18.405J/6.841J Advanced Complexity Theory	March 6, 2000
Lecture 10	
Lecturer: Dan Spielman	*Scribe: Mihai Bădoiu*

This lecture is the first one on circuit lower bounds. It introduces AC_0 and gives a proof that parity is not in this class.

1 Circuit lower bounds

Definition 1 *AC_0 is the class of languages accepted by circuits of AND, OR and NOT gates with constant depth (NOT gates are excluded when we measure the depth of the circuit), unbounded fanin and polynomial number of inputs.*

Theorem 2 *(Furst-Sipser-Saxe and Ajtai) Parity does not belong to AC_0, where parity is the language of binary strings containing and odd number of 1s. (parity($x_1, \dots, x_n$) $= \sum x_i$ (mod 2))*

This proof is due to Smolensky-Bieigel-Reingold-Spielman. We will actually prove something stronger: a parity circuit of constant depth has at least exponential size.

1.1 Special Case: depth-2 circuit

Claim 3 *A depth-2 circuit that computes parity on n inputs must have at least 2^{n-1} gates.*

Proof Suppose the circuit is an OR of ANDs of inputs or their negations. All functions can be computed by circuits of this form. Every AND gate must take every input of the circuit in either inverted or non-inverted form. Suppose that some AND gate doesn't take input x_i at all. Set up the circuit so that the output of the AND gate is 1. Now, switch the value of x_i. The output of the entire circuit is still 1, and the real value of parity has changed. Therefore the circuit cannot decide parity.
The circuit must output 1 for 2^{n-1} combination of circuit inputs. The curcuit outputs 1 when at least 1 AND gate outputs 1. Since every AND gate takes every circuit input, there's at most 1 combination that makes the output of the gate 1. Therefore, there are at least 2^{n-1} AND gates. ∎

1.2 Main Theorem

Theorem 4 *A depth-d circuit that computes parity on n inputs must have $2^{n^{\Omega(1/d)}}$ gates.*

Proof The proof idea is to write a "low-degree" polynomial that is equivalent an OR($x_1, \dots, x_n$) and to use it to write a polynomial for an entire circuit C. The idea is that if C has small depth and not too many gates, there is a "low-degree" polynomial which computes C. If there is a constant depth circuit that computes parity, there's also a "low-degree" polynomial that computes parity. We will reach a contradiction by showing that there cannot exist a "low-degree" polynomial that computes parity.

Lemma 5 *An OR (or AND) of m inputs can be computed by a probabilistic polynomial of degree $k(\log m)^{O(1)}$ with error probability at most 2^{-k}.*

Proof

Recall the Valiant-Vazirani theorem from lecture 8: there exists a random polynomial time procedure that takes a circuit C and returns $n+1$ circuits $C_1, ..., C_{n+1}$ such that if C is satistfiable,

$$\Pr[\text{there exists } i \text{ s.t. } C_i \text{ has exactly one satisfying assignment}] > 1/8,$$

and otherwise no C_i has a satisfying assignment.

Using Valiant-Vasirani, we construct $O(\log n)$ degre polynomials that approximate OR of n inputs. Notion of approximation: construct $P(\bar{\Gamma}, \bar{X})$ such that

$\forall x, \Pr_{\bar{\Gamma}}[P(\bar{\Gamma}, \bar{x}) = OR(\bar{x})] > 1 - \text{small}$

VV: $\forall S \in \{0,1\}^n$ if we choose $h_1, \ldots, h_{n+1}$, $h_i \in H_{i,n}$

$$S \neq 0 \implies \Pr[\exists i | h_i^{-1}(\bar{0}) \cap S \| = 1] \geq 1/8$$

$H_{i,n} : x \to Mx + b$, $M \in GF(2)^{i \times n}$, $b \in GF(2)^i$

There exists a poly size, $O(\log n)$ depth circuit C such that $C(h, x) = 1$ if $h(x) = \bar{0}$ and 0 otherwise. We interpret h as a hash function for n from $H_{i,n}$.

There exists a log space TM that on input 1^n outputs $C_1, \ldots, C_{n+1}$ such that for all $S \in \{0,1\}^n$

$$S \neq 0 \implies \Pr_r[\exists i, |\{x \in S | C_i(r, x) = 1\}| = 1] \geq 1/8$$

where $|r| = O(n^3)$

Note that we can boost the error probability to $(7/8)^k$ if we output $C_1, \ldots, C_{k(n+1)}$ for $|r| = O(kn^3)$.

To turn a circuit into a polynomial, we use the following transformations:

$$x \bigwedge y \to xy$$
$$\bar{x} \to (1 - x)$$

For circuit C we construct a polynomial P of poly(n) degree that simulates C.

Say we have variables $a_1, \ldots, a_{2^n} \in \{0,1\}$ indexed by $\{0,1\}^n$. We want to approximate OR$(a_1, \ldots, a_{2^n})$. Consider $P = \sum_{x \in \{0,1\}^n} P_i(\bar{r}, x) a_x$ a poly in $a_1, \ldots, a_{2^n}$ and $\bar{r}$. r has size $O(kn^3)$. P has degree $(kn^3)^c$

$$\text{if } \sum a_x > 0 \text{ then } \Pr_r \left[\exists i \text{ such that } \sum_{x \in \{0,1\}^n} P_i(\bar{r}, x) a_x = 1 \right] > 1 - (7/8)^k$$

let $V(\bar{r}, a_1, \ldots, a_{2^n}) = 1 - \prod_{i=1}^{n^k}(1 - \sum_{x \in \{0,1\}^n} P_i(\bar{r}, \bar{x}) a_x)$. if $\sum a_x = 0$ then $V(\bar{r}, a_1, \ldots, a_{2^n}) = 0$ else $Pr_r[V(\bar{r}, a_1, \ldots, a_{2^n}) = 1] > 1 - (7/8)^{n^k}$. V has degree n^k in a_x and $n^{(c+1)(k+1)}$ in $\bar{r}$. By scaling down $(2^n \to n)$, we get that we can approximate OR by a polynomial of degree $O(t (\log n)^{O(1)})$ with error $(7/8)^t$ for $t = \log^k n$. ■

Lemma 6 *For any AC_0 circuit of depth d with G gates there is a probabilistic polynomial of degree $\epsilon = (\log G)^{O(d)}$ that computes the same function with error probability at most $1/4$.*

Proof We use our transformations to turn the circuit into a polynomial. To bound the degree of the polynomial, since the depth of the circuit is d and any gate can have at most G inputs, the degree of the polynomial is $O((\log G)^{O(d)})$. To bound the error probability to at most $1/4$ we set k such that the error probability for each gate is low, such that $G 2^{-k} < 1/4$. This makes k about $\log G$. ■

1.3 Low-Degree Polynomial Gives Contradiction

First note that parity can be easily written as a degree n poly: we can transform 0,1 $\rightarrow$ +1, -1 over the reals, using a linear map:

$$x_i \rightarrow 1 - 2x_i$$

now we can consider the inputs x_i are over +1, -1 instead of 0, 1. $x_i \in \{+1, -1\}$, $x_i^2 = 1$.

$$parity = \prod x_i$$

is a polynomial of degree n that computes parity.

Also, if P is a polynomial that computes parity on 0, 1, $P_2 = 1 - 2P(\frac{1+x_1}{2}, \ldots, \frac{1+x_n}{2})$ is a polynomial that computes parity on +1, -1 and has the same degree as P.

Let P_C be the polynomial determined in the previous section that computes parity for a circuit C. $\exists r$ such that $\exists S \subseteq \{-1, +1\}^n, \forall x \in S, \prod x_i = P_c(r, x)$ and $|S| \geq (3/4)2^n$. Now consider the space of functions $S \rightarrow R$. This space has dimension greater than $(3/4)2^n$ since a basis for this space can be given by a set of functions in which one element of S is mapped to 1, and the others are mapped to 0.

For any $f : S \rightarrow R$, the exists a polynomial q of degree at most n such that $f(x) = q(x) \forall x \in S$. We can obtain q for any f by interpolation. Since $x_i^2 = 1$, we can reduce the degree of q by writing it as a sum of monomials in which each variable has degree at most 1.

Claim 7 *We can assume q has degree at most $n/2 + \epsilon$.*

Proof q is a sum of monomials. If a monomial has degree greater than $n/2$ then we rewrite it as a product of the variables which previously did not appear and its parity. (since parity is the product of all x_i and $x_i^2 = 1$, these 2 values have the product 1 and therefore they are equal). The degree of the new q is at most $n - n/2 + \epsilon = n/2 + \epsilon = n/2 + (\log G)^{O(d)}$. Since we can do this for all the monomials of q will get a polynomial equal to q that has degree at most $n/2 + \epsilon$. ■

If $\epsilon = \Omega(\sqrt(n))$, then $G = 2^{n^{\Omega(1/d)}}$. So, $\epsilon = o(\sqrt(n))$. As a basis we can use only monomials containing at most $\epsilon + n/2$ variables. The number of such monomials is:

$$\sum_{i=1}^{n/2+\epsilon} \binom{n}{i} < \frac{3}{4} 2^n \leq |S|$$

This gives us the desired contradiction.

■