ITCS:CCT09 : Computational Complexity Theory Apr 22, 2009

Lecture 13

Lecturer: Jayalal Sarma M.N. *Scribe: Xiaohui Bei*

In the previous lecture, we proved the depth lower bounds for parity function using the technique of random restrictions. In this lecture, we are going to give an alternate proof of parity $\notin \mathsf{AC}_0$ using a different technique called the polynomial method developed by Ronen Smolensky. This technique, in fact, yields a stronger result as follows.

Theorem 1 (Razborov-Smolensky theorem) *If a boolean circuit (which use* $\wedge,\vee,\neg$ *and MOD3 gates) of size* S *and depth* d *computes parity, then* $S \geq 2^{\Omega(n^{1/2d})}$.

Clearly the theorem implies parity is not in AC_0. In order to prove this theorem, again we want to come up with a property that is satisfied for all circuits of small size and constant depth, which parity function does not have. Before giving this property, we first introduce the concept of polynomial representation for circuits.

Informally speaking, we say a circuit function $f : \{0,1\}^n \to \{0,1\}$ is computed by a polynomial $g : \{0,1\}^n \to \mathsf{R}$ if $f(x) = g(x)$ for all $x \in \{0,1\}^n$. For example, consider polynomials over F_2, then the parity function can be represented as $\mathrm{PARITY}(x) = x_1 + x_2 + \ldots x_n$, and the degree of this polynomial is 1.

Now we consider the polynomials over field $\mathrm{GF}(3) = \{-1, 0, 1\}$. Here first notice that for every boolean function $f : \{0,1\}^n \to \{0,1\}$ and for any input $x = \{0,1\}^n$, if we let $y_i = 1 - 2x_i$ for all $1 \leq i \leq n$, and let function $\hat{f} = 1 - 2f$, then $\hat{f}(y)$ will be a function mapping $\{-1,1\}^n$ to $\{-1,1\}$. It has the same functionality with function f if we map 0 to 1 and map 1 to -1. Also we have $deg(f) = deg(\hat{f})$, where $deg(f)$ is the degree of a polynomial that represent f. Thus if we only care about the degree of a polynomial, we can always assume the variables are in $\{-1,1\}$ instead of $\{0,1\}$.

Then it's easy to see that parity function is computed by $f(x) = \prod_{x=1}^{n} x_i$. The degree of $f(x)$ is n. But for MOD3 gate, we know that $\mathrm{MOD}_3(x) = \sum_{i=1}^{n} x_i$ which has degree 1. The difference here suggest us that there might be a seperation between parity function and polynomials with low degrees.

However, there isn't a obvious degree upperbound that you can prove directly for the circuits which uses AND and OR gates. In order to tackle this general case, we will consider the following "relaxed" representation of a function:

Definition 2 *Let* $f(x)$ *be a boolean function and* $w_1, \ldots, w_m$ *are* m *random bits. We say*

that f is computed by a random polynomial $p(x, w_1, \ldots, w_m)$ if $f(x) = p(x)$ with probability no less than $1 - \frac{1}{n^{\omega(1)}}$ when $w_1, \ldots, w_m$ are chosen independently uniform from $\{0,1\}$.

Definition 3 *Let $f(x)$ be a boolean function. We say that f is approximated by a polynomial $p(x)$ if there exist a subset $S \subseteq \{0,1\}^n$, such that $|S| \geq (1 - \frac{1}{n^{\omega(1)}}) \cdot 2^n$ and $\forall x \in S, p(x) = f(x)$.*

Having these concepts, the proof of *parity* $\notin \mathsf{AC}_0$ will preceed in the following flow:

First assume by contradiction that *parity* $\in \mathsf{AC}_0$. We prove the following:

(1) Parity can be computed by a random polynomial of degree $o(n)$.

(2) Parity can be approximated by a polynomial of degree $o(n)$.

(3) Every boolean function can be approximated by a polynomial of degree $o(n) + \frac{n}{2}$.

Then we will use a counting argument to show that (3) cannot happen. Thus we will reach a contradiction.

Proof of (1): Let C be the AC_0 circuit that computes *parity*. We will construct the polynomial by induction on the height h of the circuit. When $h = 0$, each "gate" is just an input variable x_i, and we use degree 1 polynomial x_i to represent it. Now suppose that for all gates up to height $h - 1$, we have construct a polynomial to approximate it. And let g be a gate at height h.

- If g is a NOT gate, and the input of this gate is approximated by a polynomial f, we use $\hat{f} = 1 - f$ as the approximate polynomial for g. The degree of $\hat{f}$ is the same as f and we produce no new error here.
- If g is a MOD3 gate, and the inputs of this gate are approximated by $f_1, \ldots, f_k$. We use $\hat{f} = \sum_{i=1}^{k} f_i$ as the new polynomial for g. It's easy to see that the degree remains the same and we produce no new error.
- If g is an OR gate. We need to be more careful here. Suppose $f_1, \ldots, f_k$ are the inputs. A very naive approach would be let $\hat{f} = 1 - \prod_{i=1}^{k} (1 - f_i)$. But the degree is too high. So we need to introduce some randomness to make the degree lower, at the expense of paying a small error ϵ.

  Let $P_j = (\sum_{i=1}^{k} \alpha_i^j f_i)^2$, here $\alpha_1^j, \alpha_2^j, \ldots, \alpha_k^j$ are all random bits. Using some calculations, one can get that

$$\Pr[OR(f_1, \ldots, f_k) \neq P_j(f_1, \ldots, f_k)] \leq \frac{1}{3}$$

Then let polynomial $P = (1 - \prod_{j=1}^{t}(1 - P_j)$, and we can get $\Pr[OR \neq P] \leq (\frac{1}{3})^t$. If we choose t to be $(\log s)^c$ for some constant c. The error probability will be bounded by $(1/3)^{(\log s)^c}$ and the degree of this polynomial will be at most $(\log s)^c$ times the maximum degree of $f_1, \ldots, f_k$.

- If g is an AND gate. We can use De Morgan's law to convert it to the OR gate case.

Applying the above approach for each gate will give us an polynomial for the output gate with degree no more than $(\log n)^{cd}$, where d is the depth of the entire circuit. Thus the degree is $o(n)$ and the probability that this polynomial produce the right answer will be no less than $1 - S \cdot (1/3)^{(\log n)^c} \geq 1 - \frac{1}{n^{\omega(1)}}$.

Proof of (1) → (2): In order to prove this, it is sufficient to show that for any $0 \leq p \leq 1$, given a randomized polynomial f that represent the circuit with correct probability no less than p, one can construct a deterministic polynomial g that agree with this circuit with no less than $p \cdot 2^n$ inputs.

Suppose that there are m random variables $w_1, \ldots, w_m$ in f. So there are totally 2^m possible assignments for these random variables. And there are 2^n assignments for the input variables $x_1, \ldots, x_n$. Consider the following table in which each row represents one assignment for random variables and each column represents one assignment fot the input variables.

	Assignment for x_i				
	σ_1	σ_2	σ_3	$\cdots$	σ_{2^n}
a_1					
a_2					
a_3					
$\vdots$					
a_{2^m}					

For an assignment for random variables $a_j \in \{0,1\}^m$ and an assignment for input variables $\sigma_i \in \{0,1\}^n$, we place a 1 in the cell (a_j, σ_i) if $f(\sigma_i, a_j)$ is equals to the result of the circuit with input σ_i. By assumption we know that each column has no less than p fraction of 1's. Thus the total fraction of 1's within the entire table is no less than p. It follows that there must be some row that has a 1 in at least p fraction of its cells. Let this row be r and $g = f(x_1, \ldots, x_k, r)$ be the polynomial corresponding to this row. Then g must agree with the circuit with no less than $p \cdot 2^n$ inputs.

Proof of (2) → (3): Here we will prove the following lemma:

Lemma 4 *If parity$(x_1, \ldots, x_n)$ can be computed by a polynomial $p(x_1, \ldots, x_n)$ of degree d, then every boolean function $f : \{-1,1\}^n \to \{-1,1\}$ can be represented by a polynomial of degree $\frac{n}{2} + d$.*

Proof We write a DNF for function f by looking at its truth table. And we replace each variable $\overline{x_i}$ by $(1-x_i)$. This will give us a polynomial p of degree at most n that represents f. We can write this polynomial in form of a sum of monomials $p = \sum_{I\subseteq[n]} C_I(\prod_{i\in I} x_i)$. Notice that $x_i \in \{-1, 1\}$, thus $x_i^2 = 1$. Now consider any of its monomial terms $\prod_{i\in I} x_i$ with degree $|I| > n/2$. We can rewrite it as

$$\prod_{i\in I} x_i = \prod_{i\in I} x_i \prod_{i\in \overline{I}} x_i^2 = \prod_{i=1}^{n} x_i \prod_{i\in \overline{I}} x_i$$

which will have the same value as $parity(x_1, \ldots, x_n) \prod_{i\in\overline{I}} x_i$. Thus every monomial in this polynomial has degree at most $\frac{n}{2} + d$. ■

Having this lemma, it's easy to see that if $parity(x_1, \ldots, x_n)$ can be approximated by a polynomial of degree d, then every boolean function f can be approximated by a polynomial of degree $\frac{n}{2} + d$, thus (3) holds. □

Now we will use a counting argument to show that (3) cannot happen. First consider the number of polynomials of degree at most $n/2 + o(n)$, we know that

$$\#polynomials \leq 3^{\sum_{d=1}^{\frac{n}{2}+o(n)} \binom{n}{d}} = 3^{o(2^n)}$$

And these polynomials can differ from the desired function p on at most $k = \frac{2^n}{n^{\omega(1)}}$ inputs. Thus for one polynomial, it can approximate at most

$$\sum_{i=1}^{k} \binom{2^n}{i} \cdot 2^i \leq \sum_{i=1}^{k} \frac{(2^n)^i}{i!} \cdot 2^i \leq k \cdot 2^{(n+1)k} = 2^{o(2^n)}$$

So the total number of functions that these polynomials can approximate is no more than $3^{o(2^n)} \cdot 2^{o(2^n)} = 2^{o(2^n)}$. But the total number of boolean functions is 2^{2^n}. Which gives us a contradiction. Thus the theorem is proved.