6.841 Advanced Complexity Theory February 22, 2005

Lecture 5

Lecturer: Madhu Sudan *Scribe: Alissa Reyzin*

During the last lecture we used the Switching Lemma to show that PARITY is not in AC_0. This time we show a different proof of this fact that is due to Razborov and Smolensky ('87). As a reminder, we defined AC_0 as the class of constant depth, polynomial size circuits that have unbounded fan-in for the OR and AND gates.

Proof Idea

Assume that some circuit is too complex to analyze directly. Then we can try to approximate the function computed by that circuit by replacing the AND and OR gates by some approximations, which creates a simpler circuit. We want to show that parity is complex even to approximate. We now need to choose appropriate definitions of simplicity, complexity and approximation to make this proof idea work.

We want to define a simple function as one that can be computed by a low degree polynomial and a complex function as one that requires a high degree polynomial. That is, we think of arithmetic operations as being the gates in our circuit, rather than the standard AND and OR gates.

We first try to work with polynomials over GF(2) = $\{0, 1\}$. This gives us

AND(x,y) $= xy \mid$ AND$(x_1, x_2 ... x_n) = x_1 x_2 ... x_n$

PARITY(x,y) $= x + y \mid$ PARITY$(x_1, x_2 ... x_n) = x_1 + x_2 + ... + x_n$

However, that means that PARITY is a degree one polynomial! Since we are looking to show that PARITY is complex, using degree of polynomials in $\mathbb{Z}_2$ will not work.

What if we tried using a field of three elements so that everything wasn't automatically taken modulo two? Let's use GF(3) $= \{-1, 0, +1\}$. (We could just as well use $\{0, 1, 2\}$ but this is less convenient for our purposes). Now we have

$$\mathrm{PARITY}(x_1, x_2 ... x_n) = \prod_{i=1}^{n} x_i$$

where $x_i \in \{1, -1\}$. which gives us PARITY as a high degree function. Unfortunately, if simply use the degree of the corresponding polynomial, then AND and OR are also "complex". But since AND and OR can both be computed by a single gate, this does not provide us with a useful notion of complexity. Therefore we will need to find some way of approximating AND and OR with lower degree polynomials. ■

Lemma 1 *If $f : \{0,1\}^n \to \{0,1\}$ is computed by a circuit with depth d and size s, then there exists a set $S \subseteq \{0,1\}^n$ such that $|S| \geq \frac{3}{4}2^n$ and a polynomial p over GF(3) of degree $(\log s)^{O(d)}$ such that for every $(x_1 ... x_n) \in S$, $f(x_1 ... x_n) = p(x_1 ... x_n)$*

In essence, this lemma states that all AC^0 circuits are approximated by by simple functions. That is, there is a low degree polynomial which will get the same answer as the original circuit for $\frac{3}{4}$ of the inputs. The number $\frac{3}{4}$ is important here. For example, if we stated that we could find the correct output for $\frac{1}{2}$ of the inputs, the statement would be meaningless, since the constant 0 or 1 functions would do this.

Lemma 2 *If there exists a degree D polynomial $GF(3)^n \to$GF(3) and a set $S \subseteq \{0,1\}^n$ such that $p(x) =$ PARITY(x) for all $x \in S$ then every boolean function $g : S \to \{0,1\}$ is computed by a degree $\frac{n}{2} + D$ multilinear polynomial over GF(3).*

Lemma 3 *Any set of functions generating all functions from S to $\{0,1\}$ must have cardinality $\geq |S|$.*

Proof of Theorem: We assume, for the sake of contradiction, that PARITY has a depth D size s circuit.

- By Lemma 1, we can compute PARITY on a set S such that $|S| \geq \frac{3}{4}2^n$ using a polynomial of degree $(\log s)^{O(D)}$.
- By Lemma 4, we know that for every boolean function on the set S there is polynomial of degree $\frac{n}{2} + (\log s)^{O(D)}$.
- We know that all polynomials are just linear combinations of monomials. But we also no that there is a limited number of monomials of degree $\leq \frac{n}{2} + D$. More precisely can bound the number of monomials N as:

$$N \leq \sum_{i=0}^{\frac{n}{2}+D} \binom{n}{i}$$

$$N \leq \sum_{i=0}^{\frac{n}{2}} \binom{n}{i} + \sum_{i=\frac{n}{2}+1}^{\frac{n}{2}+D} \binom{n}{i}$$

$$N \leq 2^{n-1} + D\left(\frac{2^n}{\sqrt{n}}\right)$$

We know therefore, by Lemma 3, that $2^{n-1} + D\left(\frac{2^n}{\sqrt{n}}\right) \geq |S| \geq \frac{3}{4}2^n$. If $|S|$ were less than exponential$(n^{O(d)})$, then we would have $N < \frac{3}{4}2^n$, which is impossible according to the above. So, there is no circuit of polynomial size and constant depth that calculates PARITY.

■

Proof of Lemma 3 We view the set of functions $f : S \to 0,1$ as vectors of size $|S|$ over $\{0,1\}$. Suppose that vectors f_1 through f_k can generate every boolean function. Then, in particular, they can generate all the unit functions $\{\delta_x\}_{x \in S}$ where $\delta_x(y) = 1$ if and only if $x = y$. Clearly, the vectors corresponding to these functions are all linearly independent of each other. Therefore, we have $k \geq |S|$, and we are done. ■

Proof of Lemma 2 The idea of this proof is to switch back and forth between functions of $\{0,1\}$ and functions on $\{1,-1\}$. So, given a function $f : \{0,1\}^n \to \{0,1\}$ we define $\hat{f} : \{1,-1\}^n \to \{1,-1\}$ as the translation of f.

$$\hat{f}(y_1...y_n) = -1 \text{ iff } f(x_1...x_n) = 1 \text{ where } y_i = -1 \Leftrightarrow x_i = 1$$

We claim that the degrees of f and $\hat{f}$ will be the same since the mapping between the two functions is linear. We have:

$$\begin{aligned} \hat{f} &= 1-2f \\ y_i &= 1-2x_i \\ \hat{f}(y_1, y_2 ... y_n) &= 1-2f(\frac{1-y_1}{2}, \frac{1-y_2}{2} ... \frac{1-y_n}{2}) \\ \deg(f) &= \deg(\hat{f}) \end{aligned}$$

We define $\text{PARITY}(x_1 ... x_n) = \sum x_i \bmod 2$ and $\widehat{\text{PARITY}}(x_1 ... x_n) = \prod x_i$

Assume that PARITY can be approximated by a low degree polynomial. More formally, we let $T \subseteq \{-1,1\}^n$ such that $\widehat{\text{PARITY}}$ is equal to $q(y_1 ... y_n)$ on T and the degree of $q \leq D$. We take $\hat{f}(y_1 ... y_n)$ to be some mapping $T \rightarrow \{1,-1\}$. We know that we can express $\hat{f}$ as a polynomial $p(y)$, and therefore as a summation of monomials. We separate the monomials into the sets A and B where the terms in A have total degree less than $\frac{n}{2}$ and those in B have degree greater than or equal to $\frac{n}{2}$. Then let

$$\begin{aligned} p_1(y_1 ... y_n) &= \sum_i \alpha_i A_i \\ p_2(y_1 ... y_n) &= \sum_i \beta_j B_j \end{aligned}$$

We let $C_j = \prod_{i=1}^n y_i / B_j$ We then know that each monomial $B_j = \text{PARITY}(y_1 ... y_n) C_j$. Since this is the case, we get:

$$\begin{aligned} p_3(y_1 ... y_n) &= \sum_{i=1}^n \beta_j C_j \\ \hat{f}(y_1 ... y_n) = p_1(y_1 ... y_n) &+ \widehat{\text{PARITY}}(y_1 ... y_n) * p_3(y_1 ... y_n) \end{aligned}$$

Since both p_1 and p_3 are polynomials of degree no more than $n/2$ and $\widehat{\text{PARITY}}$ has degree D, the total degree of $\hat{f}(x)$ can be no more than $\frac{n}{2} + D$ and therefore, the degree of $f(x)$ can not be more than $\frac{n}{2} + D$ either. ■

Proof of Lemma 1 The idea of this proof is to probabilistically replace every gate by a low degree polynomial that gives the correct output almost all of the time. This will give us a circuit of low degree overall which will still output the correct result with high probability. Here is the proof summary:

- For the sake of simplicity we can assume that our circuit contains no AND gates. This can be done with no loss of generality, since using DeMorgan's law we can convert any circuit containing AND gates to one containing only OR and NOT gates with only a constant factor increase in size and depth.
- We need to create low degree polynomials that approximate the NOT and OR gates. For the NOT gate, we simply use $p = 1 - x$ which clearly gives the correct value when $x = 1$ or $x = 0$. We need not worry about $x = -1$.

- For each OR gate, we need to randomly generate some polynomial of degree $\log s$ that will calculate the correct value most of the time. While simply using the constant 1 would be a reasonable approximation at the base level of our circuit, this becomes unreasonable as we look higher up in the circuit and the distribution of inputs becomes non-uniform.
- We then show that for each gate that we replace by a polynomial, the probability that the polynomial computes the correct value is at least $1 - \frac{1}{4s}$.
- From this, we can use the Union bound to demonstrate that since the probability of an error in each gate is at most $\frac{1}{4s}$, the probability of an error in the entire circuit is no more that $\frac{1}{4}$ and therefore our approximation works for at least $\frac{3}{4}$ of the inputs.
- We can then show that the total degree of our approximation polynomial is no more than $\log(s)^{O(d)}$.

We now describe the resulting polynomial that represents the circuit. (This section is taken verbatim from a previous year's lecture notes).

In order to see what the function computed from the "poly-replaced" circuit looks like, we notice that throughout the circuit we are replacing OR gates (with fan-in of size k) with polynomials of degree $\log s$ (in $x_1, x_2 ... x_k$). At the lowest level of the circuits, where all the inputs to the gates (polynomials) are constants in $\{0, 1\}$, we have polynomials of degree $\log s$. At the second level, where the inputs to the polynomials are themselves polynomials of degree $\log s$, we will have polynomials of degree $\log^2 s$. In general, if we have polynomials of degree d_1 and all its inputs are polynomials of degree d_2, the output will be a polynomial of degree $d_1 d_2$ (can be proved using induction). So continuing all the way to depth d, we see that the resulting polynomial (which computes the output of the circuit) has degree $\leq (\log s)^d$.

This polynomial will not be computing the correct value for all inputs. However, for a fixed input and an *independent*, random choice of polynomials to replace the OR gates, we will show that the probability that a (replaced) gate computes the wrong result is at most $\frac{1}{4s}$. By the union bound, we get the right values throughout the circuit with probability $\geq 3/4$. Notice that this requires that the choice of polynomials be made *independently* of the choice of inputs (i.e. the polynomials were not chosen to suit the input, or the other way around).

Since for a particular input we have a 3/4 probability of getting the right result when the replacing polynomials (not necessarily the same for all gates) which produces the correct result for at least 3/4 of all possible inputs to the circuit.

We now need only to find this $\log s-$degree polynomials to construct our OR gates. One way to approximate an OR gate is by considering the sum of a random subset of its inputs. That is:

$$p(z_1 ... z_m) = \left(\sum_{i=1}^{m} \alpha_i z_i \right)^2$$

The reason we square the resulting sum is to remove the possibility of getting an output of -1. In this way, if all of the z_i are 0, then p is guaranteed to be 0. If at least one of the $z_i = 1$, then $p = 1$ with probability $\geq 2/3$. So how do we make sure that the probability of an error is only $\frac{1}{4s}$? We do this by observing that if we compute p on several random subsets of variables and then OR together the results of these computations, our probability of making a mistake will decrease. Therefore, our OR-replacement polynomial will be

$$\text{OR-P}(p_1(z_1...z_m), p_2(z_1...z_m)...p_k(z_1...z_m))$$

where OR-P is the exact OR polynomial. If we pick $k = \log s$, we then get a polynomial that is of degree $2 \log s$ and will only make a mistake with probability $(1/3)^{\log s}$ which is less than our desired result of $\frac{1}{4s}$.

We can repeat this process independently for every OR gate in the circuit in order to generate the polynomial that represents the circuit. ■