As we remarked in class, it is clear that monotone circuits can only compute monotone functions. Prove the converse.

Why is it not interesting to compute \( \text{PAR} \)?

We proved in class that \( \text{MOD}_{m,k,n} \) circuits with access to each input bit \( x \) and its negation \( \neg x \), where \( \vec{x} \in \{0,1\}^n \) is the input vector. As part of our proof that \( \text{PAR} \not\in \text{AC}^0 \), we showed that if such a circuit computes \( \text{PAR}_n \), it must have size at least \( 2^{n-1} \).

But what if we’re only interested in a circuit that computes \( \text{PAR}_n \) correctly on some subset of a little more than half of the \( 2^n \) different inputs?

2.1. Why is it not interesting to compute \( \text{PAR}_n \) correctly on just \( 2^{n-1} \) inputs? [1 point]

2.2. Show that there is a depth-2 circuit of size \( 2^{O(\sqrt{n})} \) that computes \( \text{PAR}_n \) correctly on at least \( 2^{n-1} + 2\sqrt{n} \) inputs. [9 points]

3. We proved in class that \( \text{PAR} \not\in \text{AC}^0 \), and later seemingly strengthened this by showing \( \text{PAR} \not\in \text{AC}^0[3] \). Prove that this latter result is in fact stronger by showing that \( \text{AC}^0 \subset \text{AC}^0[3] \) (i.e., a proper subset). For this problem, use only the random restrictions technique, and not the approximation-by-polynomials technique. [5 points]

4. Prove that \( \text{MAJ} \not\in \text{AC}^0 \).

Hint: This can be solved using either of the two techniques we used in class to show \( \text{PAR} \not\in \text{AC}^0 \). However, you can give a shorter proof by exhibiting an \( \text{AC}^0 \) circuit that reduces \( \text{PAR} \) to \( \text{MAJ} \). For this approach, it might help to use \( \text{FALSE} = +1 \), \( \text{TRUE} = -1 \) and consider sums of the form \( x_1 + \cdots + x_{n/2} - x_{n/2+1} - \cdots - x_n \). Be careful about separating the two cases: (a) \( n \) is odd (b) \( n \) is even. [10 points]
5. Revisit the random restrictions proof that \( \text{PAR} \notin \text{AC}^0 \) and perform the necessary calculations to obtain a specific quantitative lower bound, in terms of \( n \) and \( d \), on the size of depth-\( d \) circuit that computes \( \text{PAR}_n \). Your bound should be something super-polynomial in \( n \) (for constant \( d \)). Do not worry if you don’t quite get the optimal bound of \( 2^{n^{1/(d-1)}} \) — just derive what you can. [10 points]

6. Let \( p \) and \( q \) be primes with \( p \neq q \). We claimed in class that the approximation-by-polynomials technique can be extended to show that \( \text{MOD}_q \notin \text{AC}^0[p] \). This problem walks you through the proof.

The proof requires a bit of finite field theory, but that shouldn’t daunt you. Here is the crucial fact we need: the finite field \( K := \mathbb{F}_{p^k} \) contains \( \mathbb{F}_p \) (the familiar field consisting of integers mod \( p \)) as a subfield, and also contains a primitive \( q \)-th root of unity, i.e., an element \( \omega \in K \setminus \{0, 1\} \) such that \( \omega^q = 1 \).

Suppose \( C \) is an \( n \)-input \( \text{AC}^0[p] \) circuit with depth \( d \) and size \( s \) that computes the function \( \text{MOD}_q \). As in class, we can assume, thanks to de Morgan’s Laws, that \( C \) contains no \text{AND} gates. We topologically sort \( C \) and proceed to approximate each of its gates, in order, by polynomials over \( \mathbb{F}_p \).

6.1. By generalizing the random subsums construction from class suitably, prove that there exists a polynomial \( h(x_1, \ldots, x_n) \in \mathbb{F}_p[x_1, \ldots, x_n] \) such that
   \begin{itemize}
   \item \( \deg h \leq (p-1)\ell \),
   \item \( \forall \bar{x} \in \{0, 1\}^n : h(\bar{x}) \in \{0, 1\} \), and
   \item \( \Pr[h(\bar{x}) \neq \text{OR}_n(\bar{x})] \leq 1/p^\ell \), with \( \bar{x} \in_R \{0, 1\}^n \).
   \end{itemize}

6.2. Based on your construction above, prove that there exists a polynomial \( f(x_1, \ldots, x_n) \in \mathbb{F}_p[x_1, \ldots, x_n] \) such that
   \begin{itemize}
   \item \( \deg f \leq \sqrt{n} \),
   \item \( \forall \bar{x} \in \{0, 1\}^n : f(\bar{x}) \in \{0, 1\} \), and
   \item \( \Pr[f(\bar{x}) \neq C(\bar{x}) = \text{MOD}_q(\bar{x})] \leq s \cdot p^{-n^{3/(2d)}}/(p-1) \), where \( \bar{x} \in_R \{0, 1\}^n \).
   \end{itemize}

To get these bounds you will need to set \( \ell \) appropriately in the previous construction. [3 points]

6.3. The above gave us a “low degree approximation” to the single Boolean function \( \text{MOD}_q \). By suitably modifying the circuit \( C \), prove that there exists a “large” good set \( A \subseteq \{0, 1\}^n \) on which each of the Boolean functions \( \text{MOD}_{q, k} \) (with \( 0 \leq k \leq q-1 \)) can be represented by a low degree polynomial. State your results precisely. In particular, state a precise lower bound on \( |A| \) and an upper bound on the degree. [5 points]

6.4. Consider the affine map \( \alpha : K \to K \) given by \( \alpha(x) = 1 + (\omega - 1)x \). This map gives us a “notation shift” for functions with Boolean input: 0/1 notation becomes 1/\( \omega \) notation. Applying \( \alpha \) coordinatewise maps the set \( A \) to some set \( A' \subseteq \{1, \omega\}^n \). Based on your earlier observations, prove that the polynomial \( y_1 y_2 \cdots y_n \) agrees with some “low” degree multilinear polynomial \( g(y_1, \ldots, y_n) \in K[y_1, \ldots, y_n] \) on the set \( A' \). [7 points]

6.5. Argue that the equations \( y_i^{-1} = 1 + (\omega^{-1} - 1)(\omega - 1)^{-1}(y_i - 1) \) hold for \( (y_1, \ldots, y_n) \in A' \). [2 points]

6.6. Proceeding as we did in class, prove that every function from \( A' \) to \( K \) can be represented (on \( A' \)) by a multilinear polynomial in \( K[y_1, \ldots, y_n] \) of degree \( \leq n/2 + \sqrt{n} \). Using this, count the number of functions from \( A' \) to \( K \) in two ways to obtain the desired super-polynomial lower bound on \( s \). [8 points]