6.895 PCP and Hardness of Approximation	MIT, Fall 2010
Lecture 3: The Sum Check Protocol	
Lecturer: *Dana Moshkovitz*	Scribe: *Michael Forbes and Dana Moshkovitz*

In this lecture we show that under an assumption we refer to as *the low degree testing assumption*, we have a weak PCP Theorem

$$NP \subseteq PCP_{1,0.99}[O(\log n), (\log n)^{O(1)}],$$

i.e., a PCP Theorem with a poly-logarithmic number of queries.

For this lecture we use a simplified version of the low degree testing assumption. This simplified version is in fact over-simplified, and, as is, false. However, we use it as a place holder for the more intricate version of the assumption that is both correct and sufficient for our needs. More details about the assumptions are in the next section. In the next lectures we formulate the more intricate version of the assumption, prove its validity, and argue that it indeed suffices for the weak PCP Theorem.

1 The Low-Degree Testing Assumption

The low degree testing assumption states the existence of a probabilistic verifier that checks whether a function $f : \mathbb{F}^m \to \mathbb{F}$ is of degree at most $d \ll |\mathbb{F}|$. The verifier does so by evaluating the function on only a small number of inputs chosen in a randomized way. We think of f as given by its table. This is a table with $|\mathbb{F}^m|$ entries, one for every $x \in \mathbb{F}^m$. At position x we have $f(x)$. This is the way we usually represent functions for PCP purposes, and it makes sense when $|\mathbb{F}^m|$ is sufficiently small. In PCP contexts, the parameters are chosen so $|\mathbb{F}^m| \leq n^{O(1)}$.

As explained above, for this lecture we use a simplified version of the low degree testing assumption that is *false*, and is a placeholder for the true assumption:

Assumption 1.1 (Simplified Low Degree Testing)**.** *There is a constant $\delta > 0$, such that given the table of a function $f : \mathbb{F}^m \to \mathbb{F}$ and a degree $d \leq \delta |\mathbb{F}|$, there is a probabilistic verifier for the statement "$\deg f \leq d$". The verifier is given access to f and to an auxiliary proof and satisfies the following:*

- Completeness: *If $\deg f \leq d$, then there is a proof that the verifier always accepts.*
- Soundness: *If $\deg f > d$, then for any proof, the verifier accepts with probability at most $1/3$.*

The verifier uses $O(\log(|\mathbb{F}|^m))$ random bits. It makes only $|\mathbb{F}|^{O(1)}$ queries to f.

The true low degree testing assumption differs from the simplified assumption in that the soundness condition $\deg f > d$ is replaced with a stronger condition. More about the necessity of this change is in the next lecture.

2 Road Map

Let us sketch the outline of the proof. We start with the NP-complete problem Circuit-Sat. The input to this problem is a Boolean circuit over $\{\wedge, \vee, \neg\}$, and the output is whether the circuit satisfiable or not. We construct a verifier for Circuit-Sat with perfect completeness and soundness $\approx \frac{1}{2}$, whose tests are quadratic equations over $GF(2)$. This verifier makes a linear number of queries to its proof, and so falls short of proving the PCP theorem we are after. Nonetheless, it reduces our problem to verifying low degree (here: quadratic) equations. We then present a protocol for verifying a low degree equation. In the protocol, the verifier makes only a polylogarithmic number of queries to the proof. This requires the low degree testing assumption.

3 Quadratic Equations

We construct a verifier for Circuit-Sat, whose tests are quadratic equations. For the proof it will be convenient to assume that the only gates in the circuit are AND gates where each of their two inputs may be nagated. Let us call such circuits *canonical circuits*. Note that the standard Circuit-Sat problem reduces to the Circuit-Sat problem on canonical circuits, so we can assume without loss of generality that the circuits are canonical.

Theorem 1. Circuit-Sat *on canonical circuits of size n has a verifier $V \in \mathcal{V}[O(\log n), O(n)]$ with completeness 1 and soundness $(\frac{1}{2} + \varepsilon)$ for $\varepsilon = 0.001$ that performs tests that are quadratic equations over $GF(2)$.*

Proof. Let C be the input circuit. Suppose C has n wires. First we prove the theorem for randomness $O(n)$. Then, in the "derandomization" paragraph below, we explain why the randomness can be decreased to $O(\log n)$.

The prescribed auxiliary proof contains one bit for each of the n wires in the circuit. The bit is the value on the wire when the input to the circuit is satisfying. Each quadratic equation $g = 0$ with quadratic polynomial g corresponds to a gate as follows:

- An AND gate whose incoming wires are $input_1, input_2$, and its output wire is *output* corresponds to $output - input_1 \cdot input_2 = 0$.
- If $input_1$ [or $input_2$] is negated, we replace it by $(1 - input_1)$ [or $(1 - input_2)$].
- If the output is also the output of the circuit, we replace *output* by 1 (in fact, by $(1 - 0 \cdot output)$, so all polynomials depend on three wires: the two input wires and the output wire).

Denote the quadratic polynomials corresponding to the gates by $g_1, g_2, g_3, \ldots$ The verifier picks a uniformly random subset I of the gates, and checks that $\sum_{i \in I} g_i = 0$.

There are 2^n subsets the verifier may choose, and so it uses n random bits. The verifier makes at most n queries.

If C is satisfiable, for the prescribed proof, V accepts with probability 1.

Assume C is not satisfiable. Fix some Boolean value for every wire. The equation corresponding to at least one of the gates does not hold; let us denote it by i_0. Now observe the test by the verifier. The choice of the uniformly random subset can be seen as a two-step process: (1) Pick uniformly at random a subset of all the gates but i_0; (2) Decide uniformly at random

whether to add i_0 to the subset. The sum over the subset in (1) is either 0 or 1. Assume without loss of generality that is 1. The total sum is 0 and the verifier accepts exactly when i_0 is added to the subset in (2). This happens with probability 1/2. Hence, the soundness error is 1/2.

Derandomization: Below we describe a collection of $O(n)$ subsets of gates on which the completeness and soundness arguments above go through (in fact, the soundness error increases slightly from $\frac{1}{2}$ to $\frac{1}{2}+\varepsilon$). To pick a subset from the collection the verifier needs only $O(\log n)$ random bits.

The completeness argument will go through no matter which collection of subsets is chosen. For the soundness argument to go through, the collection should satisfy the following: if there is a gate whose quadratic polynomial evaluates to 1 rather than to 0, the sum over a random subset from the collection is 1 with probability at least $(\frac{1}{2}-\varepsilon)$.

To construct the collection we use the generating matrix G of a binary linear code with distance $(\frac{1}{2}-\varepsilon)$. For every $x \neq \vec{0}$, the codeword Gx has at least $(\frac{1}{2}-\varepsilon)$ fraction of ones. Think of the columns of G as corresponding to the gates, and of its rows as corresponding to the possible subsets. The (i,j) entry is 1 if the i'th subset contains the j'th gate. Think of a vector x as giving for each gate the value of its corresponding quadratic polynomial. The vector Gx indicates the result of the sum for each subset. The vector Gx having at least $(\frac{1}{2}-\varepsilon)$ fraction of ones corresponds to the verifier rejecting with probability at least $(\frac{1}{2}-\varepsilon)$.

We take G with $\Theta(n)$ rows, corresponding to $\Theta(n)$ subsets. Such can be found by running the algorithm in Theorem 7 of the coding theory notes. □

4 The Structure of the Verifier

Next we construct a verifier that given a system of quadratic equations over $GF(2)$, and oracle access to a proof, makes only a *poly-logarithmic number of queries* to the proof, and yet:

- Completeness: If there is an assignment to the variables that satisfies all equations, then there is a proof that makes the verifier to always accept.
- Soundness: If no assignment to the variables satisfies more than $(1/2-\varepsilon)$ fraction of the equations, then the verifier accepts with probability at most $\max\{d/|\mathbb{F}|, 1/3\}$.

The verifier uses only logarithmic randomness. This gives the desired PCP Theorem, using Theorem 1.

The first idea is to ask the prover to encode the assignment to the variables $x_1,\ldots,x_n$. We use a systematic encoding of (a slight variant of) the Reed-Muller code. This means that a codeword is the table of a low degree polynomial, and the first entries of the table are $x_1,\ldots,x_n$. This is often referred to as a "low-degree extension". The verifier uses the low degree testing assumption to verify that the table is indeed a low degree polynomial. The verifier then applies a protocol called the "Sum-Check" protocol, to be described. This protocol checks an equation is satisfied using only poly-logarithmic number of queries, i.e., without even going over the entire equation! It does so by querying additional proof supplied by the prover (poly-logarithmically many).

5 Low-Degree Extension

Let $\delta > 0$ be a parameter (this parameter will later determine the soundness error of the sum check protocol). Let h and m be parameters that satisfy $n = h^m$. We take $h = \log n$, $m = \log n / \log\log n$. Let $\mathbb{F}$ be a finite field of characteristic 2 and size $|\mathbb{F}| = m(h-1)/\delta$. Pick some canonical set $H \subset \mathbb{F}$ with $|H| = h$. We think of H^m as a sub-cube of $\mathbb{F}^m$. We note that $|\mathbb{F}^m| = |H^m| \cdot \Theta((m/\delta)^m)$. For $m = \log n / \log\log n$ and $\delta \geq 1/m^{O(1)}$, $|\mathbb{F}^m| \leq n^{O(1)}$.

Pick some canonical identification of H^m with the set $[n]$. With this identification, the assignment to $x_1, \ldots, x_n$, which can be thought of as a function $[n] \to \{0,1\}$, can also be thought of as a function $f : H^m \to \{0,1\}$. We now use the following lemma:

Lemma 5.1 (Low Degree Extension)**.** *Let $\mathbb{F}$ be any field. Let $H \subseteq \mathbb{F}$ be a set of size h. Then any function $f : H^m \to \mathbb{F}$ can be uniquely extended to a function $\bar{f} : \mathbb{F}^m \to \mathbb{F}$, where $\bar{f}$ is is an m-variate polynomial with individual degree at most $h-1$ in each variable. We call $\bar{f}$ the low degree extension of f.*

We note that the total degree of the low degree extension is $d \doteq m(h-1)$, and we chose the finite field so $d \leq \delta |\mathbb{F}|$. Note that a low degree extension is a systematic encoding of the variant of the Reed-Muller code that considers polynomials with bounded individual degrees, rather than bounded total degree. The individual degree bound is used to guarantee the uniqueness of the low degree extension, and is not important to the PCP construction.

6 A Protocol Using Sum Check

The verifier picks a random quadratic equation test, and verifies its satisfaction.

Assume that the quadratic equation is of the form:

$$\sum_{0 \leq i < j}^{n} p_{i,j} x_i x_j = C$$

where $p_{i,j}$, C are in $GF(2)$ and arithmetic is over $GF(2)$. In the general case, there may be terms of the form $p_i x_i$, but for notational simplicity we ignore them. Incorporating them back to the protocol is fairly straightforward.

Let $a : [n] \to \{0,1\}$ denote the assignment to the variables $x_1, \ldots, x_n$. Using the identification of $[n]$ with H^m, we can write the equation as follows:

$$\sum_{y,z \in H^m} p_{y,z} a(y) a(z) = C.$$

Let us define $\varphi : H^{2m} \to \{0,1\}$ to be:

$$\varphi(y,z) \doteq p_{y,z} a(y) a(z).$$

In this notation we wish to verify that:

$$\sum_{x \in H^{2m}} \varphi(x) = C.$$

Using low degree extension, and now carrying out arithmetic over $\mathbb{F}$, this is the same as:

$$\sum_{x \in H^{2m}} \bar{\varphi}(x) = C.$$

$\bar{\varphi}$ is a $2m$-variate polynomial of degree at most $2m(h-1)$ over the finite field $\mathbb{F}$. We used the characteristic 2 of the field $\mathbb{F}$ to ensure arithmetic over $\{0,1\}$ is as before.

Standard verification of the equality takes $\left|H^{2m}\right| \leq n^{O(1)}$ queries to $\bar{\varphi}$. The sum check protocol performs the verification using only $2m \cdot |H|$ queries to $\bar{\varphi}$ and to an additional auxiliary proof. For our choice of parameters, $|H| = \log n$, $m = \log n / \log\log n$, so this is less than $(\log n)^2$ queries. The sum-check protocol relies on the low degree of $\bar{\varphi}$, and on the low degree testing assumption.

The final protocol using the sum check protocol is as follows:

1. The prover specifies:
   (a) A table $A : \mathbb{F}^m \to \mathbb{F}$ that is supposed to be the low degree extension $\bar{a}$.
   (b) For every quadratic equation test, indexed by r, a table $\Phi_r : \mathbb{F}^m \to \mathbb{F}$ that is supposed to be the low degree extension of φ_r associated with the equation.
   (c) The auxiliary proof required for the sum check protocol.
2. The verifier verifies A is of degree at most d.
3. The verifier picks r at random.
4. The verifier verifies Φ_r is of degree at most d.
5. The verifier verifies that
$$\sum_{x \in H^{2m}} \Phi_r(x) = C$$
by performing the sum check protocol.
6. The verifier verifies that $\Phi_r(y,x) = p_{y,z} A(y) A(z)$ by picking at random $y, z \in \mathbb{F}^m$ and checking that
$$\Phi_r(y,z) \doteq p_{y,z} A(y) A(z).$$
7. The verifier accepts iff it accepts in all steps.

When the system of quadratic equations is satisfiable, and the proofs are as prescribed, the verifier always accepts. When the system of quadratic equations is not satisfiable, at least one of the conditions the verifier attempts to verify fails. In this case, no matter which condition it is, the verifier would accept with probability at most $\max\{d/|\mathbb{F}|, 1/3\}$ [we will show this is the soundness error of the sum check protocol as well].

7 The Sum Check Protocol

The sum check protocol uses the idea of "equality testing by comparing on a random point" that we already used above. The additional idea is to use the recursive structure of the sum $\sum_{u \in H^l} \varphi(u)$. We can put this sum in the root of an $|\mathbb{F}|$-ary tree. Each of its children will contain the sum $\sum_{u_1 \in H^{l-1}} \varphi(u_0, u_1)$ for some $u_1 \in \mathbb{F}$. This is again a sum of the form $\sum_{u \in H^l} \varphi_{u_0}(u_1)$ for the low degree polynomial $\varphi_{u_0} : \mathbb{F}^{l-1} \to \mathbb{F}$ defined by $\varphi_{u_0}(u_1) = \varphi(u_0, u_1)$. Each of the children will have $|\mathbb{F}|$ children of its own defined similarly, and so forth, for $l+1$ levels. The protocol verifies the original sum by going down the tree. It requires a number of queries proportional to the depth of the tree, rather than to its size!

Theorem 2 (Sum Check Protocol)**.** *Let $\varphi : \mathbb{F}^l \to \mathbb{F}$ be a polynomial of total degree at most d given by its table. Under the low degree testing assumption, for $H \subseteq \mathbb{F}$, with $|H| = h$, and $C \in \mathbb{F}$, the identity*

$$\sum_{u \in H^l} \varphi(u) = C$$

can be checked using lh queries to φ and to an auxiliary proof. The verifier uses $O(l \log |\mathbb{F}|)$ random bits, has perfect completeness and soundness $\max\{d/|\mathbb{F}|, 1/3\}$ (where $1/3$ is the soundness in the low degree testing assumption).

Proof. For $i = 0, \ldots, l$, we define the partial sum functions:

$$g_i(x_1, \ldots, x_i) := \sum_{a_{i+1}, \ldots, a_l \in H} \varphi(x_1, \ldots, x_i, a_{i+1} \ldots, \ldots, a_l) \tag{1}$$

Observe that the testing $\sum_{u \in H^l} \varphi(u) = C$ is equivalent to testing if:

1. $g_0 \equiv C$.
2. $g_l \equiv \varphi$.
3. $g_i(x_1, \ldots, x_i) = \sum_{a \in H} g_{i+1}(x_1, \ldots, x_i, a)$ for $i = 0, \ldots, l$.

With this in mind, the proof system is the list of functions $G_i : \mathbb{F}^i \to \mathbb{F}$, where G_i is supposed to be g_i. The functions are given by their tables. The test will be

1. Using the low degree testing assumption, verify that all G_i's are of degree at most d.
2. Verify that $G_0 \equiv C$.
3. Verify that $G_l \equiv \varphi$ by picking $x \in \mathbb{F}^l$ uniformly at random, and checking that $G_l(x) = \varphi(x)$.
4. Verify that $G_i(x_1, \ldots, x_i) = \sum_{a \in H} G_{i+1}(x_1, \ldots, x_i, a)$ for $i = 0, \ldots, l$, by picking $x_1, \ldots, x_l \in \mathbb{F}$ uniformly at random, and checking that the identity holds for them.
5. If all the checks were verified, accept. Otherwise, reject.

Clearly, the number of queries is $O(lh)$. Further, the randomness is $O(l \log |\mathbb{F}|)$ and the verifier has perfect completeness, i.e., accepts with probability 1 if the G_i's are indeed the g_i's, and $\sum_{u \in H^l} \varphi(u) = C$.

We now turn to soundness. If $G_0 \equiv C$, $G_l = \varphi$ and $G_i = g_i$ for all i, then $\sum_{u \in H^l} \varphi(u) = C$. Let us enumerate the possible cases when $\sum_{u \in H^l} \varphi(u) \neq C$:

1. $G_0 \not\equiv C$. In this case the verifier always rejects.
2. $G_l \neq \varphi$. In this case, by the Schwartz-Zippel lemma, the verifier rejects with probability at least $1 - d/|\mathbb{F}|$.
3. There is a G_i that is not of degree at most d. By the low degree testing assumption, in this case, the verifier rejects with probability at least $2/3$.

4. All G_j's are of degree at most d, but there is i with $G_i \neq g_i$. Let i be the largest such that $G_i \neq g_i$. The verifier tests:

$$\begin{aligned} G_i(x_1, \ldots, x_i) &\stackrel{?}{=} \sum_{a \in H} G_{i+1}(x_1, \ldots, x_i, a) \\ &= \sum_{a \in H} g_{i+1}(x_1, \ldots, x_i, a) \\ &= g_i(x_1, \ldots, x_i) \end{aligned}$$

   By Schwartz-Zippel, $G_i(x_1, \ldots, x_i) \neq g_i(x_1, \ldots, x_i)$, and hence the verifier rejects, with probability at least $1 - d/|\mathbb{F}|$.

Overall, when $\sum_{u \in H^l} \varphi(u) \neq C$, the verifier accepts with probability at most $\max\{d/|\mathbb{F}|, 1/3\}$. □