Recall from last class: Nisan-Wigderson’s construction of PRG from an average-case hard function:

**Theorem 1 (NW88)** If there is a \((\text{poly}(m), 1/2 - 1/\text{poly}(m))\) hard function on \(n\) bits, then exists PRG: \(G: \{0,1\}^\Theta(n^2) \rightarrow \{0,1\}^m\), such that \(G\)-RDP is \(1/3\)-hard for circuits of size at most \(m\).

To bridge the gap, we will see how to obtain an average-case hard function from a worst-case hard function. From the perspective of locally decodable (or locally list-decodable) error-correcting code, (the truth table of) a worst-case hard function is to be sent over a noisy channel, and one tries to recover (decode) the function from its noisy version. We require the decoding to be local, since recovering the whole truth table of a function would be too expensive and unnecessary, all that we need is to recover \(f(x)\) for given input \(x\).

We will use a locally decodable ECC known as Reed-Muller code to encode a worst case function \(f_{wc}\). Due to an inherent limitation of unique decoding property, one only gets a somewhat hard function \(f_{sh}\). From then on we ‘amplify’ the hardness using Yao’s XOR lemma. As a side note, this is usually not satisfactory enough to get full derandomization of BPP. Another approach that bypasses the XOR lemma is to use locally decodable ECC instead.

# 1 Reed-Muller code

Given \(f_{wc} : \{0,1\}^n \rightarrow \{0,1\}\), let its Fourier expansion be \(f_{wc}(x_1, \ldots, x_n) = \sum S \alpha_S x^S\). For large enough \(p\), consider \(f_{sh} : \mathbb{Z}_p^n \rightarrow \mathbb{Z}_p\) by extending \(f_{wc}\) to \(\mathbb{Z}_p\):

\[
f_{sh}(x) = f_{wc}(x) = \sum S \alpha_S x^S \mod p.
\]

**Claim 1** If \(C\) computes \(f_{sh}\) on \(1 - \frac{1}{3(n+1)}\) fraction of inputs, there exists \(C'\) of roughly the same size computing \(f_{wc}\).

We’ll construct a probabilistic circuit \(C''\) such that

\[
\forall x, \Pr_{C''}[C''(x) = f_{sh}(x)] \geq 2/3,
\]

and just let \(C'\) take the majority on \(C''\). \(^1\)

Now we construct \(C''\). The idea is known as self-reducibility. Say we’re interested in \(f_{wc}(x)\), we will decode it from \(f_{sh}(y)\), where \(y\) is chosen as follows: Start with a random direction \(a\), we evaluate \(f_{sh}\) restricted on the line \(y = x + za\) parameterized by \(z\):

\[
fl(z) = \sum S \alpha_S \prod_i (x_i + za_i)
\]

\(^1\)with the ‘best’ randomness over \(C''\) hard-wired in, SEE ALSO the proof of \(\text{BPP} \subset \text{P/poly}\) in textbook of Sipser, or Barak and Arora.
Note that \( f_l(z) \) is a degree \( n \) univariate polynomial, and \( f_l(0) = f_{sh}(x) \), which is the noisy version of \( f_{wc}(x) \). To ‘decode’ \( f_{wc}(x) \), we evaluate \( f_l(z) \) for \( z = 1, \cdots, n + 1 \), since \( a \) is uniformly random, \( y \) will be uniformly random too. Finally we interpolate \( \tilde{f}_l(z) \), and output \( \tilde{f}_l(0) \).

Now by union bound,
\[
\Pr[\exists i : 1 \leq i \leq n + 1, \tilde{f}_l(i) \neq f_l(i)] \leq \frac{(n + 1)^3}{3(n + 1)} = 1/3.
\]

Hence the (probabilistic) circuit that computes \( \tilde{f}_l \) will be the \( C'' \) we need.

Note that here we omitted the steps of truncating the decoded function back to a boolean function. This can be done via concatenating another code that encodes the binary alphabet.

### 2 Hardness amplification and the Hard-core lemma

There are 2 possibilities that a function is ‘somewhat hard’ to compute by small circuits (with probability better than \( 1 - \delta \)):

- the hardness is ‘spread’ out over the boolean cube, different circuits fail on different places.
- there is a single subset of \( \delta \) fraction of inputs such that the function is very hard on those inputs for every small circuits.

The hard-core lemma explains that the latter always happen.

**Lemma 2 (Impagliazzo’s Hard-core lemma)** Let \( f_{sh} \) be any \((m, \epsilon)\)-hard function, then \( \exists H \subset \{0, 1\}^n, |H| \geq \Omega(\epsilon 2^n) \), such that \( \forall C', |C'| \leq m \text{ poly } (\epsilon, \delta) \),
\[
\Pr_{x \in H}[C'(x) \neq f(x)] \geq \frac{1}{2} - \delta.
\]

This can be proved using min-max theorem:

**Theorem 3 (von Neumann’s min-max theorem)** For a zero-sum 2-player game, if we allow randomized strategies, the order of play doesn’t change the outcome.

Specifically let \( A \) be the payoff matrix, and \( x, y \) be distribution over \([n]\) strategies.
\[
\min_x \max_y x^\top Ay = \max_y \min_x x^\top Ay.
\]

**Proof.** Player A’s strategy is to specify a circuit \( C \) that computes \( f_{sh} \). Player B is to find \( S \subset 0, 1^n \) such that \( |S| \geq \epsilon 2^n \). The payoff for \( C, S \) is
\[
P_{C,S} = \Pr_{x \in S}[f(x) = C(x)] - \Pr_{x \in S}[f(x) \neq C(x)].
\]

Suppose the opposite, namely A’s payoff is at least \( \delta \), then by min-max theorem, A has a distribution over circuits so that \( \forall S, |S| \geq \epsilon 2^n \),
\[
\Pr_{C,x \in S}[f(x) = C(x)] - \Pr_{C,x \in S}[f(x) \neq C(x)] \geq \delta.
\]

Let \( \hat{S} = \{ x : \Pr_C[f(x) = C(x)] \leq \frac{1+\epsilon}{2} \} \), since B didn’t choose \( \hat{S} \), it must be the case that \( |\hat{S}| < \epsilon 2^n \).
To this end, we will show a small circuit that is correct for all $x \notin \hat{S}$, this will contradict that $f_{sh}$ is $(m, \varepsilon)$-hard, as $\hat{S} < \varepsilon^{2^n}$.

The idea is as before, we take roughly $n/\delta^2$ independent copies of circuits from A’s distribution, then we take the majority, and call this circuit $C$. By Chernoff bound, $\forall x \notin \hat{S}, \Pr[C(x) \neq f(x)] < 2^{-n}$.

By a union bound, there exists such a $C$ that’s correct for every $x \notin \hat{S}$. Finally we just hard-wire the ‘good randomness’ for such $C$, we get the contradiction as desired.

**Lemma 4 (Yao’s XOR Lemma)** Let

$$F(x_1, x_2, \cdots, x_k) = \sum_i f_{sh}(x_i) \mod 2,$$

and $f_{sh}$ is $(m, \varepsilon)$-hard, then $F$ is $(m \text{ poly } (\varepsilon), 1/2 - 2(1 - \varepsilon)^k)$-hard.

**Proof.** The key argument is that, as long as one of the $x_i$ fall into the hard-core set, then even if one can compute every other $x_j$, the XOR will make $F$ as hard to compute as $f_{sh}$ on the hard-core set. So if we instantiate the hard-core lemma with $\delta = (1 - \varepsilon)^k$, then for circuit $C$,

$$\Pr[C(X) = F(X)] \leq \Pr[\text{none of the } x_i \text{ fall into the hard-core set}] +$$

$$\Pr\left[x_i \text{ fall into the hard-core set, } C(X) \oplus \sum_{j \neq i} f_{sh}(x_j) = f_{sh}(x_i) \right]$$

$$\leq (1 - \varepsilon)^k + 1/2 + (1 - \varepsilon)^k.$$

To this end, we have obtained an average-case hard function from a worst case hard function.