18.405J/6.841J Advanced Complexity Theory	April 25, 2000

Lecture 21

Lecturer: Dan Spielman *Scribe: Peter Hildenbrand*

Last lecture, we showed $MIP = PCP(poly, poly)$. Today, we continued to examine $PCP(poly, poly)$ by proving that $PCP(poly, poly) = NEXP$.

1 $PCP(poly, poly)$

Definition 1 *A* **PCP(i, j) machine** *is a randomized verifier, with i random bits and j queries to a given proof string. It is like an Oracle TM in that the queries to the proof string are answered in one step.*

$$L \in PCP(poly, poly) \text{ if } \exists \text{ a randomized OTM } V^{?} \text{ s.t.}$$
$$x \in L \Rightarrow \exists\ \Pi \text{ s.t. } Pr\,[V^{\Pi}(x) \text{ accepts }] = 1.$$
$$x \notin L \Rightarrow \forall\ \Pi \text{ s.t. } Pr\,[V^{\Pi}(x) \text{ accepts }] < 1/3.$$

2 $PCP(poly, poly) = NEXP$

We will prove that $PCP(poly, poly) = NEXP$ by showing $PCP(poly, poly) \subseteq NEXP$ and $NEXP \subseteq PCP(poly, poly)$. Below, we will prove $NEXP \subseteq PCP(poly, poly)$. The proof that $PCP(poly, poly) \subseteq NEXP$ has been left to the students as problem set 4, question 1.

Lemma 2 $NEXP \subseteq PCP(poly, poly)$

Proof To prove the above lemma, we will solve an $NEXP$-complete problem using a PCP(poly,poly) machine. For this purpose, we will solve the $NEXP$-complete problem Implicit-Circuit-Sat. We will use machinery from the proof that $PSPACE \subseteq IP$ in the proof.

Let w be an instance of an Implicit-Circuit-Sat problem. w defines an exponentially big circuit, $C(w)$, to be satisfied. Let Π contain the satisfying assignment A (which has an exponential number of variables) for $C(w)$. If the variables of $C(w)$ are $x_1, x_2, ..., x_{2^n}$ then $A : x_i \to \{0, 1\}$.

Claim: $\exists$ a PSPACE TM M s.t. $M^{A(w)}$ accepts if A satisfies $C(w)$.
Proof Consider Implicit-3Sat (also an $NEXP$-complete problem), where the input is a circuit. On input i, the circuit outputs the i^{th} clause of the 3Sat formula (i.e. gives the three variables and negations if they exist).

Let M^A:

1. for $i = 1, ..., 2^n$
2. generate clause i
3. read variable assignments for the variables in clause i
4. if clause i is not satisfied, **reject**
5. if you reach this step, **accept**

■

Idea: Now that we know a PSPACE machine exists to solve Implicit-3Sat, let's try to use the machinery from $PSPACE \subseteq IP$ to solve $NEXP \subseteq PCP(poly, poly)$. To do this, Π should also contain a table of strategies of the prover that argues that M^A accepts. This cannot work, however, for two reasons:

1. Our $PSPACE \subseteq IP$ protocol did not have anything like A (or an Oracle).
2. A is exponentially large, but V^Π can only see some polynomial number of bits from $A \Rightarrow C(w)$ contains a clause that you look at with low probability.

Let's reexamine the proof that $PSPACE \subseteq IP$:
Proof Given M and input w, consider the graph on configurations of M.

Let FromTo(x,y,k) $= 1$ if you can get from x to y in 2^k steps; 0 otherwise.

Create the polynomial $P_{FTo}(\bar{x}, \bar{y}) = 1$ if you can get from $\bar{x}$ to $\bar{y}$ in 1 step; 0 otherwise.

We can evaluate P_{FTo} for $\bar{x}, \bar{y} \in \{0,1\}^n$ or $\mathcal{F}^n$. We just have to check local (constant size) conditions. If we create a polynomial for each local condition and multiply them all together, we can simulate the PSPACE machine. ■

What happens if you have access to A or an Oracle?
We must redefine the configurations of $\bar{x}$ and $\bar{y}$.
Let $\bar{x} = x_w$ (work tape), x_q (query tape), x_a (answer bit $A(x_q)$).
Let $\bar{y} = y_w$, y_q, y_a.
All of the local polynomials carry over except $x_a \to y_a$.
So, we need to check if $y_a = A(y_q)$
try: $1 - (y_a - A(y_q))^2$ (1 if equal; 0 otherwise)
We need to evaluate the above equation when $y_q \in \mathcal{F}^{|y_q|}$, not just $y_q \in \{0,1\}$.
So, Π should contain $\hat{A}$, the multi-linear extension of A

$A : \{0,1\}^m \to \{0,1\}$
$\exists$ P_A m-variable polynomial of degree ≤ 1 in each variable
$P_A(\bar{x}) = A(\bar{x})$, $\bar{x} \in \{0,1\}^n$
$\hat{A}$ should contain $P_A(\bar{x})$ $\forall \bar{x} \in \mathcal{F}^m$, but we need to check that $\hat{A}$ has the correct form.

Let Π contain $\hat{A}$, the table of the strategy of the prover who argues that M^A accepts.

Let V:

1. check that $\hat{A}$ does look like a multi-linear polynomial (**reject** if it does not)
2. go through IP that M^A accepts by evaluating P_{FTo} at one place
3. **accept** if steps 1 and 2 accepted

Let's examine step 2 of V for the 1 dimensional case:
We want to check that $\hat{A}$ looks like a degree 1 polynomial by reading a few places $\hat{A}(i)$ where i = 0, 1, ..., $p-1$:

1. choose $x \in 0, ..., p-1$ at random
2. repeat k times
3. check that $\hat{A}(x) = \hat{A}(0) + x(\hat{A}(1) - \hat{a}(0))$, if it does not, **reject**

4. **accept** if you reach this step

If Pr test accepts $> \epsilon \Rightarrow$ for most x, $\hat{A}(x)$ is what you get by interpreting a linear polynomial through $\hat{A}(0)$ and $\hat{A}(1)$

So, $\exists$ $l(x)$: linear s.t.

$$Pr_x[l(x) = \hat{A}(x)] > 3/4 \text{ (depends on k, q)}$$

■