6.841 Advanced Complexity Theory Feb 23, 2009

Lecture 6

Lecturer: Madhu Sudan *Scribe: Michael Forbes*

The goal of this lecture is to give alternate proof of PARITY$\notin$ AC^0, following the outline of Razborov and Smolensky.

0.1 Probability Review

Probability focuses on the probability of events and random variables. There are several facts that will be used throughout the course. Consult the appropriate textbook for proof. As this is a computer science course, we will restrict ourselves to the simpler case of discrete events, and discrete random variables. The results do generalize but we need not go there.

Lemma 1 (Linearity of Expectation) *For random variables X and Y, and real numbers a,$b \in \mathbb{R}$, $\mathbb{E}[aX+bY] = a\mathbb{E}[X] + b\mathbb{E}[Y]$.*

It is important to recall that this fact holds regardless of whether X and Y are independent or not.

Lemma 2 *For independent random variables X and Y, $\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y]$.*

Lemma 3 (Union Bound) *For events E_1 and E_2, $\Pr[E_1 \cup E_2] \leq \Pr[E_1] + \Pr[E_2]$.*

Recall that E_1 and E_2 are called *independent* if $\Pr[E_1 \cap E_2] \leq \Pr[E_1]\Pr[E_2]$. Two discrete random variables are said to be independent if their joint probability distribution decomposes into a product of marginal distributions. That is, the probabilities of the variables taking certain values is what you would expect.

Theorem 4 (Markov's Inequality) *If X is a non-negative random variable then $\Pr[X \geq k\mathbb{E}[X]] \leq \frac{1}{k}$.*

Theorem 5 (Chebychev's Inequality) *For random variable X with variance σ^2, $\Pr[|X - \mathbb{E}[X]| \geq k\sigma] \leq \frac{1}{k^2}$.*

Theorem 6 (Chernoff Bound) *For $X_1, \ldots, X_n$ independent, identically distributed (possibly continuous) random variables with expectation μ, such that $X_i \in [0,1]$,*

$$\Pr\left[\left|\frac{\sum_{i=1}^n X_i}{n} - \mu\right| \geq \varepsilon\right] \leq \exp(-\varepsilon^2 n)$$

0.2 Algebra

Finite fields are finite sets equipped with the usual notions of addition and multiplication. That is, a finite field $\mathbb{F}_q$ of size q is such that addition forms an abelian group, with an identity 0, and $\mathbb{F}_q \setminus \{0\}$ forms an abelian group with multiplication with identity element 1. These two operations are related through the distributive axiom. Abstract algebra gives us that there is a unique (up to isomorphism) finite field of size q if q is a prime power, and no finite field for other values of q. For q prime, these finite fields are the familiar structures of the integers modulo q. For most computer science purposes, the prime fields are enough.

A basic fact about finite fields is their relation with polynomials. If we recall Fermat's Little theorem, then we know that $x^p = x$ when working over $\mathbb{F}_p$ (for prime p). This leads us to think that no polynomial over $\mathbb{F}_p$ need have degree in a single variable above p.

Definition 7 *For a multivariate polynomial $p(x_1, \ldots, x_n)$, the degree is the largest number of variables, including multiplicity, seen in any term. Thus,* $\deg(x^2y^2) = 4$. *The degree in a single variable is the degree of the polynomial when considered only as a function in a single variable, with the other variables held constant. For example,* $\deg_x(x^2y^2) = 2$.

Proposition 8 *For any multivariate function $f : \mathbb{F}_q^n \to \mathbb{F}_q$, f can be expressed as a multivariate polynomial $p(x_1, \ldots, x_n)$ such that* $\deg_i(p) \leq q - 1$.

The proof is left as an **exercise**, with the hint of "counting", or perhaps even "vector spaces".

In the single variable case we can bound the number of roots using the degree. The following theorem shows we can do something similar in the multivariate case.

Theorem 9 (Schwartz-Zippel Lemma) *For a non-zero degree d polynomial over field $\mathbb{F}$ (possibly infinite), $p : \mathbb{F}^n \to \mathbb{F}$, and some finite set $H \subseteq \mathbb{F}$,*

$$\Pr_{\mathbf{x} \in H^n}[f(\mathbf{x}) = 0] \leq \frac{d}{|H|}$$

The proof is left as an **exercise**, with the hint of "induction on number of variables". The key thing to notice about the above theorem is that it is independent of n.

0.3 A First Cut

We examine the first idea of trying to work over $\mathbb{F}_2$, as this is the most natural field for computer science. We would want to prove that every circuit in AC^0 is computed by a low-degree polynomial. This means for a circuit family $\{C_n\}_n$ that for each input size n, there is a degree $d(n)$ polynomial that computes C_n, and the function $d(n)$ is somehow "small", or slow growing, as a function of n. We would then try to prove that PARITY can only expressed in a similar way for some $d(n)$ which is asymptotically large, that is it needs high-degree polynomials. Notice that we would never need to go above degree n as we can apply Fermat's Little Theorem to reduce the degree in each variable to at most 1.

However, this plan doesn't work. Consider the AC^0 function $\mathrm{AND}(x_1, \ldots, x_n) = \prod_{i=0}^n x_i$. We cannot find a polynomial p of degree less than n for otherwise we could take their difference and notice that it is zero on all inputs and by applying Schwartz-Zippel the polynomial must be formally-zero (in the sense that all of its coefficients are zero). (This isn't a strict implication of Schwartz-Zippel, but rather one must show something similar: if you have a polynomial $p : \mathbb{F}_q^n \to \mathbb{F}_q$ such that $\deg_i(p) < |H|$ for each i, p is identically zero on H^n, then p is formally zero. The proof is similar to that of Schwartz-Zippel, and is left as an **exercise**). This would violate the selection of p, so no such p can exist. This seems like a problem, as this is the highest degree we can expect from a polynomial over $\mathbb{F}_2$!

However, it gets even worse, as $\mathrm{PARITY}(x_1, \ldots, x_n) = x_1 + \cdots + x_n$, which is about the lowest-degree polynomial possible. Thus this plan does not work. To fix this, we can move to another field.

0.4 An Idea

We begin by noticing a common trick in Boolean analysis (the analysis of functions over boolean values). The additive group $\{0, 1\}$ can be put into a linear correspondence with the multiplicative group $\{-1, 1\}$, where the mapping is done via $x \mapsto 1 - 2x$, and inversely $y \mapsto \frac{1-y}{2}$. The usual mapping $x \mapsto (-1)^x$ is not as useful here as it is not linear.

This allows us to move from analyzing PARITY over $\{0, 1\}$ to over $\{-1, 1\}$, where instead we look at the parity of signs in the input. Notice that we can express this parity with the simple polynomial $\prod_{i=1}^n y_i$. Suppose we fix some finite field $\mathbb{F}$ with size at least three (so $-1 \neq 1$). Then suppose that the polynomial $p(x_1, \ldots, x_n)$ outputs the correct parity bit when its input is in from $\{0, 1\}^n$ (and we don't care what it does otherwise). Now we can construct $q = 1 - 2p(\frac{1-y_1}{2}, \ldots, \frac{1-y_n}{2})$ which, as we are working over the same field,

will compute the correct parity in the ± 1 sense. Notice that $\deg q \leq \deg p$. Further, using the Schwartz-Zippel-like result mentioned in the previous section, this means that because q and $\prod y_i$ agree on all inputs restricted to $\{\pm 1\}^n$ it must be that $\deg(\prod y_i) \leq \deg q$, and thus $\deg p \geq n$.

This establishes two points. First, that computing the parity over $\{0,1\}$ is equivalent to computing it over ± 1 because of the linear transformations. Second, using that the ± 1 domain is well-understood, we see that computing the parity exactly over the $\{0,1\}$ domain requires degree n polynomials. This seems like progress.

0.5 The Proof

To make the idea into a proof, we need another idea from Razborov from a previous work. The idea is to only approximate circuits by polynomials instead of requiring equality. To do this, we replace each gate by some approximate low-degree functions, and combine them to say that the entire function is approximated by a low-degree (but higher-degree than at the gate level) polynomial. Then, the plan should be that PARITY cannot be similarly approximated. The field we will use will be the simplest one that obeys the idea of the last section and this is $\mathbb{F}_3$.

We begin with approximating AC^0. The goal is to prove the following lemma, which makes precise what we mean by approximation.

Lemma 10 *Let $C : \{0,1\}^n \to \{0,1\}$ be computed by a depth d, size s, AC^0 circuit. Then*

1. *There exists a polynimal $p \in \mathbb{F}_3$ on n variables of degree at most $2^d(\log_3(s/\varepsilon)+1)^d$*
2. *There exists a set $S \subseteq \{0,1\}^n \subset \mathbb{F}_3^n$, where $|S| \geq (1-\varepsilon)2^n$ and $\forall \mathbf{x} \in S$, $p(\mathbf{x}) = C(\mathbf{x})$.*

Notice that AND is approximated in this sense by the zero-polynomial as this polynomial is only incorrect on one input, namely the all-ones vector. Thus, an adversary to this proof could somehow ensure that we are taking the AND of something that is always the all-ones vector and thus foil our plans of using this as subroutine in our circuit. To avoid this, we will use randomization. We will create a distribution of polynomials for each gate such that there is a non-zero probability that some polynomial approximates the gate well. We pass this distributions as the subroutine. By arguing that the probability of correct approximation doesn't reach zero we then have, by the probabilistic method, that some approximating polynomial must exist. In what follows, we assume that we only use NOT and OR gates. We can clearly do this without affecting the depth, and it will make the proof simpler. We now present the randomization.

Lemma 11 *Fix k. For each t, there exists a distribution on degree $2k$ polynomials p such that*

$$\forall z_1, \ldots, z_t \in \{0,1\}, \Pr_p[p(z_1, \ldots, z_t) \neq \mathrm{OR}(z_1, \ldots, z_t)] \leq \frac{1}{3^k}$$

Proof

We only concern ourselves here with inputs in $\{0,1\}$ as we do not care what happens otherwise. Notice that $\mathrm{OR}(z_1, \ldots, z_t) = 1 - \prod_{i=1}^t (1 - z_i)$ on boolean inputs. Now, pick $\alpha_1, \ldots, \alpha_t \in \mathbb{F}_3$ at random (uniform, independently). Consider $L_\alpha(z_1, \ldots, z_t) = (\sum_{i=1}^t \alpha_i z_i)^2$. We first claim that

$$\Pr_\alpha[L_\alpha(z_1, \ldots, z_t) = \mathrm{OR}(z_1, \ldots, z_t)] \geq 2/3$$

Consider the cases. If $z_1, \ldots, z_t = 0, \ldots, 0$ then with probability 1 the choice of α will be correct as the OR will always be 0 as will L_α.

In the other case, we want to get $\sum_{i=1}^t \alpha_i z_i \neq 0$, for when we then squared it shall become 1, which is the correct answer in this case. Consider $f(\alpha) = \sum_{i=1}^t \alpha_i z_i$. This is a linear, non-zero polynomial (it is non-zero because $\mathbf{z} \neq \mathbf{0}$, by hypothesis). Notice that we just pulled a switch: we are now considering the

α's as inputs and the z's as constants. Thus, we can apply Schwartz-Zippel to see that $\Pr_\alpha[f(\alpha) = 0] \leq \frac{1}{3}$. This gives the claim about L_α by squaring f.

Further, we can notice that this gives a one-sided error: if the answer is really 0, the L_α will also give zero. If the answer is 1, then with probability at least 2/3 a 1 will be returned. As usual with one-sided error, we can amplify the probability of success by taking multiple independent trials and taking the OR of them. Thus, we pick vectors $\alpha^{(1)}, \ldots, \alpha^{(k)}$ randomly and take the OR of $L_{\alpha^{(1)}}, \ldots, L_{\alpha^{(1)}}$. It might seem that we would want to use this approximate-OR in a recursive fashion, however we can use the *actual* $\mathrm{OR}(z_1, \ldots, z_t) = 1 - \prod_{i=1}^{t}(1 - z_i)$ in this case. As each $L_{\alpha^{(i)}}$ is of degree 2, this gives a polynomial p of degree $2k$. Notice that our random choices were only over the $\alpha^{(i)}$ and these determined the polynomial p. By repeating this one-sided error process we see that there is only at most a $\frac{1}{3^k}$ error that we make a mistake, and thus for any particular $z_1, \ldots, z_k$ the chance of $p(z_1, \ldots, z_k)$ making an error is also bounded by $\frac{1}{3^k}$. This is exactly the claim. ■

We have now shown that we can approximate the OR gate. By using this approximate gates in the AC^0 circuit, we can then approximate the whole thing. We can notice that NOT gates are very easy to implement by using $\mathrm{NOT}(z) = 1 - z$. This only works over the boolean domain, but don't forget that even though we are working over $\mathbb{F}_3$, the boolean domain is all that the polynomial needs to work over.

Proof of Lemma 10: Choose the smallest k such that $3^k \geq s/\varepsilon$. We will string together the circuit C using the distributions of approximate-OR gates from Lemma 11 and boolean NOT gates. By then union bounding the errors the result should follow.

For each internal node, we will construct a distribution of polynomials that will with high probability compute the correct value for that node. We start with the input nodes, where they have the monomial x_i representing input i. At at NOT gate, as argued above, we simply compute $1 - z$, where z is randomly chosen from the distribution on the child of the node. This does not affect the degree of the polynomial nor does it affect the probability of error on the distribution. For an OR gate with children $z_1, \ldots, z_t$, we randomly choose a polynomial p given by Lemma 11, and for each i, randomly sample a polynomial p_i from child i's distribution. We then create the polynomial $q = p(p_1(x_1, \ldots, x_n), \ldots, p_t(x_1, \ldots, x_n))$ for this node, with the appropriate probabilities based on its creation.

Notice that the only degree-increasing step is that of the OR gate, and this is done via composition. The polynomial p has degree $2k$ and so the degree of q is at most $2k \cdot \max_i \deg(p_i)$. By induction we can see this leads to polynomials of degree at most $(2k)^d$, as the circuit has depth d. Notice that by our choice of k, $k \leq \log_3(s/\varepsilon) + 1$, so these polynomials have degree at most $2^d(\log_3(s/\varepsilon) + 1)^d$.

Now we must analyze the probability that the polynomial p of the output node computes the correct output. Notice that p is computed by independently chosen polynomials at each node. The only source of error comes from the OR nodes, and there can be at most s of them. By the construction in Lemma 11, these choices error with probability at most $\frac{1}{3^k}$ regardless of what their input is. Therefore, the union bound indicates that polynomial p will error with probability at most $\frac{s}{3^k}$. By our choice of ε, this gives that

$$\forall \mathbf{x}, \Pr_p[p(\mathbf{x}) \neq C(\mathbf{x})] \leq \varepsilon$$

We now argue by the probabilistic method to assert the existence of a specific p to give the claim. By the above argument, any such p from this distribution has the desired degree. The probability analysis is left as an **exercise**.

■

We have just show that AC^0 can be approximated by low-degree polynomials. Specifically, as s is bounded by some polynomial in n, and d is a constant, we see that the depth of the polynomials is polylogarithmic (when ε is held constant). Thus, we only need to show that PARITY cannot be approximated by such a family of polynomials.

Lemma 12 *No degree $o(\sqrt(n))$ polynomial approximates* PARITY *in the sense that it correctly answers the question on a set of size at least $\frac{3}{4}2^n$.*

Proof Consider any set $S \subseteq \{0,1\}^n$ (it is meant to be "large") such that parity can be correctly computed on S by a polynomial p of degree D in the field $\mathbb{F}_3$. We now make the transformation to the ± 1 domain. Define the analogous set T to S such that $|T| = |S|$ and $T \subseteq \{\pm 1\}^n$. Define $q(y_1, \ldots, y_n) = 1 - 2p(\frac{1-y_1}{2}, \ldots, \frac{1-y_n}{2})$ of degree at most D. Notice that as before, q computes $\prod_{i=1}^n y_i$ on $T \subseteq \{-1,1\}^n$.

Now we want to count the number of functions from T to $\{-1,0,1\}$. It is easy to see that the number of function is just $3^{|T|}$. However, from the preliminaries we know that any function is also a polynomial. So we can count polynomials also as a way to count functions. The polynomials will be of the form $p(x_1, \ldots, x_n) = \sum_{\mathbf{d}} c_{\mathbf{d}} x_1^{d_1} \cdots x_n^{d_n}$. Notice that as we are dealing with $T \subseteq \{-1,1\}^n$, so we have the $x_i^2 = 1$, as $x_i \in \{\pm 1\}$. Thus, we can reduce all of the d_i modulo 2.

Now we have the following key idea. If $\sum_i d_i > \frac{n}{2}$, we replace

$$x_1^{d_1} \cdots x_n^{d_n}$$

by

$$x_1^{d_1+1 \pmod 2} \cdots x_n^{d_n+1 \pmod 2} q(x_1, \ldots, x_n)$$

. We first claim that these terms are in fact equivalent for inputs over T. To see this, recall that $q(x_1, \ldots, x_n) = \prod_{i=1}^n x_i$ over inputs in T by construction. Then, we use the reduction of degrees modulo 2 to get the algebraic equivalence. By doing this to each term, we see that the maximum term size is bounded by $\lfloor \frac{n}{2} \rfloor + D$. Further, we see that each term is in bijection with some subset of $\{1, \ldots, n\}$ of size at most its degree. Thus, there can be at most

$$\sum_{i=1}^{\lfloor n/2 \rfloor + D} \binom{n}{i} \leq \sum_{i=1}^{\lfloor n/2 \rfloor} \binom{n}{i} + D \binom{n}{\lfloor n/2 \rfloor}$$

And notice that $\sum_{i=1}^{\lfloor n/2 \rfloor} \binom{n}{i} \leq 2^{n-1}$ as the sets of size at most $\lfloor n/2 \rfloor$ can be paired up with their complements to get the entire set (if n is even then we would need to half count the pairs of sets each of size $n/2$). Apparently, left as an **exercise**, we also have that $\binom{n}{\lfloor n/2 \rfloor} \leq 2^n/\sqrt{n}$. So the overall bound on the number of terms is $2^{n-1} + \frac{D2^n}{\sqrt{n}}$. As each term can have one of the coefficients in $\mathbb{F}_3$, there are at most $3^{2^{n-1} + \frac{D2^n}{\sqrt{n}}}$ polynomials.

As this counts all polynomials over T, and thus all functions over T, we must have that $3^{|T|} \leq 3^{2^{n-1} + \frac{D2^n}{\sqrt{n}}}$, or that $|T| \leq 3^{2^{n-1} + \frac{D2^n}{\sqrt{n}}}$. As we had $|T| \geq \frac{3}{4} 2^n$ this implies that $D \geq \sqrt{n}$. ■

Theorem 13 (Razborov-Solemnsky) PARITY $\notin$ AC^0

Proof We saw that any AC^0 function can be approximated by a polynomial of polylogarithmic degree, while PARITY requires $\Omega(\sqrt{n})$. ■

It should be noted that the $\frac{3}{4}$ in the approximation of PARITY is important, as any constant function approximates parity on exactly half the inputs.

Observation 14 *As we were working over $\mathbb{F}_3$ we could have thrown in mod-3 gates into our circuits and we would have gotten the same results. Thus, adding mod-3 gates wouldn't help compute parity either.*

Conversely, it should be possible to show that computing mod-3 is hard for AC^0 *even given mod-2 gates.*

While these ideas seem to work for finite fields, the ideas really break down for mod-6 gates. Currently, it is not known that AC^0 with mod-6 gates fails to compute SAT. Some results from error-correcting codes suggest that this is in some way deep.

Observation 15 *I think it should be possible to get a similar result to Lemma 10 over any finite field of size bigger than 2.*