Instructor: Andrej Bogdanov **Notes by:** Hongyi Yao and Jing Xiao

We don't know whether P equals NP, we only know P $\in$ P/poly, and we want to show NP $\notin$ P/poly which we have no idea how to do. One reasonable approach is to start with some circuit model that is simpler than general polynomial size circuits and try to prove *circuit lower bounds* for this model — namely prove that explicit functions (like SAT) cannot be computed by circuits in this model. In this section, some hardness results about constant depth circuit will be presented.

1 Constant depth circuits

One restricted model of circuits we can look at are *constant depth circuits*. We define the *depth* of a circuit as the maximum number of AND and OR gates on any paths from input to output.

The reason we do not count NOT gates towards depth is that NOT gates can be pushed all the way down to the first level using the identities $\overline{x \vee y} = \overline{x} \wedge \overline{y}$ and $\overline{x \wedge y} = \overline{x} \vee \overline{y}$.

Now we will define constant depth circuit AC^0.

Class AC^0 Class of all decision problems that are decided by circuit families of polynomial size, constant depth, and unbounded fanin.

What can AC^0 circuits compute? If we do not count the NOT gates level, the depth 2 circuits compute CNF (AND of ORs), or DNF (OR of ANDs). However they can also compute some not so obvious functions, like "approximate majorities". These are functiosn of the following type: On input $(x_1, x_2, ..., x_n) \in \{0,1\}^n$,

$$\mathrm{APXMAJ}(x_1, x_2, ...x_n) = \begin{cases} 1, & \text{if } \sum_{i=1}^n x_i \geq \frac{2}{3}n \\ 0, & \text{if } \sum_{i=1}^n x_i \leq \frac{1}{3}n \end{cases}$$

However, there are also many things that we believe AC^0 circuits cannot compute. The main job of this lecture is to prove the decision problem PARITY is not in AC^0.

PARITY $\mathrm{PARITY}(x_1, x_2, ...x_n) = x_1 + x_2 + \cdots + x_n \pmod 2$

Theorem 1. $\mathrm{PARITY} \notin \mathrm{AC}^0$

The proof of theorem is divided into two parts:

Lemma 2. *For every circuit C of size s and depth d, there exists a polynomial $P : \{0,1\}^n \to \mathbb{R}$ of degree $O((\log s)^{2d})$, s.t $\Pr_{x \in \{0,1\}^n}[C(x) = P(x)] \geq 0.99$.*

Lemma 3. *For any polynomial P of degree less than $\sqrt{n}$, we have* $\Pr_{x\sim\{0,1\}^n}[\text{PARITY}(x) = P(x)] < 0.99$.

So in particular, if C computes parity then by comparing the degrees of the approximating polynomials P we have $(\log s)^{2d} = \Omega(\sqrt{n})$, or $s = \Omega(n^{1/4d})$. So there cannot be a polynomial-size circuit family that computes parity.

In this lecture we prove Lemma 2 above.

2 Approximating circuits by polynomials

We begin by approximating a single gate in the circuit by a polynomial, say an OR gate. The approximation will be randomized. Let us define what it means for an OR gate to be approximated by a random polynomial.

Definition 4. *We say a random polynomial $P : \{0,1\}^n \to \mathbb{R}$* ϵ-approximates $\text{OR}(y_1, y_2, ...y_k)$ *if for every input $(y_1, y_2, ...y_k)$, we have:*

$$\Pr_P[P(y_1, y_2, ...y_k) = \text{OR}(y_1, y_2, ...y_k)] \geq 1 - \epsilon$$

The random polynomial is allowed to take arbitrary real values; all that we require is that for any input y, the value of $P(y)$ matches $\text{OR}(y)$ with probability $1-\epsilon$ (so in particular it will be in $\{0,1\}$ in such a case). Note that the "approximation" $P(y) = 1$ for all y is not good under this definition, since it fails to give the correct answer on the specific input 0, even though this is the only such input.

One approximation that always works is

$$P(y_1, \ldots, y_k) = 1 - (1 - y_1) \ldots (1 - y_k)$$

but the degree of this approximation is too high. By considering random polynomials, we can make the degree much lower at the expense of paying a small error ϵ.

We begin by constructing a random polynomial of degree $O(\log k \cdot \log 1/\epsilon)$ that ϵ-approximates OR.

To do this, we guess at random the value of $t = \sum y_i$ approximately as follows: First we choose T to be a random from the set $\{1, 2, ..., 2^{\lg k}\}$. Then we choose S as a random subset by selecting each element in $\{1, 2, ...k\}$ independently with probability $\frac{1}{2T}$.

Define $P(y_1, y_2, ...y_k) = \sum_{i\in S} y_i$. Notice that if $y = 0$, then $P(y)$ is always zero. We show that when one of the y_is is 1, then P has a noticeable probablity of being 1. Then we will show how to make this noticeable probability very close to 1.

Let A denote the condition that $T = 2^{\lfloor \log t \rfloor}$, namely T is the largest power of 2 below t. Then for

every $y = (y_1, \ldots, y_k)$ we have

$$\begin{aligned}
\Pr_P[P(y) = \mathrm{OR}(y)] &= \Pr_P\Big[\sum_{i \in S} y_i = 1\Big] \\
&\geq Pr\Big[\sum_{i \in S} y_i = 1 \mid A\Big] \cdot \frac{1}{\log k} \\
&\geq \frac{t}{2T} \cdot \Big(1 - \frac{1}{2T}\Big)^{t-1} \cdot \frac{1}{\log k} \\
&\geq \frac{1}{8 \log k}.
\end{aligned}$$

We now want to make the quantity $1/8 \log k$ much closer to 1. For this we run the above experiments $m = O(\log k \cdot \log 1/\epsilon)$ times independently to obtain polynomials $P_1, P_2, ...P_m$ and define

$$P(y) = 1 - (1 - P_1(y)) \ldots (1 - P_m(y)).$$

Notice that if $y = 0$, then again $P(y)$ is always zero. On the other hand, if at least one of the y_i equals 1, then

$$\begin{aligned}
\Pr[P(y) = 1] &\geq \Pr[P_1(y) = 1 \text{ for some } i] \\
&\geq 1 - \Big(1 - \frac{1}{8 \log k}\Big)^m \\
&\geq 1 - \epsilon
\end{aligned}$$

by our choice of m. So for any input y, with probability $1 - \epsilon$ over the choice of P, $P(y) = \mathrm{OR}(y)$. Also notice that P is a polynomial of degree $m = O(\log k \cdot \log 1/\epsilon)$.

This solves the problem of approximating OR gates. To approximate AND gates, we can use the polynomial $Q(y_1, \ldots, y_k) = 1 - P(1 - y_1, 1 - y_2, ..1 - y_k)$. Now to approximate a whole circuit by a polynomial, we choose random approximations for each OR and AND gate in the circuit separately. Let's call the resulting polynomial P_C.

How large is the degree of P_C? Let s be the size of C, and let's layer the gates of the circuit according to their distance from the output gate. Then the gates closest to the bottom (near the inputs) are approximated by polynomials of degree at most $O(\log s \cdot \log 1/\epsilon)$ — since each gate can take no more than s inputs. (Recall that the number of inputs of a circuit count towards its size.) At the next layer, each gate is approximated by a polynomial of degree $O(\log s \cdot \log 1/\epsilon)$, but now each of its (at most) s inputs is itself a polynomial of degree $O(\log s \cdot \log 1/\epsilon)$, so the total degree is bounded by $O((\log s \cdot \log 1/\epsilon)^2)$. Continuing in this way, we have that the degree of the approximating polynomial for the output gate is $O((\log s \cdot \log 1/\epsilon)^d)$.

It remains to see that P_C indeed approximates C on a 99% fraction of inputs. For any particular input x, $P_C(x) \neq C(x)$ only if for some of the gates g in the circuit, the approximation of the corresponding polynomial on that gate fails, an event of probability $1 - \epsilon$. Taking a union bound over all s gates, we have that for every $x \in \{0,1\}^n$,

$$\Pr_{P_C}[P_C(x) = C(x)] \geq 1 - s \cdot \epsilon,$$

so if we choose $\epsilon = s/100$, it follows that for every x,

$$\Pr_{P_C}[P_C(x) = C(x)] \geq 0.99.$$

Note that this makes P_C be of degree $O((\log s)^{2d})$. In particular, for a random x,

$$\mathrm{E}_{P_C}\left[\Pr_{x\sim\{0,1\}^n}[P_C(x) = C(x)]\right] = \Pr_{x,P_C}[P_C(x) = C(x)] \geq 0.99$$

so there must exist a specific polynomial P_C such that $\Pr_x[P_C(x) = C(x)] \geq 0.99$.