18.405J/6.841J Advanced Complexity Theory	March 18, 1997
Lecture 12	
Lecturer: Daniel A. Spielman	*Scribe: Eric Lehman*

This lecture presents a first proof that parity is not in AC^0. Recall that AC^0 is the class of languages decided by a logspace-uniform family of polynomial-size, constant-depth circuits. Each circuit is composed of inverters and AND and OR gates with an unbounded number of inputs. (Inverters are excluded when measuring the depth or size of a circuit.) Also recall that parity is the language of binary strings containing an odd number of ones. Parity can also be regarded as boolean function which is 1 when an odd number of the inputs are 1.

The result is originally due to Furst-Saxe-Sipser and Ajtai. We will actually prove something stronger: that a parity circuit of constant depth must have exponential size. The proof given here is similar to that given in Beigel-Reingold-Spielman, which in turn uses an idea of Smolensky.

1 Special Case: Two-Level Circuits

First, we prove a special case: a parity circuit of depth two must have exponential size.

Theorem 1 *A depth-two parity circuit on n inputs $x_1 \ldots n_n$ must have at least $2^{n-1}+1$ gates.*

Proof Suppose that the circuit is an OR of ANDs. (A similar argument applies for an AND or ORs. All depth-two circuits can be reduced to one of these two by easy transformations.)

First, we show that every AND gate must take every input of the circuit in either inverted or non-inverted form. Suppose that some AND gate is not supplied with input x_i. Set the other circuit inputs so that the output of the AND is 1. Now regardless of the value of x_i, the output of the entire circuit is 1. But the output of a parity circuit should change value whenever any single input changes, and so the circuit does not compute parity.

Next, we show that there must be at least 2^{n-1} AND gates. These, together with the single OR gate, give the lower bound. Since each AND gate takes every circuit input, there is a unique combination of circuit inputs which makes the AND gate output 1. The entire circuit outputs 1 only when some AND gates outputs 1. However, the definition of parity requires that the entire circuit output 1 for 2^{n-1} combinations of circuit inputs. Therefore, there must be at least 2^{n-1} AND gates. ■

2 Main Theorem and Proof Idea

Now we state the main theorem of this lecture and give a rough idea of the proof. The circuits considered in the theorem consist of AND and OR gates with unbounded fanin. Inverters are permitted only at the circuit inputs.

Theorem 2 *A depth-d parity circuit on inputs $x_1 \ldots x_n$ must have $2^{n^{\Omega(1/d)}}$ gates.*

The proof has two steps. The first step is to show that a "small", constant depth circuit which computes parity implies the existence of a "low-degree" polynomial which (more or less) computes parity. The second step is to show that this low-degree polynomial which computes parity implies that there exist low-degree polynomials for a very large space of functions. We obtain a contradiction by showing that this space of functions has higher dimension than the space of low-degree polynomials. This implies that the "low-degree" polynomial, and therefore the "small" parity circuit, does not exist. These two steps are detailed in the following two sections.

3 Small Circuit Implies a Low-Degree Polynomial

Recall the following result due to Valiant and Vazirani, which says that an AND or OR gate can be approximated by a random polynomial.

Lemma 3 *There exists a constant c so that the AND (or OR) of m inputs can be computed by a probabilistic polynomial of degree $k(\log m)^c$ with error probability at most 2^{-k}.*

Corollary 4 *For any AND/OR circuit of depth d with G gates and with inverters only at the inputs, there exists a probabilistic polynomial of degree $\epsilon = (\log G)^{(c+1)d}$ that computes the same function with error probability at most $1/4$.*

Proof For each AND and OR gate, construct the probabilistic polynomial guaranteed to exist by the lemma. If a gate uses an input x_i in an inverted sense, then we can replace each instance of x_i by $1 - x_i$ in the polynomial corresponding to the gate. By composing these, we obtain a probabilistic polynomial for the entire circuit. What remains is to check that the polynomial obtained in this way has the claimed error bound and degree.

For the error bound, we make the polynomial for each gate have such low error probability that the polynomial for the entire circuit has error probability at most $1/4$. To this end, we choose the confidence paramenter k, defined in the lemma, so that $G2^{-k} < 1/4$. This implies that k is about $\log G$.

For the degree, take G as an upper bound on m, the number of inputs to any gate in the circuit. Now the degree of the polynomial for a single gate is $(\log G)^c \log G = (\log G)^{(c+1)}$. Since the circuit has depth d, the degree of the polynomial for the entire circuit is $(\log G)^{(c+1)d} = \epsilon$. ■

As a special case of the corollary, a circuit with G gates and depth d that computes parity would imply a degree ϵ probabilstic polynomial which computes parity with error probability at most $1/4$. This implies that there is some setting of the probabilistic variables so that the polynomial computes parity correctly for at least $3/4$ of the possible input combinations. But fixing the probabilistic variables of a probabilistic polynomial produces an ordinary, non-probabilistic polynomial. Therefore, a parity circuit with G gates and depth d implies an ordinary polynomial $p'(x_1 \ldots x_n)$ of degree ϵ that computes parity correctly for at least $3/4$ of the settings of $x_1 \ldots x_n$.

4 Low-Degree Polynomial Gives Contradiction

The preceding section showed that a small circuit for parity implied a low-degree polynomial p' which (more or less) computes parity. This section shows that the existence of such a polynomial implies a contradiction. This will imply that there is no small circuit for parity.

We can define a variant of the parity function over $+1, -1$ instead of the usual $0, 1$. Whereas the usual parity function is 1 when an odd number of inputs are 1, the new parity function is -1 when an odd number of the inputs are -1. The simplest way to compute this new parity function is by multiplication. That is, parity on $x_1, \ldots, x_n$ is simply $\Pi_{i=1}^n x_i$. However, this polynomial has degree n.

We can obtain a lower degree polynomial p for this new parity function from the polynomial p' constructed in the preceeding section. Define $p(x_1 \ldots x_n) = 1 - 2p'((1 - x_1)/2 \ldots (1 - x_n)/2)$. This polynomial p has the same degree as p', which is $\epsilon = (\log G)^{(c+1)d}$. However, p has the same failing as p'; namely, p only computes parity correctly for at least $3/4$ of all inputs. Hereafter, we will use S denote the subset of input combinations for which p computes this new parity function correctly. Looked at another way, S is a subset of the vertices of an n-dimensional hypercube centered on the origin.

Consider the space of all real-valued functions defined on S. A basis for this space is given by the set of functions in which one element of S is mapped to 1, and the other elements of S are mapped to 0. Therefore, the dimension of this space of functions is $|S| \geq \frac{3}{4}2^n$.

We can obtain a polynomial q for any such function by interpolation. The degree of this polynomial q can be reduced using two tricks. First, we can reduce q to a sum of monomials in which each variable x_i has degree at most 1. Since x_i is either +1 or -1, we have $x_i^2 = 1$. As a result, any higher power of x_i is equal to either x_i or 1. This reduces the degree of q to at most n.

Second, rewrite the polynomial as follows. If a monomial contains $n/2$ or fewer variables, then leave it unchanged. If a monomial contains more than $n/2$ variables, then rewrite it as the product of $p(x_1 \ldots x_n)$ and all the variables which previously did not appear. This new monomial is equal to the original because p is equal to the product of all the variables and we "cancel out" unwanted variables by multiplication. This again uses the fact that $x_i^2 = 1$. Futhermore, the degree of the resulting monomial is at most the degree of p plus the number of variables which did not previously appear. This reduces the degree of q to at most $\epsilon + n/2 = (\log G)^{(c+1)d} + n/2$.

What is the dimension of this space of polynomials of degree at most $\epsilon + n/2$? As a basis, we can use monomials containing at most $\epsilon + n/2$ variables. If $\epsilon = o(\sqrt{n})$, then number of such monomials is:

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

This gives the desired contraction. Evidently, the original circuit was not so small and so the polynomial p is not so low degree. More precisely, ϵ is not $o(\sqrt{(n)})$. But if ϵ is $\Omega(\sqrt{n})$, then G is $2^{n^{\Omega(1/d)}}$.