1 Small depth classes

The class NC (Nick’s class) corresponds to circuits with small depth and fan-in two.

**Definition 1.1 (NC).** The class \( NC^i \) is the class of circuits with AND/OR/NOT gates of fan-in 2 and depth \( O(\log(n)^i) \). We define \( NC = \cup_{i \geq 1} NC^i \).

**Definition 1.2 (AC).** The class \( AC^i \) is the class of circuits with AND/OR/NOT gates of unbounded fan-in gates and depth \( O(\log(n)^i) \).

Clearly \( NC^i \subset AC^i \subset NC^{i+1} \). These classes correspond to fast parallel algorithms. We suspect that \( NC \neq P \) but can’t even separate \( NC \) from \( PH \).

The class \( NC^0 \) corresponds to functions which depend just on a constant number of input bits. The class \( AC^0 \) is more interesting and can compute nontrivial functions, e.g., approximate majority. We don’t know how to show \( NC^1 \neq P \). However, we can prove lower bounds for \( NC^0 \), \( AC^0 \).

**Theorem 1.3.** The PARITY function cannot be computed in \( NC^0 \).

**Proof.** An \( NC^0 \) circuit can only read \( O(1) \) inputs.

**Theorem 1.4 (Hastad).** The PARITY function cannot be computed in \( AC^0 \) (not even approximately).

We first prove a baby version of this.

**Theorem 1.5.** A depth 2 circuit computing PARITY must have size \( \Omega(2^n) \).

**Proof.** A depth 2 circuit is essentially a DNF or CNF. Let w.l.o.g DNF. Let \( \varphi = D_1 \lor D_2 \ldots \lor D_m \) be a DNF computing the PARITY function. Let us first argue that all the terms must have \( n \) variables. Assume some term \( D \) has less than \( n \) variables, and let \( x_j \) be a variable not participating in \( D \). We can find an assignment to the variables \( \{x_i : i \neq j\} \) which make \( D \) true. Let us now set the variables outside \( D \) so that the parity will be 1 (false). Then we get that \( D(x) \neq PARITY(x) \) on this input. Contradiction. So, we got that all terms have exactly \( n \) variables, hence they compute 1 on a single input and 0 on the rest. PARITY has \( 2^n - 1 \) inputs where it evaluates to 1, hence if \( \phi = PARITY \) it must have at least \( 2^{n-1} \) terms.
2 Polynomials

We will view boolean functions as \( f : \{0,1\}^n \to \{0,1\} \). A polynomial is an expression of the form

\[
p(x_1, \ldots, x_n) = \sum_{S \subseteq [n]} p_S \prod_{i \in S} x_i.
\]

**Claim 2.1.** Any boolean function can be computed by a unique polynomial.

**Proof.** Let \( FUNC = \{ f : \{0,1\}^n \to \mathbb{R} \} \) denote the vector space over \( \mathbb{R} \) of functions, of dimension \( \text{dim}(FUNC) = 2^n \). Clearly every polynomial defines a function, and they are close under addition, hence the functions given by polynomials also form a vector space. Let \( POLY \) denote the vector space of functions given by polynomials. We need to show \( POLY = FUNC \). We will do so by showing that \( \text{dim}(POLY) = 2^n \). To do that, we need to show that if \( p_1, p_2 \) are polynomials with different monomials then they define different functions (when evaluated on \( \{0,1\}^n \)). Setting \( p = p_1 - p_2 \) we need to show that if \( p \) is a nonzero polynomial then there exists \( x \in \{0,1\}^n \) for which \( p(x) \neq 0 \). Let \( S \) be minimal such that \( p_S \neq 0 \), and let \( x \in \{0,1\}^n \) be such that \( x_i = 1 \) iff \( i \in S \). Then

\[
p(x) = \sum_{S \subseteq [n]} p_S \prod_{i \in S} x_i = p_S \neq 0.
\]

Let \( POLY_k \) denote the family of all polynomials of degree at most \( k \);

\[
POLY_k = \{ p(x) = \sum_{|S| \leq k} p_S \prod_{i \in S} x_i \}.
\]

Sometimes it will be convenient to represent bits as \( \{-1,1\} \). Note that we can always do this change by replacing \( x_i \in \{0,1\} \) with \( 1 - 2x_i \in \{-1,1\} \). Note also this doesn’t change the degree of a polynomial.

3 \( \text{AC}^0 \) circuits can be approximated by low-degree polynomials

**Theorem 3.1.** Let \( C \) be an \( \text{AC}^0 \) circuit with \( n \) inputs, depth \( d \) and \( s \geq n \) gates. Then for any \( \varepsilon > 0 \) there exists a polynomial \( p(x_1, \ldots, x_n) \) of degree \( O(\log^2 d(s)) \) such that

\[
|\{ x \in \{0,1\}^n : p(x) = C(x) \}| \geq 0.99 \cdot 2^n.
\]

We first prove the following lemma handling a single \( \text{AND} \) gate: \( \text{AND}(x_1, \ldots, x_n) = 1 \) if \( x_1 = \ldots = x_n = 1 \), and otherwise \( \text{AND}(x_1, \ldots, x_n) = 0 \).
Lemma 3.2. Let $n \in \mathbb{N}, \varepsilon > 0$. There exists a distribution $D$ over polynomials $p(x_1, \ldots, x_n)$ of degree $k = O(\log(n) \log(1/\varepsilon))$ such that for any $x \in \{0, 1\}^n$,

$$\Pr_{p \sim D}[p(x) = \text{AND}(x)] \geq 1 - \varepsilon.$$ 

Proof. Let $1 \leq t \leq \log(n)$ be chosen uniformly. Let $A \subseteq [n]$ be a random set of size $2^t$. Define the polynomial $p_A(x_1, \ldots, x_n) = \sum_{i \in A} x_i - |A| + 1$.

Clearly if $x_1 = \ldots = x_n = 1$ then $p_A(x) = 1$ always. We will show that if $x \in \{0, 1\}^n \setminus 1^n$ then

$$\Pr_A[p_A(x) = 0] \geq \Omega(1/\log n).$$

So, if we choose $A_1, \ldots, A_k$ randomly with $k = O((\log n) \log(1/\varepsilon))$ and define $p(x) = p_{A_1}(x) \ldots p_{A_k}(x)$ then

1. $p(1^n) = 1$ with probability one.

2. If $x \in \{0, 1\}^n \setminus 1^n$ then $\Pr[p_A(x) = 0] \geq 1 - \varepsilon$ since

$$\Pr[p_A(x) \neq 0] = \prod_{i=1}^k \Pr[p_{A_i}(x) \neq 0] = (1 - O(1/\log n))^{\log n \log(1/\varepsilon)} \leq \varepsilon.$$

To conclude, we need to show that $\Pr_A[p_A(x) = 0] \geq \Omega(1/\log n)$. Fix $x \in \{0, 1\}^n \setminus 1^n$. Assume the number of zeros in $x$ is between $2^a$ and $2^{a+1}$ for some $0 \leq a \leq \log(n)$. Let us condition on the case that $t = \log(n) - a$. In such a case, $|A| = n/2^a$ and the average number of zeros in $A$ is $O(1)$. We can show (but won’t here) that with constant probability, there is exactly one zero in $A$. In such a case however $p_A(x) = 0$.

A similar lemma holds for OR gates. NOT gates clearly can be computed by the polynomial $p(x) = 1 - x$.

Proof of Theorem. Fix an input $x$. Let $v_1, \ldots, v_s$ denote the values of the node $C$ when evaluated on input $x$. Assume we order the nodes so that $v_1(x) = x_1, \ldots, v_n(x) = x_n$ and 

$$v_i(x) = g(v_1(x), \ldots, v_{i-1}(x))$$

where $g \in \{\text{AND, OR, NOT}\}$. By the lemma with error $\varepsilon = 1/100s$ there exists a distribution $D_i$ over polynomials $p_i$ of degree $O(\log^2(s))$ for each node such that

$$\Pr_{p_i \sim D_i}[v_i(x) \neq p_i(v_1(x), \ldots, v_{i-1}(x))] \geq 1/100s.$$

By the union bound,

$$\Pr_{p_1 \sim D_1, \ldots, p_s \sim D_s} \left[ \forall i \in [s], v_i(x) = p_i(v_1(x), \ldots, v_{i-1}(x)) \right] \geq 0.99.$$
Hence, we can apply the polynomials recursively. Nodes at the bottom (inputs) have polynomials of degree 1; nodes one level up have polynomials of degree $\log^2(s)$; nodes one level up have polynomials of degree $\log^4(s)$; and so on. Hence, we get that there is a distribution over polynomials $p(x)$ of degree $O(\log^d(s/\varepsilon))$ so that, for any input $x \in \{0, 1\}^n$,

$$\Pr_{p \sim D}[C(x) = p(x)] \geq 0.99.$$ 

Let now $x \in \{0, 1\}^n$ be uniformly chosen. Then also

$$\Pr_{p \sim D, x \in \{0, 1\}^n}[C(x) = p(x)] \geq 0.99.$$ 

But this means there must be a value for $p \in \text{POLY}_k$ for $k = O(\log^d(s))$ such that

$$\Pr_{x \in \{0, 1\}^n}[C(x) = p(x)] \geq 0.99.$$ 

4 PARITY cannot be approximated by low-degree polynomials

**Theorem 4.1.** Let $p(x_1, \ldots, x_n)$ be a polynomial of degree $k$. Then

$$|\{x \in \{0, 1\}^n : p(x) = \text{PARITY}(x)\}| \leq (1/2 + O(k/\sqrt{n}))2^n.$$ 

**Corollary 4.2.** If an $\text{AC}^0$ circuit with $n$ inputs, depth $d$ and size $s$ computes the PARITY function then $s \geq \exp(n^{1/4d})$.

**Proof.** There exists a polynomial $p(x)$ of degree $k = O(\log(s))$ so that $|\{x \in \{0, 1\}^n : C(x) = p(x)\}| \geq 0.99 \cdot 2^n$. However, this requires $k \geq \Omega(\sqrt{n})$, hence $s \geq \exp(n^{1/4d})$. 

**Proof of Theorem.** It will be convenient to view $p$ as a polynomial over $\{-1, 1\}^n$. Note that $\text{PARITY}(x_1, \ldots, x_n) = \prod_{i=1}^n x_i$. Let

$$A = \{x \in \{-1, 1\}^n : p(x) = \text{PARITY}(x)\}.$$ 

Let $V = \{f : A \to \mathbb{R}\}$ denote the vector space of functions from $A$ to $\mathbb{R}$. Its dimension is $A$. We will show $|A|$ is small by finding a basis for this space. We already know that any function $f : \{-1, 1\}^n \to \mathbb{R}$ can be written as a polynomial

$$f(x) = \sum_{S \subseteq [n]} f_S \prod_{i \in S} x_i.$$
The crucial observation is that if \( x \in A \) we can replace \( \prod_{i=1}^{n} x_i \) with the low degree polynomial \( p(x) \). Moreover, if \( |S| \geq (n + k)/2 \) then for any \( i \in S \),
\[
\prod_{i \in S} x_i = \prod_{i \in S} x_i \cdot \prod_{i = 1}^{n} x_i \cdot p(x) = \prod_{i \in [n] \setminus S} x_i \cdot i \cdot p(x),
\]
which is a polynomial of degree \( n - |S| + k \leq (n + k)/2 \). Hence, all functions \( f : A \to \mathbb{R} \) can be written as polynomials of degree at most \( (n + k)/2 \). The dimension of this vector space is the number of monomials of that degree,
\[
|A| = \dim(V) = \sum_{i=0}^{(n+k)/2} \binom{n}{i} = (1/2 + O(k/\sqrt{n})) \cdot 2^n.
\]

5 Natural proofs

Natural proofs give a barrier to proof techniques which identify a property of the circuit class which they then demonstrate the hard functions doesn’t have.

**Definition 5.1.** A "natural property" against a circuit class \( C \) is a subset \( P \) of the functions \( f : \{0,1\}^* \to \{0,1\} \) such that
- If \( f_1, \ldots, f_n, \ldots \in C \) then \( f_n \notin P \) for large enough \( n \).
- A random function is with noticeable probability inside \( P \)
- Given \( f : \{0,1\}^n \to \{0,1\} \) we can test if \( f \in P \) in time \( 2^{O(n)} \).

**Theorem 5.2.** If there exists a natural property for \( P/poly \) then any poly-size one-way function can be distinguished in time \( 2^{\varepsilon n} \) for all \( \varepsilon > 0 \).

**Proof.** Let \( F(x) \) be a one-way poly-size function for inputs \( x \in \{0,1\}^n \). Let \( x \in \{0,1\}^k \) for some \( k = n^\varepsilon \) by fixing the other bits to zero. As it is in \( P/poly \) we can check it has property \( P \) in time \( 2^{O(n^\varepsilon)} \). This allows to distinguish it from random functions in sub-exponential time.

Consider for example the property we used to show that \( AC^0 \) cannot compute PARITY: we showed that every small depth circuit can be approximated non-trivially by a low-degree polynomial. Is this a natural property? the simple way to check this is too costly, as to check all polynomials of degree \( k \) we need to enumerate \( \exp(n^k) \) coefficients. However, the proof actually used the following fact: we can write each function as
\[
f(x) = p_1(x) + p_2(x) PARITY(x)
\]
where \( p_1, p_2 \) are polynomials of degree \( \leq n/2 \). Let's define \( P \) to be the property of these functions. Then if \( C(x) \) can be approximated by polynomials of degree \( \text{poly log}(n) \) then \( C(x) \notin P \), as it is not true that all functions can be approximated by a polynomial of degree \( n/2 + \text{poly log}(n) \).