6.841 Advanced Complexity Theory	April 11, 2005
Lecture 18	
Lecturer: Madhu Sudan	*Scribe: Jonathan Kelner*

1 Introduction

In the last lecture, we defined probabilistically checkable proofs (PCPs) and started to discuss how varying the amount of randomness and the number of queries available to the verifier affects the class of languages that a PCP can be constructed to accept. For completeness, we recall the definition of a PCP:

Definition 1 *A language L is in the class* $\mathrm{PCP}[r(n), q(n)]$ *if there exists a probabilistic polynomial time oracle machine V (called the* verifier*) with access to an input string x and with oracle access to a* proof *π such that:*

- *V tosses $r(|x|)$ coins, obtaining the random string R.*
- *V makes at most $q(|x|)$ oracle queries.*
- *V outputs an accept/reject verdict $V^{\pi}(x, R)$ such that*
  - *If $x \in L$ then $V^{\pi}(x, R)$ accepts for all R.*
  - *If $x \notin L$ then for all π, $\Pr_R[V^{\pi}(x, R)$ accepts$] \leq 1/2$.*

Today we shall begin the proof that $\mathrm{NP} \subseteq \mathrm{PCP}[\mathrm{polylog}(n), \mathrm{polylog}(n)]$. Before we can do that, we shall need some mathematical preliminaries.

2 Algebraic Background

In this and all subsequent sections, let $\mathbb{F}$ be a finite field, and let $H = \{\alpha_1, \ldots, \alpha_n\} \subseteq \mathbb{F}$ be a collection of distinct field elements.

We shall rely heavily on the ability to perform interpolation with low degree polynomials. The univariate form of this is quite well-known:

Theorem 2 (Univariate Interpolation) *Let $\mathbb{F}$ and H be as above, and let $f : H \to \mathbb{F}$ be arbitrary. Then there exists a polynomial $\widehat{f} : \mathbb{F} \to \mathbb{F}$ of degree at most n such that $\widehat{f}(\alpha) = f(\alpha)$ for all $\alpha \in H$.*

A slightly less well-known version of this holds for multivariate polynomials as well:

Theorem 3 (Multivariate Interpolation) *Let $f : H^m \to \mathbb{F}$. There exists a multivariate polynomial $\widehat{f} : H^m \to \mathbb{F}$ such that $\widehat{f}(\alpha) = f(\alpha)$ for all $\alpha \in H^m$. Furthermore, $\widehat{f}$ has degree at most $n = |H|$ in each variable, so it has total degree at most $mn = m|H|$.*

We shall also make use of the following algebraic fact, which follows easily from the division algorithm:

Theorem 4 *Let $g(x) = \prod_{\alpha \in H}(x - \alpha)$. A is a polynomial with $A(\alpha) = 0$ for all $\alpha \in H$ if and only if $g(x) | A(x)$.*

Theorem 4 doesn't hold for multivariate polynomials, but there is a slight variant of it that remains true:

Theorem 5 *Let $g(x) = \prod_{\alpha \in H}(x - \alpha)$, and suppose that $B : \mathbb{F}^m \to \mathbb{F}$ has $B(\gamma) = 0$ for all $\gamma \in H^m$. Then B can be written as a sum*

$$B(x_1, \ldots, x_m) = \sum_{i=1}^{m} Q_i(x_1, \ldots, x_m) \cdot g(x_i),$$

i.e., B can be written as a sum of a collection of terms, each of which is divisible by $g(x_i)$ for some i.

3 Graph 3-Coloring

We now return to PCPs by describing a PCP for graph 3-coloring. Let $G = (V, E)$ be a graph. A 3-coloring is a map $\chi : V \to \{0, 1, 2\}$ such that for all $(x, y) \in E$, $\chi(x) \neq \chi(y)$.

We can write this in a more functional form. Treat the edge set as a map $E : V \times V \to \{0, 1\}$. A graph has a 3-coloring if and only if there exists some map $\chi : V \to \{0, 1, 2\}$ such that for all $(x, y) \in V \times V$, either $E(x, y) = 0$ or $\chi(x) \neq \chi(y)$.

In order to make a PCP for this, we must first rephrase it in a more algebraic way. To do this, we assume that $\mathbb{F}$ is "large enough," say $|\mathbb{F}| > 10|V|$, and that it does not have characteristic 2, so that 0, 1, and 2 are distinct elements. We then treat V as a subset of $\mathbb{F}$, say by assigning the i^{th} element of $\mathbb{F}$ to the i^{th} vertex.

Now, by multivariate interpolation, we can extend E to a map $\widehat{E} : \mathbb{F} \times \mathbb{F} \to \mathbb{F}$ so that $\widehat{E}$ equals E when restricted to $V \subseteq \mathbb{F}$. Furthermore, we can take $\widehat{E}$ to have degree at most $2n$ (where $n = |V|$). Now, a 3-coloring exists if and only if there exists a map $\chi : \mathbb{F} \to \mathbb{F}$ of degree n such that:

1. $B(x, y) := \widehat{E}(x, y) \cdot \prod_{j \in \{-2,-1,1,2\}}(\chi(x) - \chi(y) - j) = 0$ for all $x, y \in V$.
2. $A(x) := \chi(x)(\chi(x) - 1)(\chi(x) - 2) = 0$ for all $x \in V$.
3. $\deg(A(x)) \leq 3n$.
4. $\deg(B(x)) \leq 6n$.

We shall now discuss how to construct a PCP for these four properties.

4 Proving 3-Colorability

In our first cut at a PCP, the prover will write down truth tables for χ, A, and B. The verifier will then verify:

1. The degrees of χ, A, and B are small.
2. The values given for A and B are consistent with those given for χ.
3. A is zero on $V \subseteq \mathbb{F}$, and B is zero on $V \times V \subseteq \mathbb{F} \times \mathbb{F}$.

We address these three properties in turn.

4.1 Low Degree Testing

All of the tests for property 1. are instances of *low degree testing.* We are given oracle access to a function $f : \mathbb{F}^m \to \mathbb{F}$, and we are charged with determining if f is of some degree d. We do this by making some small number of (possibly random) queries to the oracle, and we are required to succeed with some good probability.

Unfortunately, this is impossible. Suppose we started with some function f that was of degree d and modified its value at one point so that it became of much larger degree. (For example, the zero function is of degree zero, but a function that is nonzero at exactly one point is of degree $|\mathbb{F}| - 1$.) There is no way that we could distinguish the original function from the modified one without making a number of queries that is linear in the field size. As such, we shall content ourselves with a weaker notion: we shall test if a function is "close" to a low degree polynomial.

Definition 6 *Two polynomials* $f, g : \mathbb{F} \to \mathbb{F}$ *are* δ-close *if* $\Pr_x[f(x) \neq g(x)] \leq \delta$.

Definition 7 *A* (d, k, ϵ, δ) low degree tester *is a probabilistic algorithm with oracle access to a function f that:*

- *makes at most k oracle queries,*
- *always accepts if a polynomial f is of degree at most d, and*
- *rejects with probability at least $1 - \epsilon$ if there is no degree d polynomial $\widehat{f}$ that is δ-close to f.*

In the next lecture, we'll talk about how to construct low degree testers and for what parameter ranges they exist. For the remainder of this lecture, we'll just assume that we have one that does what we need it to do.

4.2 Consistency Testing

Testing consistency in property 2. turns out to be quite easy, assuming that we have already tested that the polynomials in question have low degree. Testing whether A and χ are consistent amounts to testing whether

$$A(x) = \chi(x)(\chi(x) - 1)(\chi(x) - 2).$$

We do this by randomly choosing a value of x and verifying that equality holds. This test never falsely rejects. Furthermore, if A and χ are polynomials of respective degrees $3n$ and n and they are inconsistent, this test will reject with probability at least $1 - 3n/|\mathbb{F}|$.

Our low degree testing can't tell us that A and χ are of the appropriate degrees; it can only tell us that they are δ close to polynomials of the appropriate degree. This doesn't cause any real problems though—our test will now find errors with probability at least $1 - 3n/|\mathbb{F}| - 2\delta$. (We pick an x where either A or χ differs from its nearby low degree polynomial with probability at most 2δ.)

We can thus verify that A and χ are consistent with only two queries. A similar argument allows us to test that B and χ are consistent with three queries: we randomly pick x and y and verify that

$$B(x,y) = \widehat{E}(x,y) \cdot \prod_{j \in \{-2,-1,1,2\}} (\chi(x) - \chi(y) - j).$$

4.3 Testing Whether A,B=0 When They Should Be

To test property 3., the naïve approach would be to just randomly pick values where A and B should be zero, evaluate them, and verify that they vanish. Unfortunately, this doesn't work, since we need to verify that A and B vanish at all values in V and $V \times V$, respectively, not just at most such values, and we'd have to make way too many queries to verify this fact with reasonable probability.

Instead, we use Theorems 4 and 5. To test whether A is zero at the appropriate locations, let $g(x) = \prod_{\alpha \in H}(x - \alpha)$. By Theorem 4, it suffices to check whether $g(x)$ divides $A(x)$, i.e., whether there exists some $R_A(x)$ such that $A(x) = R_A(x)g(x)$. To do this, we slightly modify the protocol by having the prover send a table for $R_A(x)$ as well. We are now back to cases we have already analyzed: we verify that $R_A(x)$ is of low degree and that A, R_A, and g are consistent. (As before, it suffices to check whether R_A is close to a low degree polynomial.)

To verify that B is zero, we use Theorem 5. If B is zero on $V \times V$, there are some polynomials Q_B and R_B such that $B(x,y) = Q_B(x,y)g(x) + R_B(x,y)g(y)$. The prover will send Q_B and R_B, and, as above, we can check the degrees and consistency.

We would therefore be done if we could actually perform the asserted low degree testing. This will be discussed in the next lecture.