is exactly 1/2. So, the probability that $\vec{x}M''\vec{y}^T = 0$ is at most 3/4. ■

How can we evaluate $\vec{x}M\vec{y}^T$ and $\vec{x}(\vec{\lambda}^T\vec{\lambda})\vec{y}^T$ given access to π? Let us assume we have access to f, and show how to correct for this later. Given access to f, it is easy to compute $\vec{x}M\vec{y}^T$ since

$$\vec{x}M\vec{y}^T = \sum_{i,j=1}^{n} \lambda_{i,j} x_i y_j .$$

Setting $v_{i,j} = x_i y_j$ and querying $f(\vec{v})$ thus gives the desired answer. For the second computation, note that

$$\vec{x}(\vec{\lambda}^T\vec{\lambda})\vec{y}^T = (\vec{x}\vec{\lambda}^T)(\vec{\lambda}\vec{y}^T).$$

[1] That f is unique follows from the fact that any two distinct linear functions are distance 1/2 from each other.

Setting $v_{i,i} = x_i$ (and $v_{i,j} = 0$ when $i \neq j$), we see that $f(\vec{v}) = \vec{x}\vec{\lambda}^T$; the value $\vec{\lambda}\vec{y}^T$ is computed similarly.

The above assumes we have access to f — but we only have access to π! However, we said that π was within distance $1/48$ from f. So we can compute $f(\vec{v})$ (for any $\vec{v}$) by choosing a random "shift" $\vec{r} \in \{0,1\}^{n^2}$ and computing $\tilde{f}(\vec{v}) = \pi(\vec{r}) + \pi(\vec{r} + \vec{v})$. Note that as long as $\pi(\vec{r}) = f(\vec{r})$ and $\pi(\vec{r} + \vec{v}) = f(\vec{r} + \vec{r})$, then $\hat{f}(\vec{v}) = f(\vec{v})$. Thus, for any $\vec{v}$ we compute the correct value of $f(\vec{v})$ except with probability $2/48 = 1/24$. This technique is called *self-correction*.

3.1 In Summary

To summarize, we perform the following *consistency test*:

1. Choose random $\vec{x}, \vec{y}$.
2. Using self-correction and the approach described above, compute $\vec{x}M\vec{y}^T$.
3. Using self-correction and the approach described above, compute $\vec{x}(\vec{\lambda}^T\vec{\lambda})\vec{y}^T$.
4. Accept if and only if the two values thus computed are equal.

Note that step 2 requires one call to f, so it requires two calls to π. Step 3 requires two calls to f, so it requires four calls to π.

We may now state the main result of this section (again, we have not tried to optimize constants):

Theorem 5 *Assume π is within distance $1/48$ of a linear function f. If f is not consistent (in the sense described above), then the consistency test rejects with probability at least $1/8$.*

Proof Assuming the test correctly computes $\vec{x}M\vec{y}^T$ and $\vec{x}(\vec{\lambda}^T\vec{\lambda})\vec{y}^T$, the test will accept with probability at most $3/4$ (by Claim 4). The probability that one of the six calls to π results in an incorrect value for f is at most $6/48 = 1/8$ (using the fact that π and f disagree on at most a $1/48$ fraction of their points, and applying a union bound). So, the probability of acceptance is at most $3/4 + 1/8$ and the theorem follows. ■

4 The Satisfiability Test

Assume π is within distance $1/48$ from the linear function

$$f(\vec{v}) = \sum_{i,j=1}^{n} \lambda_{i,j} v_{i,j}$$

and furthermore that f is consistent (i.e., the $\{\lambda_{i,j}\}$ satisfy $\lambda_{i,j} = \lambda_{i,i} \cdot \lambda_{j,j}$ for all i, j). We view π an encoding an assignment $\vec{a} = (\lambda_{1,1}, \lambda_{2,2}, \ldots, \lambda_{n,n})$. We now want to check that this assignment is a satisfying assignment for the given system of equations. (Indeed, note that until this point everything we have done has been independent of the system of equations whose satisfiability we are interested in!)

Our set of equations (cf. Eq (1)) can be written as:

$$\left\{ c^{(k)} + \sum_{i,j=1}^{n} c_{i,j}^{(k)} \cdot x_i x_j = 0 \right\}_{k=1}^{m},$$

and so we want to verify whether

$$y_k \stackrel{\text{def}}{=} c^{(k)} + \sum_{i,j=1}^{n} c_{i,j}^{(k)} \cdot a_i a_j$$

is equal to 0 for all $k \in [1, m]$. If we let $\vec{y} \stackrel{\text{def}}{=} (y_1, \ldots, y_m)$, then we want to check whether $\vec{y}$ is the 0-vector. We can't check every position individually since this will require too many queries to π. What we will do instead is to look at the dot product of $\vec{y}$ with a random vector: if $\vec{y} = \vec{0}$ then this dot product will always be 0, but if $\vec{y} \neq 0$ then the dot product will be 1 with probability 1/2.

Taking the dot product of $\vec{y}$ with a random vector is equivalent to choosing a random subset $S \subseteq [m]$ and looking at the sum. That is,

$$\begin{aligned}
\sum_{k \in S} y_k &= \sum_{k \in S} \left(c^{(k)} + \sum_{i,j=1}^{n} c_{i,j}^{(k)} \cdot a_i a_j \right) \\
&= \sum_{k \in S} c^{(k)} + \sum_{k \in S} \sum_{i,j=1}^{n} c_{i,j}^{(k)} \cdot a_i a_j \\
&= \sum_{k \in S} c^{(k)} + \sum_{i,j=1}^{n} a_i a_j \cdot \left(\sum_{k \in S} c_{i,j}^{(k)} \right).
\end{aligned}$$

We can evaluate the first term on our own, and will obtain the second term by evaluating $f(\vec{v})$ where

$$v_{i,j} = \sum_{k \in S} c_{i,j}^{(k)}.$$

To obtain this value $f(\vec{v})$, we will again use self-correction as in the previous section.

In total, we make two queries to π and achieve the following (again, constants have not been optimized):

Theorem 6 *Assume π is within distance 1/48 of a linear function f and that f is consistent (as defined previously). Then if the system of equations is not satisfiable, the satisfiability test rejects with probability at least 1/8.*

Proof If the test correctly computes $f(\vec{v})$, it accepts with probability 1/2. The probability that one of the two calls to π results in an incorrect value for f is at most 2/48 (as in the previous theorem). So, the probability of acceptance is at most $1/2 + 2/48$ and the theorem follows. ■

5 Putting it all Together

We summarize the PCP system. Given a system of equations and access to an oracle π, the verifier proceeds as follows:

- Perform the linearity test (3 queries to π).
- Perform the consistency test (6 queries to π).
- Perform the satisfiability test (2 queries to π).
- Accept only if all the above tests succeed.

If the system of equations is satisfiable, then there exists a proof string π for which the above test accepts with probability 1. We claim that if the system of equations is *not* satisfiable, then the test will reject with probability at least 1/48 (for any π). There are three cases: (1) π is not within distance 1/48 of a linear function; (2) π is within distance 1/48 of a (unique) linear function f, but f is not consistent; or (3) π is within distance 1/48 of a consistent, linear function f, but f does not encode a satisfying assignment. In case (1) the linearity test will reject with probability at least 1/48; in case (2) the consistency test will reject with probability at least 1/8; and in case (3) the satisfiability test will reject with probability at least 1/8.

Bibliographic Notes

For more on Fourier analysis of boolean functions, see O'Donnell's lecture notes [2], or the online textbook he is currently writing [3]. For a reasonably self-contained proof of the full PCP theorem (including also a proof that $\mathcal{NP} \subseteq \mathsf{PCP}(\mathsf{poly}, 1)$), see Harsha's thesis [1].

References

[1] P. Harsha. Robust PCPs of Proximity and Shorter PCPs. PhD thesis, MIT, 2004. Available at `http://www.tcs.tifr.res.in/~prahladh/papers/#theses`

[2] R. O'Donnell. Lecture notes for 15-859S: Analysis of Boolean Functions. Available at `http://www.cs.cmu.edu/~odonnell/boolean-analysis`.

[3] R. O'Donnell. `http://www.contrib.andrew.cmu.edu/~ryanod/?page_id=2`