Lecture 30

In this lecture we are going to discuss a result by Miltersen and Vinodchandran, which is the first derandomization result that does not use the Nissan-Wigderson construction.

Theorem 1
If there is a language $L \in \mathbf{DTIME}(2^{O(n)})$ such that for every sufficient large n, the non-deterministic circuit complexity of L_n is greater than $2^{\delta n}$, $\delta > 0$, then $\mathbf{P} = \mathbf{BPP}$ and $\mathbf{NP} = \mathbf{AM}$.

The non-deterministic circuit complexity S of a function $f : \{0,1\}^n \to \{0,1\}$ is the size of the smallest non-deterministic circuit that computes f. A non-deterministic circuit has two inputs x, y, where x is the real input, and $C(x)$ accepts iff there exists a y such that $C(x, y) = 1$.

We have already seen the result by Impagliazzo, that $\mathbf{P} = \mathbf{BPP}$ if there is a language with standard complexity $2^{\delta n}$. Here we use the stronger assumption of non-deterministic circuit complexity, but this is the only way we can get the second part of the theorem, namely that $\mathbf{NP} = \mathbf{AM}$. Before going into the proof of the theorem, let us give the following definitions:

Definition 1 *A set $H \subseteq \{0,1\}^n$ is a (S, ϵ)-hitting set if for every circuit C of size S:
If $\mathbf{Pr}_x[C(x) = 1] > \epsilon$ then there exists an $x \in H$: $C(x) = 1$.
This means that a hitting set is a set of strings, such that if the circuit accepts with some probability, then there exists an element of the set, for which the circuit accepts.*

If $G : \{0,1\}^t \to \{0,1\}^n$ is a (S, ϵ) pseudorandom generator then the set $G(s)$: $s \in \{0,1\}^t$ is a (S, ϵ)-hitting set.

A hitting set generator is an efficient algorithm that given n, outputs a $(S(n), \epsilon(n))$ hitting set in $\{0,1\}^n$.

Proposition 2
Given n, if we can compute in poynomial time a $(n^2, \frac{1}{2})$ hitting set in $\{0,1\}^n$, then $\mathbf{P} = \mathbf{RP}$.

Proof: Let A be an $\mathbf{RP}$ algorithm for a language $L \in \mathbf{RP}$. x is an input of length n on which A runs in time t and uses $\leq t$ random bits. Then we know that A can be simulated by a circuit C, so that $A(x, r) = C(r)$ and the size of C is t^2.

$$\text{if} \quad x \in L,\ \mathbf{Pr}_r[C(r) = 1] > \frac{1}{2}$$
$$\text{if} \quad x \notin L,\ \mathbf{Pr}_r[C(r) = 1] = 0$$

By the hypothesis, we can construct a $(t^2, \frac{1}{2})$-hitting set $H \subseteq \{0,1\}^t$ in poly(t) =poly(n) time. Then, we run the circuit on all input in H and we accept iff the circuit accepts for at least one element of H. $\Box$

In the same way, we can also prove that $\mathbf{P} = \mathbf{BPP}$.

Now, if we can construct a hitting set which is good against non-deterministic circuits we can derandomize **AM**. In the paper they construct a hitting set generator which is good against non-determinism and therefore they can derandomize $\mathbf{RP}, \mathbf{BPP}, \mathbf{AM}$.

LEMMA 3
Let $L \in \mathbf{RP}$, $\epsilon > 0$, then there is an algorithm A for L such that for every x, A runs in poly($|x|$), uses m =poly($|x|$) random bits and

$$x \in L \Rightarrow \mathbf{Pr}_r[A(x,r) = 1] \geq 1 - 2^{m^\epsilon - m}$$
$$x \notin L \Rightarrow \mathbf{Pr}_r[A(x,r) = 1] = 0$$

The way to lower down the probability of error to this small amount is by the use of extractors. This resultb also holds for the class AM.

So, according to the previous lemma in order to derandomize the class **RP** it suffices to prove:

PROPOSITION 4
For every ϵ, for evehry c and given m, we can construct in poly(m) time a $(m^c, 1 - 2^{m^\epsilon - m})$-hitting set.

PROOF: Let us start with a language $L \in \mathbf{DTIME}(2^{O(n)})$ and $f : \{0,1\}^l \to \{0,1\}$ a restriction of L to inputs of length l. We can compute f in $2^{O(l)}$ time and the non-deterministic circuit complexity of f is $NCC(f) \geq 2^{\delta n}$.

The proof goes as following: first, we are going to encode the function f with low-degree polynomials and then construct the desired hitting set. If the hitting set cannot be constructed, then this would mean that the function has a small non-deterministic circuit.

- Encoding of f with low-degree polynomials.

  We pick a field $\mathcal{F}$ of size $4 \cdot 2^{\delta l/2c}$ and a subset $I \subseteq \mathcal{F}$ of size $2^{\delta l/2c} = d$. Now, we can view the function f as $f : I^t \to \{0,1\}$, where $t = \frac{2c}{\delta}$. We are going to encode the function f as a polynomial $p : \mathcal{F}^t \to \mathcal{F}$ with degree $|I|$ in each variable.

- Construction of the hitting set.

  Let's consider all parallel-axis lines in $\mathcal{F}^t$ and the set of points

  $$l^i(y) = (a_1, a_2, \ldots, a_{i-1}, y, a_{i+1}, \ldots, a_t).$$

  This is the set of all points, when we fix all but one coordinates and let the other one range over the field.

  Now, the polynomial $p(l^i(y)) = p(a_1, a_2, \ldots, a_{i-1}, y, a_{i+1}, \ldots, a_t)$ is a univariate degree $|I| = d$ polynomial from $\mathcal{F}$ to $\mathcal{F}$, represented by $\mathcal{F} \log \mathcal{F} = m$ bits.

  The hitting set contains exactly the restriction of p to each axis-parallel line. This means, that for each restriction we write the values of p as a string and this string belongs to the hitting set. The size of the hitting set is equal to the number of axis-parallel lines, which is $t\mathcal{F}^{t-1} \approx m^{t-1} = \text{poly}(m)$.

The time of the construction is polynomial in m. This is easy to see, since the construction time is the time to compute f on the field for each line. Hence, it is equal to $m^{t-1} \cdot m \cdot 2^{O(l)} = \text{poly}(m)$, since $2^{O(l)} = m^{O(c/\delta)}$.

So, we have constructed in polynomial time a set of strings, which we claim that is a hitting set.

- The constructed set is a hitting set or else there exists a small non-deterministic circuit for f.

Suppose the construction failed. This means that there exists a circuit C of size m^c such that:

  - C accepts all but 2^{m^ϵ} inputs and
  - C accepts no element from our set.

This means that there is a small subset of $\{0,1\}^m$, where our circuit returns 0 and our set lies entirely in this subset. Let us now consider the set

$$P = \{\text{set of all } q : \mathcal{F} \to \mathcal{F} \text{ univariate polynomials of degree } d \text{ such that } C(q) = 0\}.$$

The input of the cirquit is the binary representation of the polynomial. We know that $|P| \leq 2^{m^\epsilon}$. Moreover, we can see that there exists a set $S \subseteq \mathcal{F}, |S| \leq m^\epsilon$ such that

$$\forall q_1, q_2 \in P, q_1 \neq q_2 \ \ \exists x \in S : q_1(x) \neq q_2(x).$$

Now, let us consider the complete set of values of all polynomials in P. As we have said, the degree of the polynomials is $d = \frac{1}{4}\mathcal{F}$, so any pair of polynomials agree on at most $\frac{1}{4}$ of the values. Suppose we pick an entry at random. With probability greater than $\frac{3}{4}$ the q_1, q_2 are different. If we pick now $k = m^\epsilon$ entries then the probability that the two polynomials are not distinguished is

$$\mathbf{Pr}[q_1, q_2 \text{ are not distinguished}] \leq \left(\frac{1}{4}\right)^k = \frac{1}{(2^{m^\epsilon})^2}$$

Since this probability is less than the inverse of the total number of possible pairs, we can definitely distinguish between the polynomials.

- Non-deterministic circuit to compute f.

We start by hard-wiring the value of the polynomial p on all elements of S^t. If we can compute p in the whole field, then we have the value of f as well. Also, if f is restricted on a line, then this will be a string of our set which is rejected by the circuit.

The circuit knows the value of p on S^t. Let us consider the 2-dimensional case for the rest of the construction. So, for every $q(x) = p(x, a),\ a \in S$ we know $q(x),\ x \in S$. The polynomial q is a unique degree polynomial such that $C(q) = 0$ and q agrees with the hard-wired values on S. Now, we'll try to compute the value of $p(a_1, a_2)$, $a_2 \in S, a_1 \notin S$. We can do that in a non-deterministic way, by guessing a degree d polynomial q and then check that $q(x) = p(x, a_2)$ for every $x \in S$ and $C(q) = 0$. We output $q(a_1)$. This procedure will either reject or give the right value.

In this way, if we have precomputed p in $S \times S$, then we can compute p in lines. Hence, we can compute the value of p in the whole field.

In a 3-dimensional case, the computation of $p(a_1, a_2, a_3)$ reduces to computing $p(a_1, a_2, z)$, $\forall z \in S$, which reduces to computing $p(a_1, y, z)$, $\forall y, z \in S$ which reduces to knowing $p(x, y, z)$, $\forall x, y, z \in S$.

The complexity of the construction is propotional to S^t. In the worst case we need all parallel lines from all points in S^t. We reduce $S^t \to S^{t-1} \to \ldots \to S$. The computation time is

$$T = t \cdot S^t + \{\text{interpolation time}\}$$

The size of the non-deterministic circuit is

$$\text{Size} = S^t(m^c + |\mathcal{F} \log \mathcal{F}|)$$

Since $t = \frac{2c}{\delta}$, $m, \mathcal{F} \approx O(2^{\frac{\delta l}{2c}})$, $|S| = m^\epsilon$, we have

$$\text{Size} = 2^{\epsilon l} 2^{\frac{\delta l}{2}} \approx 2^{\delta l}, \quad \text{for } \epsilon = \frac{\delta}{2}$$

So, we proved that if the set we had constructed is not a hitting set, then we have a small circuit to compute f. This concludes the proof of the theorem. $\square$

Note that in order to achieve the derandomization, we used a hard worst-case complexity and not hard average-case complexity. Also, we used the same kind of polynomials as in other constructions, but the evaluation was just on axis-parallel lines and not any more sophisticated subsets of inputs.