Notes on Complexity Theory Last updated: February, 2008

Handout 13

Jonathan Katz

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

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 a polynomial amount of randomness). The proof of this result will show how it is possible to (probabilistically) verify an "$\mathcal{NP}$ witness" by reading only a constant number of bits of the witness. In addition, this result is used as a key step of 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. (In more detail: 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 each of the clauses of ϕ is equal to 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 the 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 $\mathcal{NP} = \mathcal{P}$) since a system of linear equations can be solved using standard linear-algebraic techniques.

2 The PCP for Satisfiable Quadratic Equations: An Overview

For the remainder of these notes, we will assume a system of m equations as in Eq. (1), in the n variables $\{x_i\}$. We will let $(a_1, \ldots, a_n)$ denote a boolean assignment to these variables. For a given system of satisfiable quadratic equations, the entries of the proof string π will be indexed by a binary vector $\vec{v}$ of length n^2 (and so $|\pi| = 2^{n^2}$), with the intention that an honest prover will choose $(a_1, \ldots, a_n)$ to be a satisfying assignment and then set entry $\vec{v} = (v_{1,1}, \ldots, v_{1,n}, \ldots, v_{n,1}, \ldots, v_{n,n})$ of π equal to:

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

(A dishonest prover can do whatever he likes and, of course, in this case the system of equations may not be satisfiable.) Note that with $(a_1, \ldots, a_n)$ fixed, the above is a linear function of $\vec{v}$; i.e., it just computes 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 (which we may also view as a function $\pi : \{0,1\}^{n^2} \rightarrow \{0,1\}$) 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 are all the case 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).

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

3 The Linearity Test

A function $f : \{0,1\}^N \rightarrow \{0,1\}$ 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 $\pi : \{0,1\}^N \rightarrow \{0,1\}$ is (close to) linear.

Let us first define a notion of distance for functions. Two functions $f, g : \{0,1\}^N \rightarrow \{0,1\}$ have distance δ if they disagree on a δ fraction of their points; 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" and "distance at most δ" similarly.)

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

- Choose random $v^{(1)}, v^{(2)} \in \{0,1\}^N$.
- Query $\pi(v^{(1)}), \pi(v^{(2)})$, and $\pi(v^{(1)} + v^{(2)})$.
- Accept if and only if $\pi(v^{(1)}) + \pi(v^{(2)}) = \pi(v^{(1)} + v^{(2)})$.

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

$$\pi(v^{(1)} + v^{(2)}) \;\; = \;\; \sum_{i=1}^{N} r_i \cdot (v_i^{(1)} + v_i^{(2)})$$

$$
\begin{aligned}
&= \left(\sum_{i=1}^{N} r_i\, v_i^{(1)}\right) + \left(\sum_{i=1}^{N} r_i\, v_i^{(2)}\right) \\
&= \pi(v^{(1)}) + \pi(v^{(2)}).
\end{aligned}
$$

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

The following sections give two proofs of Theorem 1: the first is messy but totally self-contained; the second is beautiful but relies on Fourier analysis. (The necessary Fourier analysis is also explained, but it relies on some basic linear algebra.) Actually, the first proof yields a slightly weaker result (but the overall analysis of the PCP construction could easily be adapted to work with it). Neither of the proofs are necessary for understanding the PCP construction, and so the reader willing to take Theorem 1 on faith can skip directly to Section 4.

3.1 First Proof

Let π be distance ε from linear. Note that we must have $\varepsilon \leq 1/2$. (Prove it![1]) We prove the following weaker version of Theorem 1:

Theorem 2 *If π has distance ε from linear, the linearity test rejects with probability at least $\varepsilon/4$.*

We divide the analysis into two cases:

Case 1: Say $\varepsilon < 3/8$. Let f be a linear function at distance ε from π. Let G be the set of points on which f and π agree; we know that $|G| = (1-\varepsilon)\cdot 2^N$. Now, if exactly two of the points $v^{(1)}, v^{(2)}$, and $v^{(1)} + v^{(2)}$ lie in G, then the above procedure will reject. So we can lower-bound the probability of rejection in this case by the probability that this occurs; i.e.,

$$
\begin{aligned}
\Pr[\mathsf{reject}] &\geq \Pr[\text{exactly two of the points lie in } G] \\
&= 3 \cdot \Pr[v^{(1)} \notin G \bigwedge v^{(2)}, v^{(1)} + v^{(2)} \in G] \qquad \text{(by symmetry)} \\
&= 3 \cdot \Pr[v^{(1)} \notin G] \cdot \Pr[v^{(2)}, v^{(1)} + v^{(2)} \in G \mid v^{(1)} \notin G].
\end{aligned}
$$

Now,

$$
\begin{aligned}
&\Pr[v^{(2)}, v^{(1)} + v^{(2)} \in G \mid v^{(1)} \notin G] \\
&\quad = 1 - \Pr[v^{(2)} \notin G \bigvee v^{(1)} + v^{(2)} \notin G \mid v^{(1)} \notin G] \\
&\quad \geq 1 - \Pr[v^{(2)} \notin G \mid v^{(1)} \notin G] - \Pr[v^{(1)} + v^{(2)} \notin G \mid v^{(1)} \notin G] \\
&\quad = 1 - 2\varepsilon,
\end{aligned}
$$

[1] For any function π, the expected number of points on which a random linear function agrees with π is at least $\frac{2^N-1}{2}$. So there must exist a linear function that agrees with π on at least this many points (and the number of points of agreement must be an integer).

applying a union bound for the inequality and using the fact that $v^{(2)}$ is chosen independently of $v^{(1)}$ in the last step. Putting everything together we obtain

$$\Pr[\mathsf{reject}] \geq 3 \cdot \varepsilon \cdot (1 - 2\varepsilon),$$

which is at least $3\varepsilon/4$ for $\varepsilon < 3/8$ (as we are assuming here).

Case 2: Say $3/8 \leq \varepsilon \leq 1/2$. We are interested in

$$\tau \stackrel{\rm def}{=} \Pr[\mathsf{reject}] = \Pr_{v^{(1)},v^{(2)}}[\pi(v^{(1)}) + \pi(v^{(2)}) \neq \pi(v^{(1)} + v^{(2)})].$$

We show that if τ is "small" then in fact there exists a linear function L_π within distance less than $3/8$ of π. It follows that if $\varepsilon \geq 3/8$ then τ must be "large."

Define L_π as follows: for a string $v \in \{0,1\}^N$, define $L_\pi(v)$ to be the boolean value b for which

$$\Pr_{v^{(1)}}[\pi(v + v^{(1)}) - \pi(v^{(1)}) = b] \geq \frac{1}{2},$$

where ties are broken arbitrarily. (We remark that since we are working over $\mathbb{F}_2$ subtraction is the same as addition.)

We begin with a technical claim.

Claim 3 *For all v we have* $\Pr_{v^{(1)}}[\pi(v + v^{(1)}) - \pi(v^{(1)}) = L_\pi(v)] \geq 1 - 2\tau$.

Proof Fix v, and consider the probability, over random choice of $v^{(1)}, v^{(2)}$, that $\pi(v + v^{(1)}) - \pi(v^{(1)}) = \pi(v + v^{(2)}) - \pi(v^{(2)})$. Letting $p \stackrel{\rm def}{=} \Pr_{v^{(1)}}[\pi(v + v^{(1)}) - \pi(v^{(1)}) = L_\pi(v)]$ we have

$$\Pr_{v^{(1)},v^{(2)}}[\pi(v + v^{(1)}) - \pi(v^{(1)}) = \pi(v + v^{(2)}) - \pi(v^{(2)})] = p^2 + (1-p)^2$$

(since the differences are equal if they are either both equal to $L_\pi(v)$ or both not equal to $L_\pi(v)$). Evaluating this another way, we obtain:

$$\begin{aligned}
&\Pr_{v^{(1)},v^{(2)}}[\pi(v + v^{(1)}) - \pi(v^{(1)}) = \pi(v + v^{(2)}) - \pi(v^{(2)})] \\
&= \Pr_{v^{(1)},v^{(2)}}[\pi(v + v^{(1)}) + \pi(v^{(2)}) - \pi(v + v^{(1)} + v^{(2)}) = \pi(v + v^{(2)}) + \pi(v^{(1)}) - \pi(v + v^{(2)} + v^{(1)})] \\
&\geq \Pr_{v^{(1)},v^{(2)}}[\pi(v + v^{(1)}) + \pi(v^{(2)}) - \pi(v + v^{(1)} + v^{(2)}) = 0 \bigwedge \\
&\qquad \pi(v + v^{(2)}) + \pi(v^{(1)}) - \pi(v + v^{(2)} + v^{(1)}) = 0] \\
&= 1 - \Pr_{v^{(1)},v^{(2)}}[\pi(v + v^{(1)}) + \pi(v^{(2)}) - \pi(v + v^{(1)} + v^{(2)}) = 1 \bigvee \\
&\qquad \pi(v + v^{(2)}) + \pi(v^{(1)}) - \pi(v + v^{(2)} + v^{(1)}) = 1] \\
&\geq 1 - \Pr_{v^{(1)},v^{(2)}}[\pi(v + v^{(1)}) + \pi(v^{(2)}) \neq \pi(v + v^{(1)} + v^{(2)})] \\
&\qquad - \Pr_{v^{(1)},v^{(2)}}[\pi(v + v^{(2)}) + \pi(v^{(1)}) \neq \pi(v + v^{(2)} + v^{(1)})] \\
&= 1 - 2\tau,
\end{aligned}$$

using in the last step the fact that $v + v^{(1)}$ (resp., $v + v^{(2)}$) is uniformly distributed when $v^{(1)}$ (resp, $v^{(2)}$) is uniformly distributed. We conclude that $p^2 + (1-p)^2 \geq 1 - 2\tau$. Since $p \geq \frac{1}{2}$ by definition of L_π, and $p \geq p^2 + (1-p)^2$ for $p \in [\frac{1}{2}, 1]$, we have $p \geq 1 - 2\tau$ as desired. ■

Claim 4 *L_π is within distance 3τ of π.*

Proof We are interested in $\Pr_v[\pi(v) = L_\pi(v)]$. We have

$$\begin{aligned}
&\Pr_v[\pi(v) = L_\pi(v)] \\
&\geq \Pr_{v,v^{(1)}}[\pi(v) = \pi(v + v^{(1)}) - \pi(v^{(1)}) \bigwedge L_\pi(v) = \pi(v + v^{(1)}) - \pi(v^{(1)})] \\
&= 1 - \Pr_{v,v^{(1)}}[\pi(v) \neq \pi(v + v^{(1)}) - \pi(v^{(1)}) \bigvee L_\pi(v) \neq \pi(v + v^{(1)}) - \pi(v^{(1)})] \\
&\geq 1 - \Pr_{v,v^{(1)}}[\pi(v) \neq \pi(v + v^{(1)}) - \pi(v^{(1)})] - \Pr_{v,v^{(1)}}[L_\pi(v) \neq \pi(v + v^{(1)}) - \pi(v^{(1)})] \\
&\geq 1 - \tau - 2\tau.
\end{aligned}$$

The claim follows. ■

Finally, we show that if τ is "small" then L_π is linear. Combined with the previous claim, this shows that if τ is "small" then π is "close" to a linear function.

Claim 5 *If $\tau < 1/6$ then L_π is linear.*

Proof We show that for all $a, b \in \{0,1\}^N$ we have $L_\pi(a) + L_\pi(b) = L_\pi(a + b)$ (it is not too hard to show that this implies L_π is linear). Fix a, b arbitrarily. Suppose there exist $v^{(1)}, v^{(2)}$ such that

1. $L_\pi(a + b) = \pi(a + b + v^{(1)} + v^{(2)}) - \pi(v^{(1)} + v^{(2)})$;
2. $L_\pi(b) = \pi(a + b + v^{(1)} + v^{(2)}) - \pi(a + v^{(1)} + v^{(2)})$;
3. $L_\pi(a) = \pi(a + v^{(1)} + v^{(2)}) - \pi(v^{(1)} + v^{(2)})$.

Then we would have $L_\pi(a) + L_\pi(b) = L_\pi(a + b)$ as desired. We will show, using the probabilistic method, that there exist $v^{(1)}, v^{(2)}$ with the stated properties.

By Claim 3, each of the above events fails to occur with probability at most $2\tau < 1/3$. Applying a union bound, we see that the probability that at least one of the events does not occur is strictly less than 1. Thus, the probability that all the above events occur is strictly greater than 0 and hence L_π is linear. ■

Putting everything together we see that if π is distance $3/8$ (or more) from linear, then we must have $\tau \geq 1/8$; if not, then L_π is linear but π is within distance less than $3/8$ from L_π (a contradiction).

3.2 Second Proof

The second proof uses Fourier analysis. We introduce only as much background as needed.

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 "$\circ$" is used to denote a coordinate-wise product.

View the set of functions from $\{-1,1\}^N$ to the reals as a vector space (over the reals), in the natural way. This is a vector space of dimension 2^N, with one basis given by the functions $\{I_v\}_{v\in\{-1,1\}^N}$ 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 π 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:

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

We see that the basis given above is orthogonal.

The "standard" basis for the vector space of functions from $\{-1,1\}^N$ to the reals is the one just given. In our context, however, there is another basis that works even better: the Fourier basis. This basis is given by $\{\chi_v\}_{v\in\{-1,1\}^N}$ where

$$\chi_v(v') = \prod_{i\,:\,v_i=1} v'_i$$

(with the empty product interpreted as a '1'). Note that each χ_v is just a linear function (except that everything has been translated from $\{0,1\}$ to $\{-1,1\}$). One can check that these functions are all orthogonal (proving, since there are 2^N such functions, that this is indeed a basis) and in fact these functions give an *orthonormal* basis. For notational convenience (and following a standard convention in this area), we define $\hat{f}(v) \stackrel{\text{def}}{=} \langle f,\chi_v\rangle$. We then have

$$f = \sum_{v\in\{-1,1\}^N} \hat{f}(v)\cdot\chi_v\,.$$

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

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

in other words, if f is distance δ from g, then $\langle f,g\rangle = 1-2\delta$. This means that *to find the linear function closest to π, we simply need to find v for which $\langle\chi_v,\pi\rangle$ is maximized.* Furthermore, π *is far from linear if and only if $\langle\chi_v,\pi\rangle$ is small for all v.* We will use this in the proof below.

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

- If $\{f_i\}$ is an orthonormal basis, then 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 that basis. Specializing for the case of the orthonormal basis $\{\chi_v\}$ we obtain

  $$\langle f,g\rangle = \sum_{v\in\{-1,1\}^N} \hat{f}(v)\cdot\hat{g}(v)\,.$$

  This is known as *Plancherel's theorem.*

- It follows from the above that $\langle f, f\rangle = \sum_v \hat{f}(v)^2$. 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 $\sum_v \hat{f}(v)^2 = 1$. This is known as *Parseval's theorem*.

We now know enough to prove the following result:

Theorem 6 *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 7 Pr[linearity test accepts π] $= \frac{1}{2} + \frac{1}{2}\cdot\sum_v \hat{\pi}(v)^3$.

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:

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

Expanding π in terms of its Fourier coefficients gives

$$\begin{aligned}&\mathbf{Exp}_{x,y}[\pi(x)\cdot\pi(y)\cdot\pi(x\circ y)] = \\ &\qquad \mathbf{Exp}_{x,y}\left[\left(\sum_v \hat{\pi}(v)\,\chi_v(x)\right)\cdot\left(\sum_{v'}\hat{\pi}(v')\,\chi_{v'}(y)\right)\cdot\left(\sum_{v''}\hat{\pi}(v'')\,\chi_{v''}(x\circ y)\right)\right] \\ &= \mathbf{Exp}_{x,y}\left[\sum_{v,v',v''}\hat{\pi}(v)\,\hat{\pi}(v')\,\hat{\pi}(v'')\,\chi_v(x)\,\chi_{v'}(y)\,\chi_{v''}(x\circ y)\right] \\ &= \sum_{v,v',v''}\hat{\pi}(v)\,\hat{\pi}(v')\,\hat{\pi}(v'')\cdot\mathbf{Exp}_{x,y}\left[\chi_v(x)\,\chi_{v'}(y)\,\chi_{v''}(x\circ y)\right]. \qquad (3)\end{aligned}$$

By definition of χ_v, for any fixed v, v', v'' we have:

$$\begin{aligned}\mathbf{Exp}_{x,y}\left[\chi_v(x)\,\chi_{v'}(y)\,\chi_{v''}(x\circ y)\right] &= \mathbf{Exp}_{x,y}\left[\prod_{i\,:\,v_i=1} x_i \cdot \prod_{i\,:\,v'_i=1} y_i \cdot \prod_{i\,:\,v''_i=1} x_i y_i\right] \\ &= \mathbf{Exp}_{x,y}\left[\prod_{i\,:\,v_i\neq v''_i} x_i \cdot \prod_{i\,:\,v'_i\neq v''_i} y_i\right] \\ &= \mathbf{Exp}_x\left[\prod_{i\,:\,v_i\neq v''_i} x_i\right]\cdot\mathbf{Exp}_y\left[\prod_{i\,:\,v'_i\neq v''_i} y_i\right], \qquad (4)\end{aligned}$$

where 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 $\mathbf{Exp}_x\left[\prod_{i\,:\,v_i \neq v''_i} x_i\right]$ is easy: if $v = v''$ then the product is empty and thus evaluates to 1 regardless of x. On the other hand, if $v \neq v''$ 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 $\mathbf{Exp}_{x,y}\left[\chi_v(x)\,\chi_{v'}(y)\,\chi_{v''}(x \circ y)\right] = 0$ unless $v = v' = v''$, in which case the expression evaluates to 1. Working back through Equations (3) and (2) gives the claimed result. ■

Given Lemma 7 we can prove Theorem 6 in just a few lines. We have:

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

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 $\Pr[\text{linearity test accepts}] \leq 1 - \varepsilon$, proving the theorem.

4 The Consistency Test

We return to our notation from Section 2, where $\vec{v}$ represents a 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[2] 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 $\{\lambda_{i,j}\}$. 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:

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

(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.

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

Claim 8 *Let M, M' be two unequal $n \times n$ matrices over $\mathbb{F}_2$. Then*

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

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

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

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

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

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 $\hat{f}(\vec{v}) = \pi(\vec{r}) + \pi(\vec{r} + \vec{v})$. (Here, the notation $\hat{f}$ has nothing to do with the Fourier notation used in Section 3.2.) 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.*

4.1 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 9 *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 8). 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. ■

5 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

$$y_k \stackrel{\text{def}}{=} c^{(k)} + \sum_{i,j=1}^{n} c_{i,j}^{(k)} \cdot a_i a_j = 0$$

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

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

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 10 *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. ■

6 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 accept with probability 1. We claim also that if the system of equations is not satisfiable, then the test will reject with probability at least 1/48 (for any π). This is because 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 (it can't since the system is not satisfiable...). 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.

7 A PCP of Proximity

We now show that the PCP given previously can be easily adapted to give something stronger: a PCP *of proximity* (PCPP). Considering PCPPs (rather than PCPs) will be useful when we later use the scheme constructed here as a subroutine in building a PCP with better parameters.

Informally, in a PCPP the verifier is given oracle access not only to a *proof*, but also to a *witness* $\vec{a}$. (The verifier is charged for its queries to $\vec{a}$ as well.) Now, the verifier should reject whenever $\vec{a}$ is not "close" to a satisfying assignment for the system of quadratic equations. Of course, if no such satisfying assignment exists then it is impossible for any $\vec{a}$ to be close to a satisfying assignment.

A bit more formally, for two strings $x, y \in \{0,1\}^n$ define their relative distance to be $\Delta(x,y) \stackrel{\text{def}}{=} |\{i : x_i \neq y_i\}|/n$. We say x, y are δ-close if $\Delta(x,y) \leq \delta$, and x, y are δ-far otherwise. We say that x is δ-close to a set $S \subseteq \{0,1\}^n$ if there exists a $y \in S$ such that x is δ-close to y, and x is δ-far from S otherwise. If S is the empty set, then every x is 1-far from S.

Let δ be a constant. We now define a *PCPP with distance parameter δ* for the $\mathcal{NP}$-complete language of satisfiable quadratic equations: The verifier is given a system of equations as always. It is also given oracle access to *two* strings: a "witness string" $\vec{a}$ and a "proof string" π. (The verifier will be charged for its queries to both strings.) We require:

- If $\vec{a}$ is a satisfying assignment for the given system of quadratic equations, then there exists a π such that the verifier accepts with probability 1.
- Let L be the set of assignments that satisfy the given system. If $\vec{a}$ is δ-far from L, then for every π the verifier rejects with constant probability.

No requirements are imposed in case $\vec{a}$ is not a satisfying assignment itself, but *is* close to some satisfying assignment.

Assume π is within distance 1/48 from a consistent, linear function f. We can add the following test (let's call it the "distance test") to the PCP already constructed to turn it into a proof of proximity: Choose random $i \in \{1, \ldots, n\}$ and let $e_i \in \{0,1\}^{n^2}$ be the vector that is 0 everywhere except at position (i,i). Compute $f(e_i)$, using self-correction as before. Then compare this value to the ith bit of the witness $\vec{a}$, and reject if they are not equal. It is clear that if $\vec{a}$ is a satisfying assignment then there exists a proof π maintaining perfect completeness. On the other hand, if $\vec{a}$ is δ-far from a satisfying assignment then there are two possibilities: if f does not encode a satisfying assignment, then the satisfiability test will reject with high probability. But if f does encode a satisfying assignment, the distance test will reject with probability at least $\delta - 2/48$.

Bibliographic Notes

These notes are adapted from [1, Lecture 18]. Section 3.2 was based on lectures 2 and 3 of O'Donnell's course [3]. Section 7 was adapted from Harsha's thesis [2].

References

[1] O. Goldreich. Introduction to Complexity Theory (July 31, 1999).

[2] P. Harsha. Robust PCPs of Proximity and Shorter PCPs. PhD thesis, MIT, 2004. Available at `http://ttic.uchicago.edu/~prahladh/papers/thesis/`

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