Notes for Lecture 11

Circuit Lower Bounds for Parity Using Polynomials

In this lecture we prove a lower bound on the size of a constant depth circuit which computes the XOR of n bits.

Before we talk about bounds on the size of a circuit, let us first clarify what we mean by circuit depth and circuit size. The depth of a circuit is defined as the length of the longest path from the input to output. The size of a circuit is the number of AND and OR gates in the circuit. Note that, for our purpose, we assume all the gates have unlimited fan-in and fan-out. We define **AC0** to be the class of decision problems solvable by circuits of polynomial size and constant depth. We want to prove the result that PARITY is not in **AC0**.

There are two known techniques to prove this result. In this class, we will talk about a proof which uses polynomials; in the next class we will look at a different proof which uses random restrictions.

1 Circuit Upper Bounds for PARITY

Before we go into our proof, let us first look at a circuit of constant depth d that computes PARITY.

Theorem 1 *For every constant $d \geq 2$, there are circuits of size $2^{O\left(n^{\frac{1}{d-1}}\right)}$ that compute parity.*

In the next lecture, we will prove a $2^{n^{\frac{1}{d-1}}}$ size lower bound, establishing the tightness of Theorem 1. Today we will prove a weaker $2^{n^{\frac{1}{4d}}}$ lower bound.

Proof: [Of Theorem 1] Consider the circuit C shown in Figure 1, which computes the PARITY of n variables. The circuit C is a tree of XOR gates, each of which has fan-in $n^{\frac{1}{d-1}}$; the tree has depth $d-1$.

Now, since each XOR gate is a function of $n^{\frac{1}{d-1}}$ variables, it can be implemented by a CNF or a DNF of size $2^{n^{\frac{1}{d-1}}}$. Let us replace alternating layers of XOR gates in the tree by CNF's and DNF's - for example we replace gates in the first layer by their CNF implementation, gates in the second layer by their DNF implementation, and so on. This gives us a circuit of depth $2(d-1)$. Now we can use the associativity of OR to collapse consecutive layers of OR gates into a single layer. The same thing can be done for AND to get a circuit of depth d.

This gives us a circuit of depth d and size $O(n2^{n^{\frac{1}{d-1}}})$ which computes PARITY. □

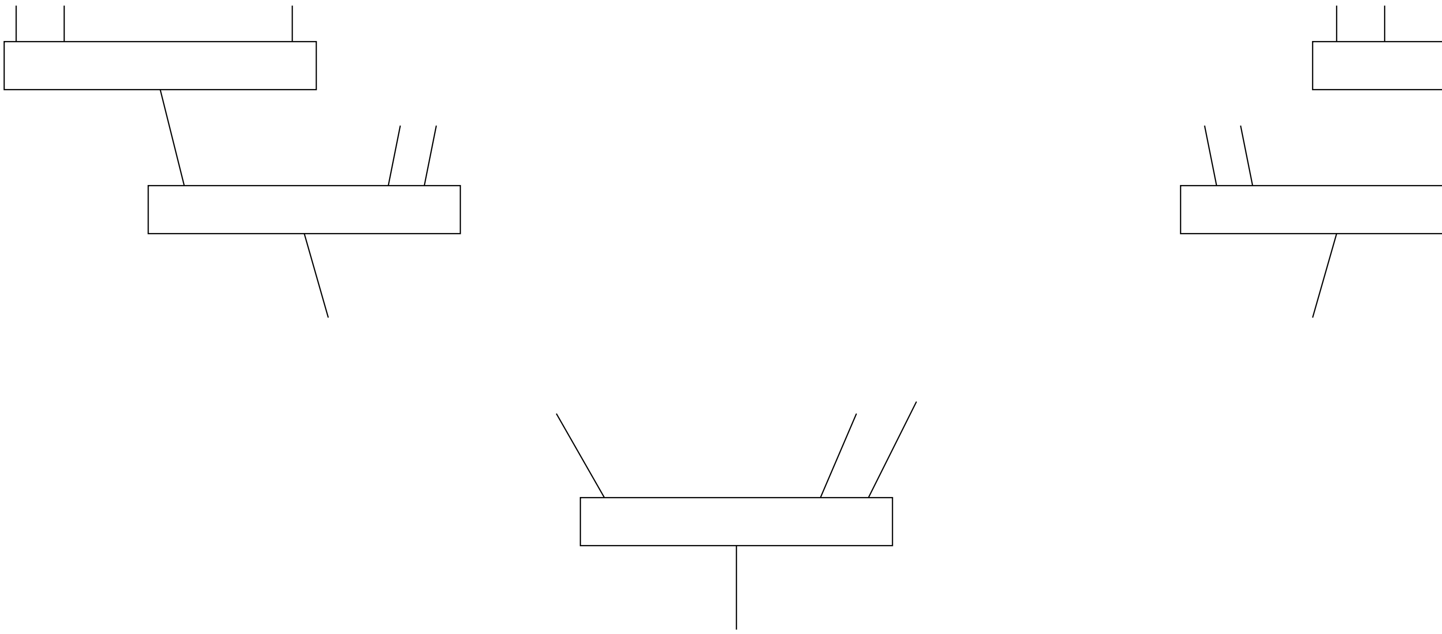

Figure 1: Circuit for Computing XOR of n variables; each small circuit in the tree computes the XOR of $k = n^{\frac{1}{d-1}}$ variables

2 Overview of the Lower Bound Proof

For our proof, we will utilise a property which is common to all circuits of small size and constant depth, which PARITY does not have.[1] The property is that circuits of small size and constant depth can be represented by low degree polynomials, with high probability. More formally, we show that if a function $f : \{0,1\}^n \to \{0,1\}$ is computed by a circuit of size s and depth d, then there exists a function $g : \{0,1\}^n \to \mathbb{R}$ such that $\Pr_x[f(x) = g(x)] \geq \frac{3}{4}$ and $\hat{g}_\alpha \neq 0$ only for $|\alpha| \leq O((\log S)^{2d})$, where $\hat{g}$ is the Fourier transform of g.

Then we will show that if a function $g : \{0,1\}^n \to \mathbb{R}$ agrees with PARITY on more than a fraction $\frac{3}{4}$ of its inputs, then there is a coefficient α such that $\hat{g}_\alpha \neq 0$ and $|\alpha| = \Omega(\sqrt{n})$. That is, a function which agrees with PARITY on a large fraction of its inputs, has to have high degree. From these two results, it is easy to see that PARITY cannot be computed by circuits of constant depth and small size.

We give a formal definition of *degree* of a function and then formally state the two results that give our lower bound.

Definition 2 *We say that a function $g : \{0,1\}^n \to \mathbb{R}$ has* degree at most d *if there is a polynomial over the reals of degree at most d such that g and the polynomial agree on $\{0,1\}^n$.*

[1] Incidentally, the property is false with high probability for random functions and it is computable in time $2^{O(n)}$ given the truth-table of a function. You may remember that this implies that our lower bound will be a *natural proof.*

An equivalent way of looking at the definition of degree is to consider the size of the largest non-zero coefficient of the Fourier transform of the function.

Fact 3 *A function $g : \{0,1\}^n \to \mathbb{R}$ has degree at most d if and only if $\hat{g}_\alpha = 0$ for all α such that $|\alpha| > d$.*

The following two lemmas are the main results of this lecture.

Lemma 4 *For every circuit C of size S and depth d, there is a function $g : \{0,1\}^n \to \mathbb{R}$ of degree $O((\log S)^{2d})$ such that g and C agree on at least a $3/4$ fraction of $\{0,1\}^n$.*

Lemma 5 *Let $g : \{0,1\}^n \to \mathbb{R}$ be a function that agrees with PARITY on at least a $3/4$ fraction of $\{0,1\}^n$. Then the degree of g is $\Omega(\sqrt{n})$.*

From Lemma 4 and Lemma 5 it is immediate to derive the following lower bound.

Theorem 6 *For every constant $d \geq 2$, if C is a circuit of depth d and size S that computes parity, then $S \geq 2^{\Omega(n^{1/4d})}$.*

Proof: From Lemma 4 we have that there is a function $g : \{0,1\}^n \to \mathbb{R}$ that agrees with PARITY on a $3/4$ fraction of $\{0,1\}^n$, and whose degree is at most $O((\log S)^{2d})$. From Lemma 5 we deduce that the degree of g must be at least $\Omega(\sqrt{n})$, so that

$$(\log S)^{2d} = \Omega(\sqrt{n})$$

which is equivalent to

$$S = 2^{\Omega(n^{1/4d})}$$

□

3 Proof of Lemma 4

Most of the work in the proof of Lemma 4 will be in showing how to give a "probabilistic approximation" of a single gate using low-degree functions.

3.1 Approximating OR

The following lemma says that we can approximately represent OR with a polynomial of degree exponentially small in the the fan-in of the OR gate. We'll use the notation that x is a vector of k bits, x_i is the ith bit of x, and $\mathbf{0}$ is the vector of zeros (of the appropriate size based on context).

Lemma 7 *For all k and ϵ, there exists a distribution G over functions $g : \{0,1\}^k \to \mathbb{R}$ such that*

1. *g is of degree $O((\log \frac{1}{\epsilon})(\log k))$, and*

2. for all $x \in \{0,1\}^k$,

$$\Pr_{g \sim G}[g(x) = x_1 \vee \ldots \vee x_k] \geq 1 - \epsilon. \tag{1}$$

The idea of the proof is that we want a random polynomial $p : \{0,1\}^k \to \mathbb{R}$ that computes OR. An obvious choice is

$$p_{\text{bad}}(x_1, \ldots, x_k) = 1 - \prod_{i \in \{1,\ldots,k\}} (1 - x_i)\,, \tag{2}$$

which computes OR with no error. But it has degree k, whereas we'd like it to have logarithmic degree. To accomplish this amazing feat, we'll replace the tests of all k variables with just a few tests of random batches of variables. This gives us a random polynomial which computes OR with one-sided error: when $x = \mathbf{0}$, we'll have $p(x) = 0$; and when some $x_i = 1$, we'll almost always (over our choice of p) have $p(x) = 1$.

Proof: We pick a random family F of subsets of the bits of x. (That is, for each $S \in F$ we have $S \subseteq \{1, \ldots, k\}$). We'll soon see how to pick F, but once the choice has been made, we define our polynomial as

$$p(x_1, \ldots, x_k) = 1 - \prod_{S \in F} \left(1 - \sum_{i \in S} x_i\right). \tag{3}$$

Why does p successfully approximate OR? First, suppose $x_1 \vee \ldots \vee x_k = 0$. Then we have $x = \mathbf{0}$, and:

$$p(0, \ldots, 0) = 1 - \prod_{S \in F} \left(1 - \sum_{i \in S} 0\right) = 0. \tag{4}$$

So, regardless of the distribution from which we pick F, we have

$$\Pr_F[p(\mathbf{0}) = 0] = 1. \tag{5}$$

Next, suppose $x_1 \vee \ldots \vee x_k = 1$. We have $p(x) = 1$ if and only if the product term is zero. The product term is zero if and only if the sum in some factor is 1. And that, in turn, happens if and only if there is some $S \in F$ which includes *exactly one* x_i which is 1. Formally, for any $x \in \{0,1\}^k$, $x \neq \mathbf{0}$, we want the following to be true with high probability.

$$\exists S \in F.\, (|\{i \in S : x_i = 1\}| = 1) \tag{6}$$

Given that we do not want F to be very large (so that the degree of the polynomial is small), we'll have to pick F very carefully. In order to accomplish this, we turn to the Valiant-Vazirani reduction, which you may recall from an earlier class.

Lemma 8 (Valiant-Vazirani) *Let* $A \subseteq \{1, \ldots, k\}$*, let* a *be such that* $2^a \leq |A| \leq 2^{a+1}$*, and let* H *be a family of pairwise independent hash functions of the form* $h : \{1, \ldots, k\} \to \{0,1\}^{a+2}$*. Then if we pick* h *at random from* H*, there is probability at least* $1/8$ *that there is a unique element* $i \in A$ *such that* $h(i) = \mathbf{0}$*. Precisely,*

$$\Pr_{h \sim H}[|\{i \in A : h(i) = \mathbf{0}\}| = 1] \geq \frac{1}{8} \tag{7}$$

With this as a guide, we will define our collection F in terms of pairwise independent hash functions. Let $t > 0$ be a value that we will set later in terms of the approximation parameter ϵ. Then we let $F = \{S_{a,j}\}_{a\in\{0,\ldots,\log k\}, j\in\{1,\ldots,t\}}$ where the sets $S_{a,j}$ are defined as follows.

- For $a \in \{0, \ldots, \log k\}$:
    - For $j \in \{1, \ldots, t\}$:
        - Pick random pairwise independent hash function $h_{a,j} : \{1, \ldots, k\} \to \{0, 1\}^{a+2}$
        - Define $S_{a,j} = h^{-1}(\mathbf{0})$. That is, $S_{a,j} = \{i : h(i) = \mathbf{0}\}$.

Now consider any $x \neq \mathbf{0}$ which we are feeding to our OR-polynomial p. Let A be the set of bits of x which are 1, i.e., $A = \{i : x_i = 1\}$, and let a be such that $2^a \leq |A| \leq 2^{a+1}$. Then we have $a \in \{0, \ldots, \log k\}$, so F includes t sets $S_{a,1}, \ldots, S_{a,t}$. Consider any one such $S_{a,j}$. By Valiant-Vazirani, we have

$$\Pr_{h_{a,j}\sim H}\left[|\{i \in A : h_{a,j}(i) = \mathbf{0}\}| = 1\right] \geq \frac{1}{8} \tag{8}$$

which implies that

$$\Pr_{h_{a,j}\sim H}\left[|\{i \in A : i \in S_{a,j}\}| = 1\right] \geq \frac{1}{8} \tag{9}$$

so the probability that there is some j for which $|S_{a,j} \cap A| = 1$ is at least $1 - \left(\frac{7}{8}\right)^t$, which by the reasoning above tells us that

$$\Pr_{p}\left[p(x) = x_1 \vee \ldots \vee x_k\right] \geq 1 - \left(\frac{7}{8}\right)^t. \tag{10}$$

Now, to get a success probability of $1-\epsilon$ as required by the lemma, we just pick $t = O(\log \frac{1}{\epsilon})$. The degree of p will then be $|F| = t(\log k) = O((\log \frac{1}{\epsilon})(\log k))$, which satisfies the degree requirement of the lemma. □

Note that given this lemma, we can also approximate AND with an exponentially low degree polynomial. Suppose we have some G which approximates OR within ϵ as above. Then we can construct G' which approximates AND by drawing g from G and returning g' such that

$$g'(x_1, \ldots, x_k) = 1 - g(1 - x_1, \ldots, 1 - x_k). \tag{11}$$

Any such g' has the same degree as g. Also, for a particular $x \in \{0, 1\}^k$, g' clearly computes AND if g computes OR, which happens with probability at least $1 - \epsilon$ over our choice of g.

3.2 Proof of Lemma 4

Given a circuit C of size S and depth d, for every gate we pick independently an approximating function g_i with parameter $\epsilon = \frac{1}{4S}$, and replace the gate by g_i. Then, for a given input, the probability that the new function so defined computes $C(x)$ correctly is at least the probability that the results of all the gates are correctly computed, which is at least $\frac{3}{4}$. In particular, there is a function among those generated this way that agrees with $C()$ on at least a $3/4$ fraction of inputs. Each g_i has degree at most $O((\log S)^2$, because the fan-in of each gate is at most S, and the degree of the function defined in the construction is at most $O((\log S)^{2d})$.

4 Proof of Lemma 5

Let $g : \{0,1\}^n \to \mathbb{R}$ be a function of degree at most t that agrees with PARITY on at least a 3/4 fraction of inputs. Let $G : \{-1,1\}^n \to \mathbb{R}$ be defined as

$$G(x) := 1 - 2g\left(\frac{1}{2} - \frac{1}{2}x_1, \cdots, \frac{1}{2} - \frac{1}{2}x_n\right) \tag{12}$$

Note that:

- G is still of degree at most t,
- G agrees with the function $\Pi(x_1, \ldots, x_n) = x_1 \cdot x_2 \cdots x_n$ on at least a 3/4 fraction of $\{-1,1\}^n$.

Define A to be the set of $x \in \{-1,1\}^n$ such that $G(x) = \Pi(x)$.

$$A := \left\{ x : G(x) = \prod_{i=1}^{n} x_i \right\} . \tag{13}$$

Then $|A| \geq \frac{3}{4}2^n$, by our initial assumption. Now consider the set F of all functions $f : A \to \mathbb{R}$. These form a vector space of dimension $|A|$ over the reals. We know that any function f in this set can be written as

$$f(x) = \sum_{\alpha} \hat{f}_\alpha \prod_{i \in \alpha} x_i \tag{14}$$

Over A, $G(x) = \prod_{i=1}^{n} x_i$, and so for $x \in A$,

$$\prod_{i \in \alpha} x_i = G(x) \prod_{i \notin \alpha} x_i \tag{15}$$

By our initial assumption, $G(x)$ is a polynomial of degree at most t. Therefore, for every α, such that $|\alpha| \geq \frac{n}{2}$, we can replace $\prod_{i \in \alpha} x_i$ by a polynomial of degree less than or equal to $t + \frac{n}{2}$. Every such function f which belong to F can be written as a polynomial of degree at most $t + \frac{n}{2}$. Hence the set $\left\{\prod_{i \in \alpha} x_i\right\}_{|\alpha| \leq t + \frac{n}{2}}$ forms a basis for the set S. As there must be at least $|A|$ such monomials, this means that

$$\sum_{k=0}^{t+\frac{n}{2}} \binom{n}{k} \geq \frac{3}{4} \cdot 2^n \tag{16}$$

and, in particular,

$$\sum_{k=\frac{n}{2}}^{t+\frac{n}{2}} \binom{n}{k} \geq \frac{1}{4} \cdot 2^n \tag{17}$$

We know from Stirling's approximation that every binomial coefficient $\binom{n}{k}$ is at most $O(2^n/\sqrt{n})$, so we get

$$O\left(\frac{t}{\sqrt{n}} \cdot 2^n\right) \geq \frac{1}{4} \cdot 2^n \tag{18}$$

And so $t = \Omega(\sqrt{n})$.