Instructor: Andrej Bogdanov **Notes by:** Andrej Bogdanov

In the last lecture we began describing a probabilistically checkable proof for arbitrary decision problems in nondeterministic exponential time. Now we fill in the remaining pieces.

1 Probabilistically checkable proofs for NEXP

Let us briefly sketch the probabilistically checkable proofs for NEXP problems from last time. We started with an arbitrary decision problem $L \in \text{NEXP}$. We then sketched how, given a description of a nondeterministic Turing Maching N for L, an input x, and a bound on the running time on N of the form $2^{t(|x|)}$ (where $t(n)$ is some polynomial), we can construct in polynomial-time a polynomial $p(y, b) = p(y_1, \ldots, y_m, b_1, \ldots, b_c)$ of degree $2m$ such that $N(x)$ accepts iff there exists an assignment $A : \{0,1\}^m \to \{0,1\}$ such that

$$p(y, \tilde{A}(y)) = 1 \text{ for all } y \in \{0,1\}^m. \tag{1}$$

where $\tilde{A}(y) = (A(y), A(y+1), \ldots, A(y+c/2-1), A(y+2^{m/2}), \ldots, A(y+2^{m/2}+c/2-1))$. Roughly, p checks that a small window of the computation tableau of N on input x starting at position y is valid. If all such windows are valid, then the computation should accept.

We want to check conditions (1) via a probabilistically checkable proof. The proof should roughly play the same role that the prover played in the interactive proof for #3SAT: We should be able to verify the conditions (1) by "asking" the proof to evaluate certain polynomials derived from p at random points. In addition to the conditions 1, we also want to enforce the conditions $A(y)(A(y) - 1) = 0$ for all $y \in \{0,1\}^m$, which guarantee that A is a boolean assignment.

For the interactive protocol to work, we must be able to enforce the condition that $p(y, \tilde{A}(y))$ is a polynomial of low degree in y. But how can we enforce this condition if we don't even know what A is? The trick is to require A to be of a special form – a *multilinear* polynomial. Since a multilinear polynomial can have degree at most m, the total degree of $p(y, A(y))$ will not exceed $O(m^2)$ (recall that p has degree $O(m + c)$).

We now show that if A is indeed multilinear, then given access to A (over a sufficiently large field) we can verify condition (1) using a probabilistically checkable proof. In fact, this can be done using an interactive proof.

Claim 1. *For every problem $L \in \text{NEXP}$ there is an interactive proof (P^A, V^A) in which both the prover and verifier have oracle access to a function family $A : \mathbb{F}^m \to \mathbb{F}$ with the following properties:*

$$x \in L \implies \exists A : \Pr[(V^A, P^A) \text{ accepts } x] = 1$$
$$x \notin L \implies \forall P^* \ \forall \text{ multilinear } A : \Pr[(V^A, P^{*A}) \text{ accepts } x] < 1/3.$$

Here $\mathbb{F}$ is any prime field of size at least $12 \cdot 2^m$.

Notice that this protocol is weaker than usual because the soundness condition holds not for all A, but only for multilinear A. We will somehow have to enforce the condition that A is multilinear. For this we will take advantage of the fact that we are working with probabilistically checkable proofs and not simply interactive proofs.

Proof sketch. The verifier will attempt to check the condition

$$\sum_{y \in \{0,1\}^m} \left((p(y, \tilde{A}(y)) - 1) \cdot t^{0y} + A(y)(A(y) - 1) \cdot t^{1y} \right) = 0 \tag{2}$$

for a random $t \in \mathbb{F}$. Here t^z is the number t raised to the integer whose binary expansion is z. To check this condition, the verifier strips the summations as in the interactive protocol for #3SAT. Recall that this is a protocol that checks statements of the form

$$\sum_{y \in \{0,1\}^m} q(y) = k$$

as long as q is a polynomial of degree $\text{poly}(m)$ and the verifier can evaluate q at arbitrary points in $\mathbb{F}^m$.

To apply this protocol, we need to check that each of the above terms is a polynomial of degree at most $\text{poly}(m)$ that can be evaluated in $\mathbb{F}^m$ (using access to the oracle A). By assumption, the terms $p(y, \tilde{A}(y)) - 1$ and $A(y)(A(y) - 1)$ are both polynomials of degree $O(m)$ that can be evaluated by V given oracle access to A. It remains to check that for fixed t, t^{0y} and t^{1y} are also such polynomials. To see this, note that if $z_m \dots z_1$ is the binary expansion of z, we can write

$$t^z = \prod_{i=1}^{m} t^{2^i \cdot z_i} = \left(1 + (t^{2^i} - 1) \cdot z_i\right).$$

This is a multilinear polynomial in the z_i that can be evaluated in polynomial-time by the verifier, so all terms in the summation are polynomials of degree $O(m^3)$ that can be evaluated by V given oracle access to A.

By the completeness analysis of the interactive protocol for #3SAT, if $x \in L$, no matter which value t the verifier chooses, if the prover answers the queries correctly then V^A accepts with probability 1. If $x \notin L$, then at least one of the conditions $p(y, \tilde{A}(y)) - 1 = 0$, $A(y)(A(y) - 1) = 0$ must be violated for some y. It follows that the expression 2 is not identically zero *as a polynomial in* t. Since the degree of t in this expression is 2^{m+1}, for a random $t \sim \mathbb{F}$ this expression vanishes with probability at most $2^{m+1}/|\mathbb{F}| < 1/6$. Conditioned on this event not happening, by the soundness analysis of the interactive protocol for #3SAT, V^A accepts with probability at most $1/6$. □

2 Implementing the oracle

In proving Claim 1 we made the crucial assumption that the function A given to the verifier is multilinear. Now we must find a way to enforce this assumption.

The function A is provided to the verifier in the form of an exponentially long table $A : \mathbb{F}^m \to \mathbb{F}$. The verifier, who is restricted at examining only polynomially many entries in this table, must somehow be convinced of the fact that A is multilinear.

A moment's thought shows that this objective is impossible: One can start with a multilinear A, and then modify A at a random point to make it non-multilinear. From the perspective of the verifier, this point is very unlikely to be seen on a particular input x, so the verifier will never "know" that the function he is looking at is not multilinear.

However, this is not particularly interesting. As long as the verifier never sees the deficiencies in A, for the purposes of the interactive protocol above, one can pretend that the verifier has access not to A, but to the multilinear function closest to A. Since the verifier will never see the difference between the two, this will not affect the soundness of the protocol.

In general, the function A can differ from a multilinear function in much more that one place. If the two functions differ at not too many places, then perhaps the verifier can "pretend" that A is multilinear. In our setting, the verifier queries the function A at $c + 2$ random points. If we can ensure that the function A differs from its "closest" multilinear function F on at most a $1/9(c+2)$ fraction of points, then with probability at least $1/9$ the verifier would behave as if it were looking at F instead of A.

Let's make this formal. We say that a function $A : \mathbb{F}^m \to \mathbb{F}$ is *δ-far from multilinear* if for every multilinear function $F : \mathbb{F}^m \to \mathbb{F}$, $\Pr_{x \sim \mathbf{F}^m}[F(x) \neq A(x)] > \delta$. A *$\delta$-multilinearity test* is a randomized oracle algorithm T that, on input 1^m and given oracle access to A, runs in time $\mathrm{poly}(m)$ and

$$\begin{aligned} A \text{ is multilinear} &\implies \Pr[T^A(1^m) \text{ accepts}] = 1 \\ A \text{ is } \delta\text{-far from multilinear} &\implies \Pr[T^A(1^m) \text{ accepts}] < 1/2. \end{aligned}$$

As usual, the constant $1/3$ can be replaced with anything from $2^{-\mathrm{poly}(m)}$ to $1 - 1/\mathrm{poly}(m)$ in the second condition.

In the next section we will show that for every constant $\delta > 0$, there exists a δ-multilinearity test as long as $|\mathbb{F}| \gg m/\delta$. Using this fact, we now sketch how to finish the proof that $\mathrm{NEXP} \subseteq \mathrm{PCP}(\mathrm{poly}, \mathrm{poly})$.

Theorem 2. $\mathrm{NEXP} \subseteq \mathrm{PCP}(\mathrm{poly}, \mathrm{poly})$.

Proof sketch. Let L be an arbitrary decision problem in NEXP. The probabilistically checkable proof for L will consist of two parts: The first part is the description of a supposedly multilinear function $A : \mathbb{F}^m \to \mathbb{F}$. The second part describes the responses of the prover in the interactive protocol from Claim 1 when the verifier and prover have both oracle access to A. (There is also a third part, which provides the verifier a description of a prime field $\mathbb{F}$ of size about $2^{O(m)}$ to work over, e.g. by specifying a prime number between $12 \cdot 2^m$ and $24 \cdot 2^m$.)

Here is a description of the verifier V: On input x,

1. Run the δ-multilinearity test on A with $\delta = 1/9(c+2)$. If the test rejects, reject.
2. Simulate the verifier in the interactive protocol from Claim 1 using A as the oracle.

If $x \in L$, then the first part A of the proof equals the unique multilinear extension of some assignment $A : \{0,1\}^m \to \{0,1\}$ that satisfies (1) and the second part gives the responses of the prover in Claim 1. By construction, the verifier will accept with probability 1.

If $x \notin L$, there are several events that may make the verifier accept.

- The function A is δ-far from multilinear
- A is not δ-far from multilinear, but A and its closest multilinear function F differ on some query made by the verifier
- A and F are identical on all queries made by the verifier, but the simulation in step 2 above accepts.

We bound the probability of each of the above events. For event 1, the probability is of accepting is bounded by the soundness of the multilinearity test, which is at most $1/2$.

For event 2, notice that the verifier makes at most $c+2$ queries in F, each of which is uniformly distributed in $\mathbb{F}^m$. On any single query z, the probability that $A(z) \neq F(z)$ is therefore at most $1/9(c+2)$. By a union bound, the probability that A and F differ on some query made by the verifier is at most $1/9$.

For event 3, notice that the behavior of the protocol in Claim 1 would be unchanged if F was used instead of A as the oracle. By the soundness of that protocol, the probability that the verifier accepts is at most $1/3$.

Summarizing, the total accepting probability of the verifier is $\leq 1/2 + 1/9 + 1/3 = 17/18 < 1$. By repeating the protocol several times this probability can be dropped to $1/3$ (while the completeness remains 1). $\square$