Readers who are familiar with the theory of error-correcting codes can skim through Sections 19.2 and 19.3 in a first reading (pausing to remind themselves of the Reed-Solomon and Reed-Muller codes in Definitions 19.10 and 19.12 and their associated decoding algorithms) and go on to Section 19.4.

19.1 MILD TO STRONG HARDNESS: YAO'S XOR LEMMA

Yao's XOR Lemma transforms a function that has "mild" average-case hardness to a function that has strong average-case hardness. The transformation is actually quite simple and natural, but its analysis is somewhat involved (yet, in our opinion, beautiful). To state the lemma, we need to define precisely the meaning of worst-case hardness and average-case hardness of a function.

Definition 19.1 *(Average-case and worst-case hardness)* For $f : \{0,1\}^n \to \{0,1\}$ and $\rho \in [0,1]$ we define the *ρ-average case hardness of f*, denoted $\mathsf{H}^{\rho}_{\mathsf{avg}}(f)$, to be the largest S such that for every circuit C of size at most S, $\Pr_{x \in_R \{0,1\}^n}[C(x) = f(x)] < \rho$. For an infinite $f : \{0,1\}^* \to \{0,1\}$, we let $\mathsf{H}^{\rho}_{\mathsf{avg}}(f)(n)$ denote $\mathsf{H}^{\rho}_{\mathsf{avg}}(f_n)$ where f_n is the restriction of f to $\{0,1\}^n$.

We define the *worst-case hardness* of f, denoted $\mathsf{H}_{\mathsf{wrs}}(f)$, to equal $\mathsf{H}^{1}_{\mathsf{avg}}(f)$ and define the *average-case hardness of f*, denoted $\mathsf{H}_{\mathsf{avg}}(f)$, to equal $\max\left\{S : \mathsf{H}^{1/2+1/S}_{\mathsf{avg}}(f) \geq S\right\}$. That is, $\mathsf{H}_{\mathsf{avg}}(f)$ is the largest number S such that $\Pr_{x \in_R \{0,1\}^n}[C(x) = f(x)] < 1/2 + 1/S$ for every Boolean circuit C on n inputs with size at most S.

Note that for every function $f : \{0,1\}^n \to \{0,1\}$, $\mathsf{H}_{\mathsf{avg}}(f) \leq \mathsf{H}_{\mathsf{wrs}}(f) \leq O(2^n/n)$ (see Exercise 6.1). This definition of average-case hardness is tailored to the application of derandomization and, in particular, only deals with the uniform distribution over the inputs. See Chapter 18 for a more general treatment of average-case complexity. We can now state Yao's lemma.

Theorem 19.2 *(Yao's XOR Lemma* [Yao82a]*)*
For every $f : \{0,1\}^n \to \{0,1\}$, $\delta > 0$ and $k \in \mathbb{N}$, if $\epsilon > 2(1-\delta)^k$ then

$$\mathsf{H}^{1/2+\epsilon}_{\mathsf{avg}}(f^{\oplus k}) \geq \frac{\epsilon^2}{400n}\mathsf{H}^{1-\delta}_{\mathsf{avg}}(f)$$

where $f^{\oplus k} : \{0,1\}^{nk} \to \{0,1\}$ is defined by $f^{\oplus k}(x_1,\ldots,x_k) = \sum_{i=1}^{k} f(x_i) \pmod 2$.

Yao's Lemma says that if small circuits cannot compute f with probability better than $1-\delta$, then somewhat smaller circuits cannot compute $f^{\oplus k}$ with probability better than $1/2 + 2(1-\delta)^k$. Intuitively, it makes sense that if you can only compute f on a $1-\delta$ fraction of the inputs, then given a random k tuple $x_1,\ldots,x_k$, unless all of these k inputs fall into this "good set" of inputs (which happens with probability $(1-\delta)^k$), you will have to guess the answer to $\sum_{i=1}^{k} f(x_i) \pmod 2$ at random and be successful with probability at most $1/2$; see also Exercise 19.1. But making this

intuition into a proof takes some effort. The main step is the following beautiful result of Impagliazzo.

Lemma 19.3 *(Impagliazzo's Hardcore Lemma* [Imp95a]*) Say that a distribution H over $\{0,1\}^n$ has density δ if for every $x \in \{0,1\}^*$, $\Pr[H = x] \leq 1/(\delta 2^n)$. For every $\delta > 0, f : \{0,1\}^n \to \{0,1\}$, and $\epsilon > 0$, if $\mathrm{H}^{1-\delta}_{\mathrm{avg}}(f) \geq S$, then there exists a density-δ distribution H such that for every circuit C of size at most $\frac{\epsilon^2 S}{100n}$,*

$$\Pr_{x \in_R H}[C(x) = f(x)] \leq 1/2 + \epsilon \qquad \diamond$$

A priori, one can think that a function f that is hard to compute by small circuits with probability $1-\delta$ could have two possible forms: (a) the hardness is sort of "spread" all over the inputs (different circuits make mistakes on different inputs), and the function is roughly $1-\delta$-hard on every significant set of inputs or (b) there is a subset H of roughly a δ fraction of the inputs such that on H the function is *extremely hard* (cannot be computed better than $\frac{1}{2}+\epsilon$ for some tiny ϵ) and on the rest of the inputs the function may be even very easy. Such a set may be thought of as lying at the core of the hardness of f and is sometimes called the *hardcore set*. Impagliazzo's Lemma shows that actually *every* hard function has the form (b). (While the lemma talks about distributions and not sets, it is possible to transform it into a result on sets, see Exercise 19.2.)

19.1.1 Proof of Yao's XOR Lemma using Impagliazzo's Hardcore Lemma

We now show how to use Impagliazzo's Hardcore Lemma (Lemma 19.3) to prove Yao's XOR Lemma (Theorem 19.2). Let $f : \{0,1\}^n \to \{0,1\}$ be a function such that $\mathrm{H}^{1-\delta}_{\mathrm{avg}}(f) \geq S$, let $k \in \mathbb{N}$, and suppose, toward a contradiction, that there is a circuit C of size $S' = \frac{\epsilon^2}{400n} S$ such that

$$\Pr_{(x_1,\ldots,x_k) \in_R U_n^k}\left[C(x_1,\ldots,x_k) = \sum_{i=1}^{k} f(x_i) \pmod 2\right] \geq 1/2 + \epsilon \tag{19.1}$$

where $\epsilon > 2(1-\delta)^k$. We will first prove the lemma for the case $k = 2$ and then indicate how the proof can be generalized for every k.

Let H be the hardcore density-δ distribution obtained from Lemma 19.3, on which every S'-sized circuit fails to compute f with probability better than $1/2 + \epsilon/2$. We can think of the process of picking a uniform element in $\{0,1\}^n$ as follows: First toss a biased coin that comes up "Heads" with probability δ. Then, if the coin came up "Heads" then pick a random element according to H, and if it came up "Tails" pick an element according to the distribution G, which is the "complement" of H. Namely, G is defined by setting $\Pr[G = x] = (2^{-n} - \delta \Pr[H = x])/(1-\delta)$. (Exercise 19.3 asks you to verify that G is indeed a distribution and that this process does indeed yield a uniform element.) We shorthand this and write

$$U_n = (1-\delta)G + \delta H \tag{19.2}$$

If we consider the distribution $(U_n)^2$ of picking *two independent* random strings and concatenating them, then by (19.2) we can write

$$(U_n)^2 = (1-\delta)^2 G^2 + (1-\delta)\delta GH + \delta(1-\delta)HG + \delta^2 H^2 \tag{19.3}$$

where we use G^2 to denote the concatenation of two independent copies of G, GH to denote the concatenation of a string chosen from G and a string chosen independently from H, and so on.

Now for every distribution $\mathcal{D}$ over $\{0,1\}^{2n}$, let $P_{\mathcal{D}}$ be the probability of the event of the left-hand side of (19.1). That is, $P_{\mathcal{D}}$ is the probability that $C(x_1,x_2) = f(x_1)+f(x_2) \pmod 2$ where x_1, x_2 are chosen from $\mathcal{D}$. Combining (19.1) and (19.3) we get

$$1/2 + \epsilon \leq P_{(U_n)^2} = (1-\delta)^2 P_{G^2} + (1-\delta)\delta P_{GH} + \delta(1-\delta)P_{HG} + \delta^2 P_{H^2} \tag{19.4}$$

But since $\epsilon > 2(1-\delta)^2$ and $P_{G^2} \leq 1$, (19.4) implies

$$1/2 + \frac{\epsilon}{2} \leq (1-\delta)\delta P_{GH} + \delta(1-\delta)P_{HG} + \delta^2 P_{H^2} \tag{19.5}$$

Since the coefficients on the right-hand side of (19.5) sum up to less than 1, the averaging principle implies that at least one of these probabilities must be larger than the left-hand side. For example, assume that $P_{HG} \geq 1/2 + \epsilon/2$ (the other cases are symmetrical). This means that

$$\Pr_{x_1 \in_R H, x_2 \in_R G}[C(x_1,x_2) = f(x_1)+f(x_2) \pmod 2] > 1/2 + \frac{\epsilon}{2}$$

Thus by the averaging principle, there exists a fixed string x_2 such that

$$\Pr_{x_1 \in_R H}[C(x_1,x_2) = f(x_1)+f(x_2) \pmod 2] > 1/2 + \frac{\epsilon}{2}$$

or, equivalently,

$$\Pr_{x_1 \in_R H}[C(x_1,x_2) + f(x_2) \pmod 2 = f(x_1)] > 1/2 + \frac{\epsilon}{2}$$

But this means that we have an S'-sized circuit D (the circuit computing the mapping $x_1 \mapsto C(x_1,x_2) + f(x_2) \pmod 2$) that computes f on inputs chosen from H with probability better than $1/2 + \epsilon/2$, contradicting the fact that H is hardcore!

This completes the proof for the case $k = 2$. The proof for general k follows along the same lines, using the equation

$$(U_n)^k = (1-\delta)^k G^k + (1-\delta)^{k-1}\delta G^{k-1}H + \cdots + \delta^k H^k$$

in place of (19.3); we leave verifying the details to the reader as Exercise 19.4. ■

19.1.2 Proof of Impagliazzo's Lemma

We now turn to proving Impagliazzo's Hardcore Lemma (Lemma 19.3). Let f be a function with $H^{1-\delta}_{avg}(f) \geq S$ and let $\epsilon > 0$. To prove the lemma we need to show a density

δ distribution H on which *every* circuit C of size $S' = \frac{\epsilon^2 S}{100n}$ cannot compute f with probability better than $1/2 + \epsilon$.

Let's think of this task as a game between two players named *Russell* and *Noam*. Noam wants to compute the function f, and Russell wants Noam to fail. The game proceeds as follows: Russell first chooses a δ-density distribution H, and then Noam chooses a circuit C of size at most S'. At the game's conclusion, Russell pays Noam v dollars, where $v = \Pr_{x \in_R H}[C(x) = f(x)]$. Assume toward a contradiction that the lemma is false, and hence for every δ-density distribution H chosen by Russell, Noam can find an S'-sized circuit C on which $\Pr_{x \in_R H}[C(x) = f(x)] \geq 1/2 + \epsilon$.

Now this game is a zero-sum game, and so we can use von Neumann's *Min-Max Theorem* (see Note 1) that says that if we allow *randomized* (also known as mixed) strategies, then Noam can achieve the same value even if he plays first. By randomized strategies we mean that Noam and Russell can also select arbitrary distributions over their choices. In Russell's case this makes no difference as a distribution over density-δ distributions is still a density-δ distribution.[1] However in Noam's case we need to allow him to choose a *distribution* $\mathcal{C}$ over S'-sized circuits. Our assumption, combined with the min-max theorem, means that there exists such a distribution $\mathcal{C}$ satisfying

$$\Pr_{C \in_R \mathcal{C}, x \in_R H}[C(x) = f(x)] \geq 1/2 + \epsilon \tag{19.6}$$

for *every* δ-density H.

Call a string $x \in \{0,1\}^n$ "bad" if $\Pr_{C \in_R \mathcal{C}}[C(x) = f(x)] < 1/2 + \epsilon$ and call x "good" otherwise. There are less than $\delta 2^n$ bad x's. Indeed, otherwise we could let H be the uniform distribution over the bad x's and it would violate (19.6). Now let us choose a circuit C as follows: Set $t = 50n/\epsilon^2$, pick $C_1, \ldots, C_t$ independently from $\mathcal{C}$, and define $C(x)$ to equal the majority of $C_1(x), \ldots, C(x)$ for every $x \in \{0,1\}^n$. Note that the size of C is $tS' < S$. We claim that if we choose the circuit C in this way, then for every good $x \in \{0,1\}^n$, $\Pr[C(x) \neq f(x)] < 2^{-n}$. Indeed, this follows by applying the Chernoff bound (see Corollary A.15). Since there are at most 2^n good x's, we can apply the union bound to deduce that there exists a size S circuit C such that $C(x) = f(x)$ for *every* good x. But since there are less than $\delta 2^n$ bad x's, this implies that $\Pr_{x \in_R U_n}[C(x) = f(x)] > 1 - \delta$, contradicting the assumption that $H^{1-\delta}_{avg}(f) \geq S$.

Taken in the contrapositive, Lemma 19.3 implies that if for every significant chunk of the inputs there is some circuit that computes f with on this chunk with some advantage over $1/2$, then there is a single circuit that computes f good probability over all inputs. In machine learning, such a result (transforming a way to weakly predict some function into a way to strongly predict it) is called *boosting* of learning methods. Although the proof we presented here is nonconstructive, Impagliazzo's original proof was constructive and was used to obtain a boosting algorithm yielding some new results in machine learning, see [KS99].

[1] In fact, the set of density δ distributions can be viewed as the set of distributions over $\delta 2^n$-flat distributions, where a distribution is K-flat if it is uniform over a set of size K (see Exercise 19.7). This fact means that we can think of the game as finite and so use the Min-Max Theorem in the form it is stated in Note 19.4.

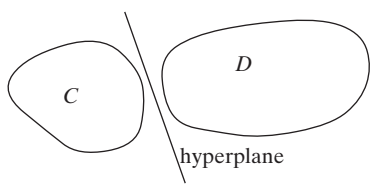
NOTE 19.4 *(The Min-Max Theorem)*

A *zero-sum game* is, as the name implies, a game between two parties in which whatever one party loses is won by the other party. It is modeled by an $m \times n$ matrix $A = (a_{i,j})$ of real numbers. The game consists of only two moves. One party, called the *minimizer* or *column player*, chooses an index $j \in [n]$, while the other party, called the *maximizer* or *row player*, chooses an index $i \in [m]$. The *outcome* is that the column player has to pay $a_{i,j}$ units of money to the row player (if $a_{i,j}$ is negative, then the row player pays the column player $|a_{i,j}|$ units). Clearly, the *order* in which players make their moves is important. The Min-Max Theorem says that, surprisingly, if we allow the players randomized strategies, then the order of play is immaterial.

By *randomized* (also known as *mixed*) strategies, we mean that the column player chooses a *distribution* over the columns; that is, a vector $\mathbf{p} \in [0,1]^n$ with $\sum_{i=1}^{n} p_i = 1$. Similarly, the row player chooses a distribution $\mathbf{q}$ over the rows. The amount paid is the expectation of $a_{i,j}$ for j chosen from $\mathbf{p}$ and i chosen from $\mathbf{q}$. If we think of $\mathbf{p}$ as a column vector and $\mathbf{q}$ as a row vector then this is equal to $\mathbf{q}A\mathbf{p}$. The Min-Max Theorem says that

$$\min_{\substack{\mathbf{p}\in[0,1]^n \\ \sum_i p_i = 1}} \max_{\substack{\mathbf{q}\in[0,1]^m \\ \sum_i q_i = 1}} \mathbf{q}A\mathbf{p} = \max_{\substack{\mathbf{q}\in[0,1]^m \\ \sum_i q_i = 1}} \min_{\substack{\mathbf{p}\in[0,1]^n \\ \sum_i p_i = 1}} \mathbf{q}A\mathbf{p} \tag{19.7}$$

As discussed in Exercise 19.6, the Min-Max Theorem can be proven using the following result, known as the *Separating Hyperplane Theorem*: If C and D are disjoint convex subsets of $\mathbb{R}^m$, then there is a hyperplane that separates them. (A subset $C \subseteq \mathbb{R}^m$ is *convex* if whenever it contains a pair of points $\mathbf{x}, \mathbf{y}$, it contains the line segment $\{\alpha\mathbf{x} + (1-\alpha)\mathbf{y} : 0 \leq \alpha \leq 1\}$ with $\mathbf{x}$ and $\mathbf{y}$ as its endpoints.) We ask you to prove (a relaxed variant of) the separating hyperplane theorem in Exercise 19.5 but here is a "proof by picture" for the two-dimensional case.



19.2 TOOL: ERROR-CORRECTING CODES

Our next goal will be to construct average-case hard functions using functions that are only worst-case hard. Our main tool will be *error-correcting codes*. An error-correcting code maps strings into slightly larger strings in a way that "amplifies differences" in the sense that every two distinct strings (even if they differ by just one bit) get mapped into two strings that are "very far" from one another. The formal definition follows.