1. Parity does not have low degree approximations

In the last lecture we showed the following theorem.

**Theorem 1.** If $C$ is a size $S$ depth $d$ circuit (with Mod$_3$ gates) then there is a polynomial $p$ of degree $(2k)^d$ such that $\Pr[p(x) = C(x)] \geq 1 - \frac{s}{3^k}$.

We will set the degree to $\sqrt{n}$ so $(2k)^d = n^{1/2}$. Then $k = \frac{n^{1/2}d}{2}$, so $1 - \frac{s}{3^k} \approx \frac{s}{3^{n^{1/2}d}}$.

This is not true about parity: there is no low-degree polynomial over GF(3) such that the polynomial is a good approximation of parity.

**Theorem 2.** Let $p$ be a $\sqrt{n}$ degree polynomial. Let $S = \{x \in \Sigma^n | p(x) = \text{parity}(x)\}$. Then $|S| \leq 0.92^n$.

**Corollary 1.** Parity does not have polynomial-size $AC_0$ circuits with Mod$_3$ gates.

**Proof of Theorem 2.** One way to express parity as a polynomial over GF(3) is: $\text{parity}(x_1, \ldots, x_n) = \prod_{i=1}^n (1 + x_i) - 1$. (If the number of 1’s is odd, parity is 1; if the number of 1’s is even, parity is zero.) Look at $A = \{f | f : S \rightarrow GF(3) \text{ is a function}, |A| = 3^{|S|}\}$. We will count the size of $A$ in a different way now. Consider any function $f$ from $S$ to GF(3). We can write any function as a multilinear polynomial, as we saw last time. So $f = \sum \alpha_k m_k$, where the $\alpha_k$ are constants and the $m_k$ are monomials. So $m_k = \prod_{i \in T} x_i$ for some $T \subseteq \{1, \ldots, n\}$. But we don’t know anything about the degree of the $m_k$.

If we let $y_i = 1 + x_i$, then the parity polynomial looks like a product of monomials, i.e., $\prod_{i=2}^n y_i - 1$. Our goal is to rewrite each $m_k$ as a monomial of small degree. We will take advantage of the fact that the parity polynomial can, by hypothesis, be written as a $\sqrt{n}$-degree polynomial.

Let $m = \prod_{i \in T} x_i$

$$= \prod_{i \in T} (1 + x_i - 1)$$

$$= \alpha_1 \prod_{i \in T_1} 1 + \alpha_2 \prod_{i \in T_2} (1 + x_i) + \alpha_3 \prod_{i \in T_3} (1 + x_i) + \cdots \text{ where each } T_j \subseteq T, \text{ and } \alpha_i 's \text{ are some constants}$$

We obtain the third line from the second by expanding out the product and sorting it according to variables that have common coefficients. That leaves a summation of products of variables, plus some constant term; we rewrite the constant $\alpha_1$ as $\alpha_1 \prod_{i \in T_1} 1$.

So we can write $f$ as $\sum B_k m'_k$ where $m'_k = \prod_{i \in T} (1 + x_i)$ for some $T \subseteq \{1, \ldots, n\}$.

**Claim 1.** For every $x = x_1 \cdots x_n$ from $\{0, 1\}^n$, and $T \subseteq \{1, \cdots, n\}$,

$$\prod_{i \in T} (1 + x_i) = \prod_{i=1}^n (1 + x_i) \prod_{i \notin T} (1 + x_i).$$
This is because \((1 + x_i)^2 = 1\) (over \(\text{GF}(3)\)) when \(x_i \in \{0, 1\}\), and
\[
\prod_{i=1}^{n}(1 + x_i) \prod_{i \notin T}(1 + x_i) = \prod_{i \notin T}(1 + x_i)^2 \prod_{i \in T}(1 + x_i).
\]

Recall that \(f = \sum B_k m'_k\) where each \(m'_k = \prod_{i \in T}(1 + x_i)\) for some \(T\), and we do not have any control over the degree of \(m'_k\). We would like to rewrite \(f\) as a small degree polynomial. Consider an \(m'_k\), if its degree is at most \(n/2\), we leave it as it is. However, if \(m'_k\) has degree \(n/2 + d, d > 0\), then \(m'_k = \prod_{i \in T}(1 + x_i)\) where \(|T| = n/2 + d\). By the Claim 1, \(m'_k = \prod_{i=1}^{n}(1 + x_i) \prod_{i \notin T}(1 + x_i)\).

Recall that
\[
\text{parity}(x_1, \ldots, x_n) = \prod_{i=1}^{n}(1 + x_i) - 1,
\]
and \(p(x) = \text{parity}(x)\) on \(S\). So
\[
m'_k = (p(x) + 1) \prod_{i \notin T}(1 + x_i)
\]
Since \(x_i^2 = x_i\) when \(x_i \in \{0, 1\}\) and the degree of \(p\) is at most \(\sqrt{n}\), \(m'_k\) can be written as a multilinear polynomial with degree at most \(n/2 + \sqrt{n}\).

To summarize, every function \(f\) in \(A\) can be expressed as a multilinear polynomial \(g\) with degree at most \(n/2 + \sqrt{n}\) and for every \(x \in S\), \(f(x) = g(x)\).

Now, how many monomials can have degree at most \(n/2 + \sqrt{n}\)? The number of monomials with at most that degree is \(\sum_{i=0}^{n/2+\sqrt{n}} \binom{n}{i} < 0.92^n\). (The exact calculation can be obtained by expanding the binomial coefficient, or by using Chernoff bounds.)

So the number of functions from \(S\) to \(\text{GF}(3)\) is less than \(3^{\text{number of monomials}} = 3^{0.92^n}\). The number of functions from \(s\) to \(\text{GF}(3)\) is \(3^{|S|}\). Therefore \(|S| \leq 0.92^n\). \(\square\)

We have to work in a field other than \(\text{GF}(2)\), because in \(\text{GF}(2)\) parity has a low-degree polynomial. It seems strange that by changing the field, we change the degree of the polynomial. This was a special case of the Razborov Smolensky proof of the following theorem.

**Theorem 3.** If \(p\) and \(q\) are two distinct primes, then \(\text{Mod}_p \notin \text{AC}_0[\text{Mod}_q]\).

This is true even if we consider prime powers, e.g., \(\text{AC}_0[\text{Mod}_{p^k}]\). The first number that cannot be expressed as a prime power is 6. We have no idea what is the power of \(\text{ACC}_0[6]\). As far as we know, nondeterministic exponential time is in there.

In terms of circuits, these are essentially the best results we know. We have no lower bounds if we consider other than constant-depth circuits, even if the depth is \(\log \log n\). Our overall goal has been to exhibit an explicit function in \(P\) that requires large circuits. Best known require \(5.5n\)-size circuits.

2. **LOWER BOUND ON CIRCUIT DEPTH OF LANGUAGES IN \(\Sigma_3\)**

Our overall goal has been to exhibit an explicit function in \(P\) that requires large circuits. Best known require \(5.5n\)-size circuits. Let us consider something weaker: a function in a class higher than \(P\) that requires large circuits. We believe that \(\text{NP}\) requires superpolynomial-size circuits, maybe even exponential-size circuits. Can we show that \(\text{NP}\) does not have \(n^2\)-size
circuits? And we don’t know the answer to this thing either – the best is 5n-size circuits, because the function in P is also in NP.

What is the smallest class we can actually show does not have \( n^2 \)-size circuits?

**Theorem 4.** \( \Sigma_2 \) does not have \( n^2 \)-size circuits.

Recall: \( L \in \Sigma_2 \) means that there is a polynomial-time-computable relation \( R(\cdot, \cdot, \cdot) \) and a polynomial \( p \) such that:

1. if \( x \in L \) then \( \exists y \in \{0, 1\}^{\leq p(|x|)} \forall z \in \{0, 1\}^{\leq p(|x|)} \in R(x, y, z) = 1 \),
2. if \( x \not\in L \) then \( \forall y \in \{0, 1\}^{\leq p(|x|)} \exists z \in \{0, 1\}^{\leq p(|x|)} \in R(x, y, z) = 0 \).

A language \( L \) is in \( \Pi_2 \) if \( \exists \in \Sigma_2 \).

A language \( L \) is in \( \Sigma_3 \) if there is a polynomial \( p \), and a polynomial-time-computable relation \( R \) such that

1. \( x \in L \) then \( \exists y \forall z \exists u R(x, y, z, u) = 1 \)
2. \( x \not\in L \) then \( \forall y \exists z \forall u R(x, y, z, u) = 0 \)

Here the strings \( y, z, u \) have length bounded by \( p(|x|) \).

We will first show that \( \Sigma_3 \) does not have \( n^2 \)-size circuits.

**Theorem 5.** \( \Sigma_3 \) does not have \( n^2 \)-size circuits.

*Proof.* The basic idea once again is counting. Fix a length \( n \). Look at the number of circuits of size \( \leq n^2 \). We can write a circuit of size \( s \) as a string of size \( s \log s \). Thus there are about \( 2^{s \log s} \) circuits of size \( s \). So there are at most \( 2^{n^4} \)-many circuits of size \( \leq n^2 \). Fix a length \( n \). Let \( x_i \) be the \( i \)th string at length \( n \). Take a circuit \( C \) of size \( \leq n^2 \). Look at the sequence \( C(x_1)C(x_2)\cdots C(x_{n^5}) \). Call this sequence \( \chi_C \). \( T = \{ \chi_C \mid C \) has size \( \leq n^2 \} \). What is the maximum size of \( T \)?

\[ |T| \leq 2^{n^4} \]

We would like to come up with a \( \Sigma_3 \) language \( L \) that cannot be computed with \( n^2 \)-size circuits. If we can define \( L \) such that \( L(x_1)L(x_2)\cdots L(x_{n^5}) \notin T \), then we are done. So we need a string of length \( n^5 \) that does not belong to \( T \). Since \( 2^{n^4} < 2^{n^5} \), such strings do exist. Look at the smallest such string. Call it \( Y \). We want \( Y \) to be the characteristic sequence of our target language \( L \).

What is the complexity of \( Y \)? We can describe \( Y \) as “Smallest string of length \( n^5 \) that is not in \( T \).” In terms of quantifiers, \( Y \) is a strings that the following property: For every \( C \), \( \chi_C \neq Y \), \( \forall Z \neq Y \) \( YZ \in T \). Based on this we now define our language \( L \).

Definition of \( L \) (at length \( n \)).

1. \( i > n^5 \Rightarrow x_i \notin L \)
2. \( x_i \in L \iff \exists y \) such that \( |y| = n^5 \) and \( \forall C \) of size \( \leq n^2 \) \( \chi_C \neq y \) and \( \forall z < y \exists C \) of size \( \leq n^2 \) st \( z = \chi_C \) and the \( i \)th bit of \( y \) is \( 1 \)

This ensures that \( L(x_1)L(x_2)\cdots L(x_{n^5}) \) is the smallest string that is not in \( T \). Observe that we can collapse consecutive universal quantifiers into a single universal quantifier. Also observe that the conditions such as \( \chi_C = y \) and \( \chi_C = z \) can be decided in polynomial time. Thus \( L \in \Sigma_3 \).

\( \square \)

So \( \Sigma_3 \) does not have \( n^2 \)-size circuits. The above argument can be easily modified to show that for any fixed \( k \), there is a language in \( \Sigma_3 \) that does not have \( n^k \)-size circuits. This is different from saying that there is an \( L \in \Sigma_3 \) such that, for all \( k \), \( L \) does not have \( n^k \)-size circuits. The second statement is much stronger, and is not known.