18.405J/6.841J Advanced Complexity Theory	April 27, 2000
Lecture 22	
Lecturer: Dan Spielman	*Scribe: Ryan O'Donnell*

1 NEXP $\subseteq$ PCP(poly, poly)

1.1 Overview

Recall from last time the way we intend to prove **NEXP** $\subseteq$ **PCP**(poly, poly). At first we tried to require the proof Π to contain a satisfying assignment $A : \{0,1\}^n \to \{0,1\}$ for the implicit circuit. But in order to make the proof work, we instead had to require the proof to contain the (unique) multilinear extension $\hat{A} : \mathbf{F}^n \to \mathbf{F}$ of A, where $\mathbf{F}$ is some field.

Given this, we can have the proof also contain a table of prover strategies which argue that a **PSPACE** machine verifies that A satisfies the circuit. This completes the PCP for IMPLICIT-CIRCUIT-SAT, modulo one detail: The verifier needs to be able to convince itself with high probability that there is some multilinear function L such that L and $\hat{A}$ agree at almost all places. This requirement takes us into a field of study known as linearity testing.

1.2 Linearity Testing

First, a definition:

Definition 1 *Given two functions $f, g : \mathbf{F}^n \to \mathbf{F}$, we say f and g are "ϵ-close" if $Pr_{\bar{x}\in\mathbf{F}^n}[f(\bar{x}) = g(\bar{x})] > 1 - \epsilon$.*

We would like to prove a statement of the following flavor:

Theorem 2 *Given a function (table) $\hat{A} : \mathbf{F}^n \to \mathbf{F}$, and $\epsilon > 0$, we can in polynomial time construct and perform a test on $\hat{A}$ such that:*
$Pr[\hat{A}$ passes test$] > \epsilon \Rightarrow \exists$ multilinear $L : \mathbf{F}^n \to \mathbf{F}$ such that L and $\hat{A}$ are ϵ-close.

Notice that in the IP we use, we end up needing the value of $\hat{A}$ at only one, uniformly randomly chosen element of $\mathbf{F}^n$ (to see this, one needs to study the protocol we gave). Thus, if our verifier becomes convinced that L and $\hat{A}$ are ϵ-close, then it can simply determine L at the requisite place, and be almost certain it is the desired value of $\hat{A}$.

Now as it turns out, proving a theorem like Theorem 2 is fairly hard. The main problem is that the domain of $\hat{A}$ is $\mathbf{F}^n$. First we will give a test that works in the one-variable case (i.e., when the domain of $\hat{A}$ is $\mathbf{F}^n$). This sort of test will not generalize in polynomial time to the n-variable case. Consequently, we will give a different test for the two-variable case, which will illustrate the sort of test needed for the general case.

One more remark: By repeating the test many times, it's easy to see that in Theorem 2, we can just require the result when the probability that $\hat{A}$ passes the test is greater than $1 - \delta$ for some constant δ.

1.3 The One-Variable Test

Notice that if $\ell : \mathbf{F} \to \mathbf{F}$ is a linear function, then ℓ is determined by its values at two points; in particular, $\ell(x) = \ell(0) + x(\ell(1) - \ell(0))$. So our test is this: Read $\hat{A}$ at three places, 0, 1, and a uniformly randomly chosen $x \in \mathbf{F}$. Then accept if and only if $\hat{A}(x) = \hat{A}(0) + x(\hat{A}(1) - \hat{A}(0))$. It's immediate that if this test passes with probability α, then $\hat{A}$ is at least $(1-\alpha)$-close to some linear function, namely the one which is a linear interpolation of $\hat{A}(0)$ and $\hat{A}(1)$.

If we repeat the test many times, rejecting if the test fails even once, then we can get a composite test with conditions like those in Theorem 2.

1.4 The Two-Variable Case

Now to handle the two-variable case, we could just apply the obvious extension of the one-variable case, namely: Read $\hat{A}$ at four points, say $(0,0)$, $(1,0)$, $(0,1)$, and $(1,1)$; interpolate to get a two-dimensional linear function; then check if $\hat{A}$ agrees with this function in a random place. Unfortunately, this scheme does not generalize to n dimensions, since it would involve interpolating a polynomial on 2^n points. So we perform a different test, an "axis-parallel" test, which *does* generalize to many dimensions.

The test is this:

- Choose $(x,y) \in \mathbf{F}^2$ at random.
- Check if $\hat{A}(x,y) = \hat{A}(x,0) + y(\hat{A}(x,1) - \hat{A}(x,0))$.
- Check if $\hat{A}(x,y) = \hat{A}(0,y) + x(\hat{A}(1,y) - \hat{A}(0,y))$.
- Accept if and only if both checks succeed.

Claim 3 *If the probability that this test exceeds $1-\epsilon$, then there is a multilinear L in two dimensions which is 4ϵ-close to L.*

Proof Call a point (x,y) "good" if it passes the two checks. Assume that all but an ϵ fraction of the points are good. Then there must exist two "rows", corresponding to say y_1 and y_2, such that all but a 2ϵ-fraction of the points in the rows are good. (By a "row corresponding to y_0", we mean the set $\{(x,y_0) : x \in \mathbf{F}\}$. Similarly for "columns".)

By the analysis in the one-dimensional case, there are two one-dimensional linear functions L_1 and L_2 such that L_i is 2ϵ-close to the row corresponding to y_i, for $i = 1,2$. Further, the points of agreement are exactly the good points in the row.

Define the two-dimensional linear function L by interpolating L_1 and L_2 between y_1 and y_2, i.e., put $L(x,y) = \frac{y-y_1}{y_2-y_1}L_2(x) + \frac{y-y_2}{y_1-y_2}L_1(x)$. Now any point whose column intersects the two rows in good points will agree with L. But at most 4ϵ columns can fail to intersect both rows in a good point.

Hence at least a $(1-4\epsilon)$-fraction of the points agree with L, and we are done. ■

1.5 MIP = NEXP

This completes the proof that **MIP** = **NEXP**. To recap, in this lecture we've completed the proof that **NEXP** $\subseteq$ **PCP**$(poly, poly)$. To see this, suppose the implicit circuit is unsatisfiable, and that $\hat{A}$ is a proof designed to fool our verifier into acceptance. If $\hat{A}$ does not pass the multilinearity test with high probability, then the verifier will reject. But if does pass the test, then it must be ϵ-close to some multilinear $L : \mathbf{F}^n \to \mathbf{F}$. There is no way L restricted to the hypercube $\{0,1\}^n$ can provide a satisfying assignment for the implicit circuit, since the circuit is unsatisfiable. Thus the IP part of the proof-testing will almost surely reject.

1.6 Further Results

Definition 4 **PCP**$(r(n), q(n))$ *is defined to be the class of languages which have a probalistically checkable proof (with one-sided constant error) in which the verifier uses* $r(n)$ *random bits and reads* $q(n)$ *proof bits.*

We have just finished showing that $\mathbf{PCP}(poly, poly) = \mathbf{NEXP}$. After this was proved, people thought, why don't we try to restrict the resources, and see what classes we get?

It's clear that $\mathbf{NP} \subseteq \mathbf{PCP}(0, poly)$. By scaling down the proof of $\mathbf{PCP}(poly, poly) = \mathbf{NEXP}$, one can fairly easily get $\mathbf{NP} \subseteq \mathbf{PCP}(polylog, polylog)$.

But in fact, we have the very striking result that $\mathbf{NP} = \mathbf{PCP}(O(log), O(1))$! In fact, we even have $\mathbf{NP} = \mathbf{PCP}(O(log), 3)$.

This fact has led to great strides in the seemingly unrelated field of approximation algorithms. For example, consider the following problem, GAP3SAT. You are given a 3-CNF, and you are promised that it is either satisfiable, or at most an α-fraction of its clauses are satisfiable, for α a constant less than 1. You have to decide which case the 3-CNF is in. Using the **PCP**-result above, we can show that this problem is **NP**-hard for some value of α. This leads to hardness results for the approximation of many **NP**-complete problems.