1 Chapter 7: Property Testing, PCPPs and CSPs

7.1 Suppose $T$ is an $r$-query local tester for $C$ with rejection rate $\lambda$. Let $T'$ be the algorithm which, on inputs $f, \epsilon, \delta$, runs $T$ on $f$ a total of $k = \Omega\left(\frac{\log(1/\delta)}{\epsilon^2 \lambda}\right)$ times, and accepts iff $f$ passes each of the $k$ tests. Since $T$ has completeness 1, so does $T'$, while if $d(f, C) \geq \epsilon$, we have

$$\Pr[T' \text{ accepts } f] \leq (1 - \lambda \epsilon)^k \leq \delta$$

as desired. \hfill \Box

7.2 Let $\mathcal{M} = \{(x, y) \in \{0, 1\}^{2n} : x = y\}$. Note that $d(z, \mathcal{M}) = \frac{1}{2n} |\{i \in [n] : z_i \neq z_{n+i}\}|$, so let’s have our local tester pick a uniformly random $j \sim [n]$ and accept iff $z_j = z_{n+j}$. The completeness of this test is obviously 1, and in general we see that $z$ is accepted with probability equal to $1 - \frac{1}{2n} |\{i \in [n] : z_i \neq z_{n+i}\}| = 1 - 2d(z, \mathcal{M})$, so the rejection rate is actually 2. \hfill \Box

7.3 We’ll still want our proof string $\Pi$ to contain the partial sums $\Pi_j = \sum_{i=1}^{j+1} w_j \mod 2$, but the last 2 partial sums are unnecessary, since we’re allowed to make 3 queries. Explicitly, let $\Pi = (\Pi_1, \ldots, \Pi_{n-3})$, and perform one of the following checks, uniformly at random:

$$\Pi_1 = w_1 + w_2$$
$$\Pi_2 = \Pi_1 + w_3$$
$$\ldots$$
$$\Pi_{n-3} = \Pi_{n-4} + w_{n-2}$$
$$\Pi_{n-3} + w_{n-1} + w_n = 1$$

7.4 Suppose $d(w, \mathcal{E}) = \epsilon$ and set $A = \{i : w_i = 1\}$. Then $\left\{\frac{|A|}{n}, 1 - \frac{|A|}{n}\right\} = \{\epsilon, 1 - \epsilon\}$. Then picking $i, j$ uniformly and independently from $[n]$, the probability that $w_i \neq w_j$ is exactly $2\epsilon(1 - \epsilon) = 2\epsilon - 2\epsilon^2$, which is at least $\epsilon$ since $\epsilon = d(w, \mathcal{E}) \leq 1/2$. \hfill \Box

7.5 Suppose $T$ queries the $n - 1$-bits in locations $i_1, \ldots, i_{n-1}$, and computes some predicate $\phi$ on those bits. I claim $\phi \equiv 1$: indeed, for any values of $x_{i_1}, \ldots, x_{i_{n-1}}$, we can always find a string $y$ in $\mathcal{O}$ such that $y_{i_k} = x_{i_k}$ for all $k = 1, \ldots, n - 1$, just by choosing the correct value of some bit not included in this list. Thus, since $T$ accepts $y$ with probability 1,
\( \phi \) must be identically 1. The same is true of every predicate used in \( T \), so it accepts all strings with probability 1.

7.6 Let \( T \) be a 2-query testing algorithm which accepts all dictators with probability 1. Let \( \phi : \{0,1\}^2 \to \{0,1\} \) be any predicate and \( x \) and \( y \) be any strings in \( \{-1,1\}^n \) used by \( T \) with positive probability, such that \( \phi(\text{MAJ}_n(x), \text{MAJ}_n(y)) = 0 \). There are four possible values for \( z := (\text{MAJ}_n(x), \text{MAJ}_n(y)) \), and in each case we reach a contradiction, as follows. If \( z = (0,0) \), then by the pigeonhole principle there must be some \( i \in [n'] \) such that \( x_i = 0 = y_i \), in which case \( (\chi_i(x), \chi_i(y)) = (0,0) \), which would cause \( T \) to reject the dictator \( \chi_i \) with positive probability – a contradiction. A similar argument works for \( z = (1,1) \); if \( z = (1,0) \), then \( x \) has more 1’s than \( y \), so in particular there exists \( i \in [n'] \) with \( x_i = 1, y_i = 0 \), and thus \( \chi_i \) fails the test (and similarly for \( z = (0,1) \)). Therefore there is no 2-query local tester for dictatorship assuming \( n > 2 \).

7.7 Let \( \alpha < 1 \), and \( \lambda \) be any function with \( \lambda(\epsilon) \to 0 \) as \( \epsilon \to 0 \). Pick \( \epsilon > 0 \) such that \( \alpha + \lambda(\epsilon) < 1 \). Suppose \( T \) is a test using only predicates of the form \( f(x) + f(y) + f(z) = b \) which accepts all dictators with probability 1. Then for each such predicate, the strings \( x, y, z \) in the support of \( T \)’s distribution must satisfy \( x_i + y_i + z_i = b \) for all \( i \in [n] \). Then for any odd-sized subset \( S \subset [n] \), we have \( \chi_S(x) + \chi_S(y) + \chi_S(z) = |S|b = b \mod 2 \), and so all odd parities pass each of \( T \)’s tests with probability 1 as well. But \( I_{\epsilon}[\chi_S] \leq (1 - \epsilon)^{|S| - 1} \), which is \( \leq \epsilon \) for \( |S| \) sufficiently large. This means \( T \) cannot be a \((\alpha, 1)\)-dictator-vs-no-notables test.

7.8 (a) By symmetry, we may assume \(|S| = k = n\). Then in the \( \pm 1 \) setting, we have

\[
\chi_S(x) \land \chi_S(y) = \frac{1}{2}(1 + \prod_{i=1}^{n} x_i + \prod_{i=1}^{n} y_i - \prod_{i=1}^{n} x_i y_i) \tag{2}
\]

\[
\chi_S(z) = \frac{1}{2^n} \prod_{i=1}^{n} (1 + x_i + y_i - x_i y_i) \tag{3}
\]

From here it’s easy to compute \( E_{x,y}[(\chi_S(x) \land \chi_S(y)) \chi_S(z)] = (\frac{3}{2} + \frac{(-1)^k}{2^n})2^{-k} \), and thus \( p_k = \Pr[\chi_S(x) \land \chi_S(y) = \chi_S(z)] = \frac{1}{2} + (\frac{3}{4} + \frac{(-1)^k}{4})2^{-k} \), which is, of course, always at most \( \frac{1}{2} + 2^{-|S|} \).

(b) Call the above test \( A \). Consider a new test \( T \) which performs \( A \) on \( f \) (after locally correcting on \( z \)) with probability 1/2 and the Odd BLR Test with probability 1/2. Suppose \( T \) accepts \( f \) with probability at least \( 1 - \lambda \epsilon \). Then the Odd BLR Test accepts \( f \) with probability at least \( 1 - 2\lambda \epsilon \), and hence \( d(f, \chi_S) \leq 2\lambda \epsilon \) for some \( S \) with odd cardinality. Since \( f \) is \( 2\lambda \epsilon \)-close to \( \chi_S \), when \( A \) asks for the value of \( f \) on the uniform, independent random inputs \( x \) and \( y \), we’ll just give it \( f(x) \) and \( f(y) \), and when it asks for \( f(z) \), we’ll give it \( f(z + w) \) for a random string \( w \). Thus, we’ll be giving \( A \) the true values of \( \chi_S(x), \chi_S(y), \chi_S(z) \) with probability at least \( 1 - 8\lambda \epsilon \). Since this portion of the test must accept with probability at least \( 1 - 2\lambda \epsilon \), we see that \( p_k \geq 1 - 10\lambda \epsilon \), for \( k = |S| \). Picking \( \lambda = \frac{1}{30} \) implies that both \( d(f, \chi_S) < \epsilon \) (for some \( |S| \) odd) and \( \frac{1}{2} + 2^{-|S|} \geq \frac{2}{3} \). The only odd value of \( |S| \) for which this last inequality holds is \( |S| = 1 \), and hence \( f \) is \( \epsilon \) close to a dictator.
7.9 Rather than keeping track of all the higher order terms and obtaining a (slightly worse) rejection rate that works for all $\epsilon \in [0,1]$, I’ll ignore terms which are $o(\epsilon)$ as $\epsilon \to 0$, and hence obtain a rejection rate that works for sufficiently small $\epsilon$. If we perform the BLR test with probability $p$, and the overall test accepts $f$ with probability $1 - \lambda \epsilon$, then the BLR test accepts with probability $1 - \frac{\lambda}{p} \epsilon$, and so, letting $\chi_{S^*}$ be the closest linear function to $f$ and $d = d(f, \chi_{S^*})$, we have $\hat{f}(S^*) = 1 - 2d$ and $|\hat{f}(S)| \leq 2d$ for $S \neq S^*$, hence

$$1 - 2\frac{\lambda}{p} \epsilon = \sum_S \hat{f}(S)^3 \leq \hat{f}(S^*)^3 + 2d(1 - \hat{f}(S^*)^2) = 1 - 6d + O(d^2) \quad (4)$$

which implies $d \leq \frac{1}{6p} + o(\epsilon)$. On the other hand, if the NAE test accepts with probability $1 - \frac{\lambda}{q} \epsilon$, then know $W^1[f] \geq 1 - \frac{\lambda a}{2q} \epsilon$ and so by the (strong form) of the FKN theorem, we know $f$ is $\frac{\lambda a}{2q} \epsilon + o(\epsilon)$-close to some (possibly negated) dictator. A fine choice of $p$ and $q$ is to set our two guarantees equal, which yields $p = 8/35$, and gives a rejection rate of $\lambda = \frac{24}{35} - o(\epsilon)$.

7.10 (a) If $A$ is an $(\alpha, \beta)$-approximation algorithm for a CSP $\Phi$, then consider the following algorithm $B$: run $A$ on $\Phi$ and accept iff $\text{val}(A(\Phi)) \geq \alpha$. Indeed, if $\text{val}(\Phi) < \alpha$, then certainly $\text{val}(A(\Phi)) < \alpha$, and so $B$ will reject. If $\text{val}(\Phi) \geq \beta$, then $\text{val}(A(\Phi)) \geq \alpha$ and hence $B$ will accept. So, $B$ is an $(\alpha, \beta)$-distinguisher for $\Phi$.

(b) If $A$ is a $(1,1)$-distinguisher for $\Phi$, we can use $A$ a total of $|\Sigma|n$ times to find a satisfying assignment of all the constraints in $\Phi$, where $n$ is the number of variables in $\Phi$ and $\Sigma$ is the alphabet. Indeed, suppose $\text{val}(\Phi) = 1$. We begin by replacing each constraint $\phi(x_1, \ldots, x_n)$ with $\phi(a, x_2, \ldots, x_n)$, for each $a \in \Sigma$, and hit each resulting CSP $\Phi_\alpha$ with $A$ – since $\text{val}(\Phi) = 1$, our distinguisher $A$ must accept at least one of them. Repeating this for $x_2, \ldots, x_n$, we end up with an assignment $(a_1, \ldots, a_n)$ with $\text{val}(a_1, \ldots, a_n) = 1$. □

7.11 (a) For each clause $C_i = \bigvee_{j=1}^k x_j^i$ in $\phi$ of width $k > 3$, we introduce $k - 3$ new variables $\Pi_1, \ldots, \Pi_{k-3}$, and replace it with the $k - 2$ new clauses

$$(x_1^i \lor x_2^i \lor \Pi_1^i)$$

$$(x_3^i \lor -\Pi_3^i \lor \Pi_2^i)$$

$$\ldots$$

$$(x_{k-2}^i \lor -\Pi_{k-4}^i \lor \Pi_{k-3}^i)$$

$$(x_{k-1}^i \lor x_k^i \lor -\Pi_{k-3}^i)$$

Clearly our new CNF $\phi'$ has width 3, size at most $(w - 2)s$ and at most $(w - 3)s$ new variables. If $\phi(x) = 0$, then $\phi'(x, \Pi) = 0$ for all $\Pi$, since to make $(x_1^i \lor x_2^i \lor \Pi_1^i) = 1$ we need to set $\Pi_1^i = 1$, which forces $\Pi_2^i = 1$, and so on, until we reach a contradiction at the final clause, $(x_{k-1}^i \lor x_k^i \lor -\Pi_{k-3}^i)$. Conversely, if $\phi(x) = 1$, then for each $i$, there is some $j$ such that $x_j^i = 1$. Then we can set $\Pi_1^i = \Pi_2^i = \cdots = \Pi_{j-2}^i = 1$, and set $\Pi_{j-1}^i = \cdots = \Pi_{k-3}^i = 0$. So, $\phi$ and $\phi'$ are equivalent in this sense.

(b) If any clause $C_i = (x_i^i)$ in $\phi$ has width 1, we instead replace it with the four clauses $(x_i^i \lor \Pi_1^i \lor \Pi_2^i), (x_i^i \lor -\Pi_1^i \lor \Pi_2^i), (x_i^i \lor -\Pi_1^i \lor -\Pi_2^i), (x_i^i \lor \Pi_1^i \lor -\Pi_2^i)$. Similarly, if $C_i = (x_i^i \lor x_2^i)$ has width 2, we replace it with the two clauses $(x_1^i \lor x_2^i \lor \Pi_1^i)$ and $(x_1^i \lor x_2^i \lor -\Pi_1^i)$. This
ensures that \( \phi' \) has size at most \((w - 2)s + 3s = (w + 1)s\) and each clause has width exactly 3 and contains 3 distinct variables. (If \( \phi \) has any repeated variables within a clause, we can just eliminate those during the construction of \( \phi' \)).

7.12 Given an \( r \)-query PCPP system \( T \) with rejection rate \( \lambda \), proof length \( \ell \) and description length \( m \), we show how to construct a 3-query PCPP system \( T' \) for the same language with rejection rate at least \( \lambda / r^2 \), proof length \( \ell + r^2m \), all of whose predicates are 3-bit ORs. The idea is that each of \( T' \)'s predicates \( \phi : \{0,1\}^r \rightarrow \{0,1\} \) have size \( 2^r \) width \( r \) CNFs which can be converted into size \( r^2 \) 3-CNFs \( \phi' \), and we can investigate the truth of any random clause in \( \phi' \) with only 3 queries. More explicitly, given an input-proof pair \((w,\Pi')\), we’ll expect our proof \( \Pi' \) to consist of 1) first, a valid proof \( \Pi \) for \( T \), and then 2) for each possible outcome \( R \) of \( T \)'s randomness, include an assignment of the auxiliary variables \( \Pi(R) \) such that \( \phi'_R(x(R),\Pi(R)) = 1 \). Then \( T' \) will do the following: pick a random \( R \) and a random clause of \( \phi'_R \), and query the three bits from \((w,\Pi')\) which come up in that clause. If the clause is satisfied, accept; reject otherwise.

Completeness is obvious; for soundness, just note that if \( \phi(x) = 0 \), then for any \( \Pi' \), there must be some clause of \( \phi'(x,\Pi') \) which is unsatisfied, and our algorithm finds it with probability at least \( 1/r^2 \). It’s also clear that this construction of \( T' \) can be done in time \( \text{poly}(\ell + r^2m, n, \text{size}(C)) \) if \( T \) has a \( \text{poly}(\text{size}(C), \ell) \) time construction.

7.13 (a) Given a circuit \( C \) with gates \( G_1, \ldots, G_s \) (including input gates), we construct the following formula in the variables \( g_1, \ldots, g_s \). For each gate \( G_i \) of type \( \Psi \) with input wires from gates \( G_j \) and \( G_k \), we set \( \phi_i(g_i, g_j, g_k) = (g_i \iff \Psi(g_j, g_k)) \). For \( \Psi \in \{\lor, \land, \neg\} \) it is easy to express such formulas as 3-CNFs with \( O(1) \) size. Thus, if \( G_s \) is the top gate of \( C \), we have that \( C(x) \) is satisfiable iff \( \Phi \) is, where \( \Phi \) is the 3-CNF

\[
\Phi(g_1, \ldots, g_s) = g_s \land \bigwedge_{i=1}^{s} \phi_i(g_i, g_j, g_k) \tag{5}
\]

(b) This reduction from CIRCUIT-SAT to 3-SAT is clearly computable in polynomial (actually linear) time, and so if one can \((1,1)\)-distinguish MAX-3-SAT (i.e. decide 3-SAT) in polynomial time, one can also decide CIRCUIT-SAT in polynomial time.

(c) Using Ex. 7.11, any 3-CNF \( \phi \) can be converted efficiently into a 3-CNF \( \phi' \) with exactly 3 distinct variables in each clause, with the property that \( \phi \) is satisfiable iff \( \phi' \) is. Hence \((1,1)\)-approximating (or even distinguishing) MAX-E3-SAT is NP-hard.

7.14 A graph has a cut of value 1 iff it is bipartite. To find a bipartition \((U, W)\) of a bipartite graph \( G \), simply select a vertex \( v \) and put it in \( U \). Starting at \( v \), perform a BFS through \( G \), putting a node in \( W \) if it is adjacent to a node in \( U \), and vice versa. This will find the unique bipartition of each connected component of \( G \) in polynomial time.

7.15 (a) Given \( f : \mathbb{F}_2^n \rightarrow \{-1, 1\} \), consider the 3-query local tester which picks \( x, y \) uniformly and independently at random from \( \mathbb{F}_2^n \) and \( h \) uniformly at random from the subspace \( H \), and accepts iff \( f(x + y)f(x + h) = f(y) \). We can use the Poisson summation formula to
compute the quantity $E_{x,y}[E_h[f(x+h)f(x+y)f(y)]]$

$$= E_{x,y} \left[ \left( \sum_{\gamma \in H^\perp} \chi(x)f(\gamma) \right) \left( \sum_{\gamma' \in F^2} \hat{f}(\gamma') \chi_{\gamma'}(x) \chi_{\gamma'}(y) \right) \left( \sum_{\gamma'' \in F^2} \hat{f}(\gamma'') \chi_{\gamma''}(y) \right) \right]$$

$$= \sum_{\gamma \in H^\perp} \hat{f}(\gamma)^3$$

From here, the analysis proceeds exactly as in the BLR linearity test, and like BLR we achieve a rejection rate of 1.

(b) If in general $A = a + H$, then to test for the class $H = \{ \chi_\gamma : \gamma \in A \}$ we let $x, y$ be uniform and independent, choose $h$ uniformly from $H^\perp$, and accept iff $f(x+h)f(y) = f(x+y)\chi_a(h)$. We’re interested in the quantity $E_{x,y}[E_h[f(x+h)\chi_a(h)f(x+y)f(y)]]$, so observe that

$$f(x+h)\chi_a(h) = \sum_{\gamma \in F^2} \hat{f}(\gamma) \chi(\gamma) \chi_{\gamma+a}(h) \quad (6)$$

$$\implies E_{h \sim H^\perp}[f(x+h)\chi_{\gamma+a}(h)] = \sum_{\gamma \in a+H} \hat{f}(\gamma) \chi(\gamma) \quad (7)$$

and thus, after carrying out the rest of the computation as above, we see that the probability of passing the test is $\frac{1}{2} + \frac{1}{2} \sum_{\gamma \in A} \hat{f}(\gamma)^3$, and so again we have rejection rate 1. 

7.16 Given $w \in F_2^n$, we expect the proof $\Pi$ to contain the truth table of $\chi_w$. With probability $1/2$, we perform the test from 7.15 on $\Pi$ (interpreted as a function $F_2^n \rightarrow \{-1, 1\}$); with probability $1/2$, we choose $i \sim [n]$ uniformly at random and locally correct $\Pi$ on $e_i$ (i.e. compute $\Pi(x+e_i)\Pi(x)$) and accept iff the result equals $w_i$. Clearly this test will accept $(w, \chi_w)$ with probability 1 when $w \in A$.

Set $\lambda = 1/8$. If this test passes with probability $1 - \lambda \epsilon$, then each subtest passes with probability at least $1 - 2\lambda \epsilon$. In particular, $\Pi$ must be $2\lambda \epsilon$-close to a function $\chi_{w'}$ for some $w' \in A$. Also, since the second test passes with probability at least $1 - 2\lambda \epsilon$, we must have

$$E_{i \sim [n]}[Pr_x[\Pi(x+e_i)\Pi(x) = w_i]] \leq 2\lambda \epsilon \quad (8)$$

and so by Markov’s inequality, the fraction of $i$ for which $Pr_x[\Pi(x+e_i)\Pi(x) = w_i] \geq 1/2$ is at most $4\lambda \epsilon$. However, if $w_i \neq w'_i$, then this probability is at least $1 - 4\lambda \epsilon \geq 1/2$, and hence $d(w, w') \leq 4\lambda \epsilon \leq \epsilon$. 

7.17 (a) Let $C = \{(x, y) \in F_2^{2n} : \langle x, y \rangle = 1\}$. First observe that $d(w, C) \leq 1/n$ for all strings $w \in F_2^{2n}$, so it suffices to obtain a PCPP with rejection probability $\Omega(1/n)$ whenever $w \not\in C$. We expect our proof to consist of the string $(a_1, a_2, \ldots, a_n) = (x_1y_1, x_2y_2, \ldots, x_ny_n)$, followed by $(\Pi_1, \ldots, \Pi_{n-1})$, where $\Pi_i = \sum_{j=1}^{i+1} a_j$ are the partial sums. We perform a
(uniform) random check from the following list:

\[ a_1 = x_1 y_1 \]
\[ \ldots \]
\[ a_n = x_n y_n \]
\[ \Pi_1 = a_1 + a_2 \]
\[ \ldots \]
\[ \Pi_{n-1} = \Pi_{n-2} + a_n \]
\[ \Pi_{n-1} = 1 \]

Each test involves at most 3 queries, and if \( w \notin C \), then at least one of them must go wrong, and we’ll detect this with probability at least 1/2n, as desired.

(b) It’s easy to see that \( CQ_n(x) = \sum_{i \neq j} x_i x_j = \frac{S(x)^2 - S(x)}{2} \), where \( S(x) = \sum_i x_i \), and thus \( CQ_n(x) = 1 \iff S(x) \equiv 2 \text{ or } 3 \mod 4 \). Since \( S(x) \) can always be made congruent to 2 or 3 modulo 4 by flipping at most 2 bits, we know that \( d(w, C) \leq 2/n \) for all \( w \).

To make a 3-query PCPP for \( C \) with \( O(n) \) proof length, we first design one over the alphabet \( \{0, 1, 2, 3\} \), which can be viewed as a 6-query PCPP over \( \{0, 1\} \), which can then be converted to a 3-query PCPP using Exercise 7.12, without too much loss in the parameters. Indeed, our first PCPP will work just like the one for parity, but performs all the arithmetic in \( \mathbb{Z}_4 \) instead of \( \mathbb{Z}_2 \), which is why we need a larger alphabet. But of course, this alphabet can be expressed via 2-bit combinations from \( \{0, 1\} \), and so our 3-query PCPP becomes a 6-query PCPP over \( \{0, 1\} \), with proof length \( O(n) \), rejection rate 1/2, and description length \( O(n) \). Using the construction from Exercise 7.12, it can be converted into a 3-query PCPP with proof length \( O(n) \) and rejection rate at least 1/768.

7.18 (a) Since \( D \) is a non-zero matrix, its null space \( N \) is a proper subspace of \( \mathbb{F}_2^n \), and thus the probability that \( x \in N \) is at most 1/2. Given that \( Dx \neq 0 \), the probability that \( (y, Dx) = 0 \) is exactly 1/2, and hence \( \Pr[y^T Dx \neq 0] \geq 1/4 \).

(b) Observe that
\[
(\gamma^T x)(\gamma^T y) - \Gamma \cdot (xy^T) = \sum_{i,j} y_j (\gamma_i \gamma_j - \Gamma_{ij}) x_i = y^T (\gamma\gamma^T - \Gamma)x
\]
which is always zero when \( \Gamma = \gamma\gamma^T \), and is otherwise non-zero with probability at least 3/4 by (a).

(c) Consider the following test: given \( \ell : \mathbb{F}_2^n \to \mathbb{F}_2 \) and \( q : \mathbb{F}_2^n \to \mathbb{F}_2 \), with probability 1/3, we perform a BLR linearity test on \( \ell \); with probability 1/3, we perform a BLR linearity test on \( q \); with probability 1/3, we pick \( x, y \sim \mathbb{F}_2^n \) and \( M \sim \mathbb{F}_2^n \times \mathbb{F}_2^n \) independently and uniformly at random, and accept iff \( \ell(x)\ell(y) = q(M + xy^T)q(M) \).

Suppose the overall test accepts with probability \( 1 - \lambda \epsilon \), for \( \lambda = 1/50 \). Then each subtest must accept with probability at least \( 1 - 3\lambda \epsilon \), and hence \( \ell \) and \( q \) are \( 3\lambda \epsilon \)-close to functions \( \chi_\gamma : \mathbb{F}_2^n \to \mathbb{F}_2 \) and \( \chi_\Gamma : \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{F}_2 \) respectively. This implies that \( \ell(x)\ell(y) = (\gamma^T x)(\gamma^T y) \) with probability at least \( 1 - 6\lambda \epsilon \), and \( q(M + xy^T)q(M) = \Gamma \cdot (xy^T) \) with probability at
least $1 - 6\lambda \epsilon$, and hence $(\gamma^T x)(\gamma^T y) = \Gamma \cdot (xy^T)$ with probability at least $1 - 12\lambda \epsilon \geq 38/50 > 3/4$, and hence $\gamma \gamma^T = \Gamma$. So, $\ell$ must be $3\lambda \epsilon < \epsilon$-close to $\chi_\gamma$ and $q$ is $3\lambda \epsilon < \epsilon$-close to $\chi_\gamma \gamma^T$.

(d) Given a string $w$ and proof $\Pi \in \{0,1\}^{2n^2 + 2n^2}$, we think of $\Pi$ as the truth tables of a pair of functions $\ell : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$ and $q : \mathbb{F}_2^{n} \times \mathbb{F}_2 \rightarrow \mathbb{F}_2$. Consider the following 4-query test: with probability $1/3$, we perform the test from (c) on $\ell$ and $q$; with probability $1/3$, we select $x \sim \mathbb{F}_2^n$ and $i \sim [n]$ uniformly at random and test whether $\ell(x + e_i)\ell(x) = w_i$; and with probability $1/3$, we apply the local tester from Ex. 7.15(b) for the set $H := \{\chi_{\Gamma} : \Gamma \in A_L\}$, where $A_L \subset \mathbb{F}_2^{n^2}$ is the affine subspace of solutions to the system $L$, viewed as linear equations in $n^2$ variables.

Suppose this test passes with probability $1 - \lambda \epsilon$, for $\lambda = 1/300$. Then $\ell$ is $\epsilon/100$ close to $\chi_\gamma$, for some $\gamma \in \mathbb{F}_2^n$ and $\epsilon/100$ close to $\chi_\gamma \gamma^T$. Since $q$ passes the test from 7.15(b) with probability at least $1 - \epsilon/100$, we know it is $\epsilon/100$-close to some $\chi_{\Gamma'}$ with $\Gamma' \in A_L$. But since $d(\chi_{\Gamma}, \chi_{\Gamma'}) = 1/2$ for $\Gamma \neq \Gamma'$, the triangle inequality implies that $\Gamma = \gamma \gamma^T \in A_L$. Finally, the $\ell(x + e_i)\ell(x) = w_i$ test being likely to pass is what tells us that $w$ and $\gamma$ must be close – using an argument like the one used in Ex 7.16, we conclude that $w$ and $\gamma$ are $\epsilon$-close.

(e) Given a circuit $C$, we construct a PCPP for $\{w \in \{0,1\}^n : C(w) = 1\}$ as follows. On input $(w, \Pi)$, our tester will expect $\Pi = (z, \ell, q)$ to consist of the values $g = (g_i)$ of each of $C$’s gates on input $w$, along with the truth tables of $\chi_g$ and $\chi_{g^0}$. Observe that we can express the condition that a gate assignment $(g_i)$ satisfies $C$ in terms of a system $L$ of quadratic equations over $\mathbb{F}_2^{\text{size}(C)}$, and we can construct such a system in poly($\text{size}(C)$) time: for each NOT gate $g_i$ with an incoming wire from $g_j$, we add the equation $g_i + g_j = 1$; for each OR gate $g_i$ with incoming wires from $g_j$ and $g_k$, we add the equation $g_i + g_j + g_k + g_jg_k = 0$; for each AND gate $g_i$ we add $g_i + g_jg_k = 0$, and then finally for the top gate $g_s$ we add $g_s = 1$. Set $\mathcal{L} := \{z \in \mathbb{F}_2^{\text{size}(C)} : z$ satisfies $L\}$.

After constructing $L$, our tester works as follows: given $(w, z, \ell, q)$, with probability $1/2$ we run our PCPP system from (d) on $(z, \ell, q)$, and with probability $1/2$ we check for consistency between $w$ and the values of $\ell$ corresponding to the input gates – namely, we choose a random $i \sim [n]$ and $x \sim \mathbb{F}_2^{\text{size}(C)}$ and check if $\ell(x + e_i)\ell(x) = w_i$.

For $\lambda = 1/600$, if the overall test passes with probability $1 - \lambda \epsilon$, then we know that $z$ is $\epsilon$-close to a satisfying assignment $g$ of the gates, and that $\ell$ is $\epsilon$-close to $\chi_g$. But since $\ell(x + e_i)\ell(x) = w_i$ with probability at least $1 - 2\lambda \epsilon$, we know from earlier analysis that $w_i = g_i$ for at least an $\epsilon$-fraction of $i \in [n]$. Completeness is obvious as usual.

This is a 4-query PCPP reduction with proof length $2^{O(\text{size}(C)^2)}$, but using 7.12 we can reduce the number of queries to 3, while the proof length is still only $2^{O(\text{size}(C)^2)}$, thus establishing Theorem 7.19.

7.19 If $\lambda$ is the rejection rate of a PCPP for a circuit $C$ using predicates $\mathcal{P}$, then since every string is $1$-far from the empty property, if $C$ is unsatisfiable, we must have $\text{val}(\mathcal{P}) \leq 1 - \lambda$. \hfill \box

7.20 (a) If $A$ is a deterministic algorithm which produces a randomized assignment $F : V \rightarrow \text{Prob}(\Omega)$, then consider the randomized algorithm $A'$, which, for each $v_i \in V$, uses $A$ to
compute the distributions $F(v_i)$, and then makes random choices $F'(v_i) \sim F(v_i)$, and outputs the resulting assignment $F' : V \rightarrow \Omega$. The value of $F'$, in expectation, is clearly equal to

$$E[E_{(S,\Psi) \sim \mathcal{P}}[\Psi(F(S))]]$$

(10)

(b) By (a), the value of this randomized assignment is the same as the expected value of a deterministic assignment generated by giving each variable a label in $\{0, 1\}$ uniformly and independently at random. Any given clause of a MAX-E3-SAT instance has a $7/8$ probability of being satisfied by this assignment, and hence the expected value is exactly $7/8$, which means this is a $(7/8, \beta)$-approximation algorithm for any $\beta$. Essentially the same argument shows that this is a $(1/2, \beta)$-approximation algorithm for MAX-3-Lin.

(c) Write $\psi(v_1, \ldots, v_r) = \sum_{S \subseteq [r]} \hat{\psi}(S)v^S$, and so the value of the randomized assignment with $f(v_i) = \mu_i$ is

$$E[\sum_{S \subseteq [r]} \hat{\psi}(S)v^S] = \sum_{S \subseteq [r]} \hat{\psi}(S)E[v^S] = \sum_{S \subseteq [r]} \hat{\psi}(S)\mu^S = \psi(f(v_1), \ldots, f(v_r))$$

(11)

(d) By (c), under the randomized assignment $f = 0$, the value of any constraint $(S, \psi)$ is $\hat{\psi}(\emptyset)$, and hence, taking the expectation over all constraints, we see that the value of $f$ is at least $\nu = \min_{\psi \in \Psi} \hat{\psi}(\emptyset)$, which is a $(\nu, \beta)$-approximation for any $\beta$. □

7.21 This is essentially the method of conditional expectation – for each possible labeling of the first variable, we compute the expected value that would be obtained if the remaining variables were picked according to the distributions prescribed by $F$, and give the first variable the label which maximizes this quantity. We repeat this iteratively for the other labels, each time maintaining an expected value of at least $\alpha$, until we are finished and have a deterministic assignment of value at least $\alpha$. □

7.22 Even though, $x, y$ and $z'$ are not jointly independent, one they have been selected, the values of the random bits $f(x), f(y), f(z')$ are independent – in other words, conditioned on the choice of strings, the probability the test passes is exactly $\frac{1}{2} + \frac{1}{2}f(x)f(y)f(z')b \in [0, 1]$. Therefore, taking the expectation over the choice of strings, we get the same expression for the success probability, namely

$$\Pr[f \text{ passes Hastad}_{\delta} \text{ test}] = \frac{1}{2} + \frac{1}{2} \sum_{|S| \text{ odd}} (1 - \delta)^{|S|} \hat{f}(S)^3$$

(12)

From here, all of the analysis persists, since the only place we used $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$ was to say $\sum_S f(S)^2 = 1$, but all we need is $\sum_S \hat{f}(S)^2 \leq 1$, which is true for $f : \{-1, 1\}^n \rightarrow [-1, 1]$. □

7.23 (a) Since $f^{\text{odd}}(x) = \frac{f(x) - f(-x)}{2}$, this is basically just a restatement of linearity of expectation.

(b) Since dictators are odd, and $I^{(1-\epsilon)}_i[f] \geq I^{(1-\epsilon)}_i[f^{\text{odd}}]$, it follows from (a) that the guarantees of an $(\alpha, \beta)$-Dictators-vs-no-notables test carry over to functions which aren’t odd as well. □
7.24 (a) Since $E_{y \sim N_1-\delta}(x)[h(y)] = T_{1-\delta}h(x)$, it follows that $U'$ accepts $h$ with the same probability as $U$ accepts $T_{1-\delta}h$.

(b) Since $I_1^{(1-\epsilon)}[h] \geq I_1^{(1-\epsilon)}[T_{1-\delta}h]$, it only remains to investigate what $U'$ does with dictators. If $h$ is a dictator, then $T_{1-\delta}h = (1 - \delta)h$, which we interpret as the function which returns the true value $h(x) = x_i$ with probability $1 - \delta/2$, and the incorrect value with probability $\delta/2$. Since all predicates have arity-$r$, the probability $U$ treats $T_{1-\delta}h$ differently from $h$ is at most $r\delta/2$ by a union bound. Hence, it accepts dictators with probability at least $\beta - r\delta/2$, as desired. \hfill \Box

7.25 Using exercises 7.23 and 7.26, it suffices to design a test using 3XOR predicates with the following guarantees: if $f : \{-1,1\}^n \to \{-1,1\}$ is odd, then if $\inf_i[f] \leq \epsilon$ for all $i$, the probability $f$ is accepted is $O(\epsilon)$, while if $f$ is a dictator, it is accepted with probability 1. However, the usual BLR test certainly has these guarantees for odd $f$. \hfill \Box

7.26 Suppose there exists an $(\alpha, \beta)$-dictator-vs-no-influentials test $U$ using predicates from some CSP $\Psi$. Then for any $\delta \in (0, 1)$, we can construct a test $U'$ as in Ex. 7.24 such that the probability $U'$ accepts $f$ is equal to the probability that $U$ accepts $T_{1-\delta}(f)$. Assume $\epsilon$ is small enough so that $\epsilon < \delta$. Then $I_i[T_{1-\delta}f] \leq I_i^{(1-\epsilon)}[f]$ for all $i$. In particular, if $I_i^{(1-\epsilon)}[f] \leq \epsilon$, then $I_i[T_{1-\delta}f] \leq \epsilon$, and hence the probability that $U'$ accepts $f$ is at most $\alpha + \lambda(\epsilon)$. So, $U'$ is a $(\alpha, \beta - r\delta/2)$-dictator-vs-no-notables test, and hence it is UG-hard to $(\alpha + \delta', \beta - r\delta/2 - \delta')$-approximate MAX-CSP($\Psi$) for all $\delta' > 0$. Since $\delta > 0$ was arbitrary, it is UG-hard to $(\alpha + \epsilon, \beta - \epsilon)$-approximate MAX-CSP($\Psi$) for any $\epsilon > 0$, which is the conclusion of Khot's theorem. \hfill \Box

7.27 We can think of these constraints as a graph, with the permutation constraints labeling the edges. Suppose $G$ is a satisfiable instance of Unique-Games. For any label given to $i$, this uniquely determines the labels of all vertices in the same connected component, via a simple BFS-like procedure. If at any point we encounter a contradiction (i.e. two vertices are already labeled but they don’t satisfy their edge constraint), we restart with a different label for $i$. Eventually, we’ll find a satisfying labeling in time $O(qn^2)$. \hfill \Box

7.28 (a) Observe that if $x_1 + x_2 + x_3 = 0$, then all four of the clauses

\begin{equation}
(\neg x_1 \lor x_2 \lor x_3) (x_1 \lor \neg x_2 \lor x_3) (x_1 \lor x_2 \lor \neg x_3) (\neg x_1 \lor \neg x_2 \lor \neg x_3)
\end{equation}

(13)

are satisfied, while if the equation doesn’t hold, then exactly 3 clauses are satisfied. Similarly, if $x_1 + x_2 + x_3 = 1$, then we have a similar situation for the clauses

\begin{equation}
(x_1 \lor x_2 \lor x_3) (\neg x_1 \lor \neg x_2 \lor x_3) (x_1 \lor \neg x_2 \lor \neg x_3) (\neg x_1 \lor x_2 \lor \neg x_3)
\end{equation}

(14)

(b) Suppose we’re given an instance $\Psi$ of MAX-E3-Lin with value $v \geq 1 - \delta$, and we want to produce an assignment whose value is at least $1 + \delta$. Then using the clauses above, we transform $\Psi \mapsto \Psi'$, an instance of MAX-E3-Sat with 4 clauses, whose value is exactly $\frac{1}{4} + \frac{\delta}{2}$. Pick $\delta' > \delta$ such that $\frac{1}{4} + \frac{\delta}{2} \geq 1 - \delta'$. Since we have a $(7/8 + \delta', 1 - \delta')$-approximation algorithm for MAX-E3-Sat, we can use it on $\Psi'$ to get an assignment which satisfies at least a $7/8 + \delta'$-fraction of the clauses in $\Psi'$, which means it satisfies at least a fraction $4(7/8 + \delta') - 3 = 1/2 + 4\delta' \geq 1/2 + \delta$ of the equations in $\Psi$. So, having an efficient
(7/8 + \delta, 1 - \delta)-approximation algorithm for MAX-E3-Sat for every \delta > 0 implies there is an efficient (1/2 + \delta, 1 - \delta) approximation algorithm for MAX-E3-Lin for every \delta > 0.

(c) This is basically the same thing – when the original test wants to check a predicate of the form \( x_1 + x_2 + x_3 = b \), our new test will simply choose one of the four corresponding 3-SAT clauses at random and check that. If the original test accepts \( f \) with probability at most \( \alpha \), then the new test will accept \( f \) with probability at most \( \frac{\alpha}{4} + \frac{\delta}{4} \), while if the original test accepts \( f \) with probability at least \( \beta \), then the new test also accepts \( f \) with probability at least \( \beta \) (actually \( \frac{\beta}{4} + \frac{\delta}{4} \), which is \( \geq \beta \)).

7.29 (a) Note that \( \text{OXR}(x_1, x_2, x_3) = (x_1 \lor (x_2 \oplus x_3)) \) is true iff both of the clauses

\[
(x_1 \lor x_2 \lor x_3)(x_1 \lor \neg x_2 \lor \neg x_3)
\]  

(15)

are satisfied; conversely, any unsatisfying assignment for \( \text{OXR} \) will satisfy exactly 1 of the above clauses. So, if \( \Psi \) is a MAX-OXR instance with value \( v \), then the corresponding MAX-E3-Sat instance \( \Psi' \) has value \( \frac{v}{2} + \frac{1}{2} \). Then by the same reasoning in Ex. 7.28(c), this reduction converts \((\frac{3}{4} + \delta/4, 1)\)-Dictator-vs-No-Notables tests using \( \text{OXR}s \) to \((\frac{5}{8} + \delta/8, 1)\)-Dictator-vs-No-Notables tests using E3-Sat predicates.

(b) First we compute the Fourier expansion of \( \text{OXR}(x, y, z) \) to be

\[
-\frac{1}{2} + \frac{1}{2}x + \frac{1}{2}yz + \frac{1}{2}xyz
\]  

(16)

and so (using oddness of \( f \) and independence of \( x_i \) and \( y_i \)) the probability this test accepts \( f \) is

\[
\frac{3}{4} - \frac{1}{4}\mathbb{E}[f(y)f(z)] - \frac{1}{4}\mathbb{E}[f(x)f(y)f(z)]
\]  

(17)

where the expectation is of course taken over the distribution which, independently for each \( i \), chooses \( x_i, y_i \) independently and sets \( z_i = -y_i \) with probability \( \delta \) and \( z_i = -x_iy_i \) with probability \( 1 - \delta \). Notice that \( z \sim \mathcal{N}_{-\delta}(y) \), since for each \( i \), \( z_i \) and \( y_i \) are uniform with \( \mathbb{E}[y_iz_i] = -\delta \). Hence, the middle term above is \(-\frac{1}{4}\text{Stab}_{-\delta}(f)\). To evaluate the last term, we expand \( f \) as a Fourier series, and observe that if \( J \subseteq \{1, \ldots, n\} \) is the \((1 - \delta)\)-random subset of coordinates for which \( z_i = -x_iy_i \), we have

\[
\mathbb{E}[f(x)f(y)f(z)] = \mathbb{E}_{x,y,J}[\sum_{S \subseteq T \cup U} \hat{f}(S)\hat{T}(U)\hat{T}(U)(-1)^{|U|}x^Sx^Jx^Uy^Ty^U]
\]  

(18)

\[
= \mathbb{E}_J[\sum_S \hat{f}(S)^2\hat{T}(S \cap J)(-1)^{|T|}]
\]  

(19)

\[
= \sum_S (-1)^{|S|}\hat{f}(S)^2\mathbb{E}_{J \subseteq \{1, \ldots, n\}}[\hat{T}(J)]
\]  

(20)

We can clean up the \((-1)^{|S|}\) since \( \hat{f}(S) = 0 \) for \(|S|\) even, giving, finally, a probability

\[
\frac{3}{4} + \frac{1}{4}\text{Stab}_{\delta}(f) + \frac{1}{4}\sum_S \hat{f}(S)^2\mathbb{E}_{J \subseteq \{1, \ldots, n\}}[\hat{T}(J)]
\]  

(21)

of passing. For dictators, this is \( \frac{3}{4} + \frac{\delta}{4} + \frac{1-\delta}{4} = 1 \).
(c) We can use Cauchy Schwarz to upper bound
\[
\mathbb{E}_{J \subseteq 1-\delta} [\hat{f}(J)] = \sum_{J \subseteq S} (1 - \delta)^{|J|} \delta^{|S| - |J|} |\hat{f}(J)|
\]
\[
\leq \sqrt{\sum_{k=0}^{|S|} \binom{|S|}{k} (1 - \delta)^{2k} \delta^{2|S| - 2k} = (1 - 2\delta + 2\delta^2)^{|S|/2}}
\]
which for \( \delta \leq 1/2 \) is at most \((1 - \delta)^{|S|/2} \). Therefore
\[
\frac{1}{4} \sum_{S} \hat{f}(S)^2 \mathbb{E}_{J \subseteq 1-\delta} [\hat{f}(J)] \leq \frac{1}{4} \sum_{|S| \leq t} \hat{f}(S)^2 \mathbb{E}_{J \subseteq 1-\delta} [\hat{f}(J)] + \frac{1}{4} (1 - \delta)^{t/2}
\]
and since \( \text{Stab}(f) \leq \delta \) as \( f \) is odd, we get
\[
\Pr[f \text{ is accepted}] \leq \frac{3}{4} + \frac{\delta}{4} + \frac{1}{4} (1 - \delta)^{t/2} + \frac{1}{4} \sum_{|S| \leq t} \hat{f}(S)^2 \mathbb{E}_{J \subseteq 1-\delta} [\hat{f}(J)]
\]
(d) Suppose \( f \) (odd) has no \((\epsilon, \epsilon)\)-notable coordinates, meaning \( \text{Inf}_i^{(1-\epsilon)}(f) < \epsilon \) for all \( i \). Then for any \( t > 1 \):
\[
\frac{1}{4} \sum_{|S| \leq t} \hat{f}(S)^2 \mathbb{E}_{J \subseteq 1-\delta} [\hat{f}(J)] \leq (\max_{|S| \leq t} |\hat{f}(S)|^2)^{1/2} \leq \frac{\sqrt{\epsilon}}{(1 - \epsilon)^{t/2}}
\]
which implies that this is a \((\frac{3}{4} + \delta/4, 1)\)-Dictator-vs-No-Notables test.

7.30 (a) Let \( \ell : V \rightarrow [q] \) be the labelling which achieves the value \( \geq 1 - \delta \), and set \( f_v = \chi_{\ell(v)} \) for all \( v \in V \). By the regularity of \( G \), choosing \( u \) uniformly and then a random neighbor \( v \) is the same as choosing a uniformly random edge \((u, v)\) – so the probability that one of these \( r \) random edges lies in the \( \delta \)-fraction of unsatisfied edges is at most \( r\delta \). Otherwise, we have \( f_{\bar{v}_i}^\pi(x) = \chi_{\pi_i(\ell(v_i))}(x) = \chi_{\ell(u)}(x) \) for each \( i \). Since dictators pass with probability at least \( \beta \), we conclude \( \text{Opt}(\mathcal{P}) \geq \beta - r\delta \).

(b) Given \( F = (f_v)_{v \in V} \), set \( g_u(x) = \mathbb{E}_{v \sim \text{nbr}(u)}[f_{\bar{v}}(x)] \). Then
\[
\text{val}_F(F) = \mathbb{E}_{u,v_1,...,v_n \sim \text{nbr}(u), x}[\psi(f_{\bar{v}_1}^\pi(x^{(1)}), \ldots, f_{\bar{v}_r}^\pi(x^{(r)}))]
\]
\[
= \mathbb{E}_u[\mathbb{E}_{v_1,...,v_r \sim \text{nbr}(u), x}[\psi(f_{\bar{v}_1}^\pi(x^{(1)}), \ldots, f_{\bar{v}_r}^\pi(x^{(r)}))]]
\]
\[
= \mathbb{E}_u[\text{val}_F(g_u)]
\]
(c) If \( \text{val}_F(F) \geq \alpha + 2\lambda(\epsilon) \), then \( \mathbb{E}_u[\text{val}_F(g_u)] \geq \alpha + 2\lambda(\epsilon) \), and hence by a simple averaging argument, with probability at least \( \lambda(\epsilon) \), a random choice of \( u \) will have \( \text{val}_T(g_u) \geq \alpha + \lambda(\epsilon) \), and hence \( g_u \) has some \((\epsilon, \epsilon)\)-notable coordinates, that is, the set \( \text{NbrNotable}_u := \{i \in [q] : \text{Inf}_i^{(1-\epsilon)}g_u > \epsilon \} \) is non-empty.

(d) Since \( \text{Inf}_i^{(1-\epsilon)} \) is a sum of quadratic functions of linear functionals (i.e. Fourier coefficients), it is convex, and hence
\[
\mathbb{E}_{v \sim \text{nbr}(u)}[\text{Inf}_i^{(1-\epsilon)}f_v] = \mathbb{E}_{v \sim \text{nbr}(u)}[\text{Inf}_i^{(1-\epsilon)}f'_{\bar{v}}] \geq \text{Inf}_i^{(1-\epsilon)}[\mathbb{E}_{v \sim \text{nbr}(u)}[f_{\bar{v}}]] = \text{Inf}_i^{(1-\epsilon)}[g_u]
\]
(e) Set \( \text{Notable}_u := \{ i \in [q] : \inf (1-\epsilon)[f_u] \geq \epsilon/2 \} \). Let \( i \) have \( \inf (1-\epsilon)[g_u] > \epsilon \). Then by averaging, at least \( \epsilon/2 \) fraction of the neighbors \( v \) of \( u \) have \( \pi^{-1}(i) \in \text{Notable}_v \).

(f) Applying Proposition 2.54 (essentially Markov’s inequality with attenuation) to \( f_u \) and \( g_u \) (both of which have variance at most 1) gives \( |\text{Notable}_u \cup \text{Notable}_v| \leq 4/\epsilon^2 \).

(g) For at least a \( \lambda(\epsilon) \) fraction of \( u \), the set \( \text{NbrNotable}_u \) is non-empty, and hence for such \( u \), our assignment \( \ell(u) \in \text{NbrNotable}_u \) with probability \( \Omega(\epsilon^2) \) (by (f)). Then for at least a \( \epsilon/2 \)-fraction of \( v \in \text{nbr}(u) \), we have \( \pi^{-1}(\ell(u)) \in \text{Notable}_v \). Using (f) again, we see that \( v \) receives this compatible assignment with probability \( \Omega(\epsilon^2) \). Combining all these conditional probabilities, we see that the expected number of satisfied edge constraints is at least \( \lambda(\epsilon)\Omega(\epsilon^5) \).

(h) Assume the Unique Games Conjecture is true and that we can \( (\alpha+\delta, \beta-\delta) \)-distinguish \( \text{Max-CSP}(\Psi) \). Then pick \( \epsilon > 0 \) such that \( \lambda(\epsilon) < \delta/2 \) and \( \epsilon < (10r)^{-2}\delta \), and pick \( q \in \mathbb{N} \) such that \( \text{GAP-UGC}[q](10^{-3}e^5\lambda(\epsilon),1-10^{-3}e^5\lambda(\epsilon)) \) is NP-hard. I claim that our algorithm \( R \) gives an efficient algorithm for this gap problem: indeed, if \( \text{Opt}(G) < 10^{-3}e^5\lambda(\epsilon) \), then by the above analysis, \( \text{Opt}(R(G)) < \alpha + 2\lambda(\epsilon) < \alpha + \delta \). If \( \text{Opt}(G) \geq 1-\epsilon \), then \( \text{Opt}(R(G)) > \beta - \delta \) by (a). Hence, composing our reduction \( R \) with the \( (\alpha+\delta, \beta-\delta) \) distinguisher, we have solved an NP-hard gap problem, and hence \( \text{GAP-CSP}(\Psi) \) must be NP-hard. \( \square \)

2 Errata/Comments

- p. 186, Ex. 7.5: change \( T \) to \( T \)
- p. 187, Ex. 7.6: I think the hint suggests an unnecessarily ugly approach to this problem, see above for cleaner solution
- p. 188, Ex. 7.15(b): I believe the intended generalization was to \( \mathcal{H} = \{ \chi_\gamma : \gamma \in A \} \) where \( A \) is any affine subspace – this is what’s needed for Ex. 7.16 anyway – but this is not what is written
- p. 189, Ex. 7.18(c): replace “probability at least \( 1 - \gamma \cdot \epsilon \)” with “probability at least \( 1 - \lambda \epsilon \)”
- p. 193, Ex. 7.30(e): replace “\( \text{Notable}_u \)” with “\( \text{Notable}_v \)”
- p. 194, Ex. 7.31: replace “at most \( 1/m \)” by “at most \( r^2/M \), where \( r \) is the maximum arity of any constraint in \( \Psi \).”
- p. 226: replace “follow” by “follows”
- p. 230, Ex. 8.18: change \( \rho^k f^{=S} \) to \( \rho^{|S|} f^{=S} \)
- p. 230, Ex. 8.25(c): I’m not sure what the author was going for here, but there’s clearly something wrong with this inequality as written (the right side depends on \( S \) while the left doesn’t).
• p. 231, Ex. 8.26(a): I believe this exercise is incorrect as stated. To see where it goes wrong, note that a coordinate $i$ is $b$-pivotal for $f$ on $x$ iff it is $(-b)$-pivotal for $-f$ at $x$. Since $I_i(f) = I_i(-f)$, but $\pi_p(b) \neq \pi_p(-b)$ for $p \neq 1/2$, the formula given here cannot be correct.

• p. 232, Ex. 8.29(d): I suspect the $\Theta(\tau)\epsilon^{1+\tau}$ can be improved to $\Theta(\epsilon)$ with an equally simple argument – see my solution.

• p. 234, Ex. 8.35(a): The definition of the character $\chi_\alpha$ is missing its dependence on $\alpha$.

• p. 234, Ex. 8.36(d): This is incorrect – RDT False($\mathcal{T}$) should be replaced with RDT True($\mathcal{T}$).

• p. 234, Ex. 8.34(a): This is false as stated. A correct statement is: if $(x,y)$ is a $\rho$-correlated pair, then $(\omega(x),\omega(y))$ is a $1/2 + \rho^2$-correlated pair.

• p. 236, Ex. 8.45(b): Replace $\sqrt{\text{Var}[f]/n}$ with $\sqrt{n\text{Var}[f]}$.

• p. 236, Ex. 8.46(c) Replace $T$ with $\mathcal{T}$.

3 Chapter 8: Generalized Domains

8.1 Almost everything remains the same in this setting, once we suitably define a Fourier basis: if $\{\phi_i^{(j)} : i = 1, \ldots, m_j\}$ is a Fourier basis for $L^2(\Omega_j, \pi_j)$, then

$$\{\phi_\alpha(x) = \prod_{j=1}^n \phi_{\alpha_j}^{(j)} : \alpha \in [m_1] \times \cdots \times [m_n]\}$$

is a Fourier basis for $L^2(\prod_{j=1}^n \Omega_j, \pi_1 \otimes \cdots \otimes \pi_n)$. Since all of the results in these sections rely on nothing more than orthonormality, their proofs carry over to this more general setting.

8.2 Bilinearity of $\langle \cdot, \cdot \rangle : L^2(\Omega^n, \pi \otimes n) \times L^2(\Omega^n, \pi \otimes n) \to \mathbb{R}$ is clear, while positive definiteness follows from the fact that $\pi$ has full support.

8.3 By counting dimensions, we see that $\{\phi_\alpha\}$ is a basis for $L^2(\Omega^n, \pi \otimes n)$ (i.e. $\hat{f}(\alpha)$ exists and is unique), and by orthonormality, it must be that $\hat{f}(\alpha) = \langle f, \phi_\alpha \rangle$.

8.4 Straightforward verification.

8.5 Straightforward verification.

8.6 The first identity follows from applying Plancherel to $\langle f, 1 \rangle$; the second from applying it to $\langle f, f \rangle$, and the expressions for $\text{Var}[f]$ and $\text{Cov}[f,g]$ follow from these.

8.7 $E_I$ is obviously a linear projection, and it is self-adjoint since $E[fE_Ig] = \mathbb{E}[E_I f E_I g] = \mathbb{E}[gE_I f]$. Then since $T_\rho$ is a linear combination of the $E_I$, it is also self-adjoint.
8.8 Since $L_1 f := f - E_1 f$, we have
\[
\langle f, g \rangle = \langle E_1 f, E_1 g \rangle + \langle L_1 f, L_1 g \rangle + \langle L_1 f, E_1 g \rangle + \langle E_1 f, L_1 g \rangle
\] (28)

but
\[
\langle L_1 f, E_1 g \rangle = \mathbb{E}_{x \sim \pi}[(f(x) - E_x[f(x)])E_x[g(x)]]
\]
\[= \mathbb{E}_{x \sim \pi}[E_x[f(x)]g(x)] - \mathbb{E}_{x \sim \pi}[E_x[f(x)]E_x[g(x)]] = 0
\] (29)
and so $\langle f, g \rangle = \langle E_1 f, E_1 g \rangle + \langle L_1 f, L_1 g \rangle$. \qed

8.9 (a) Since $|f(x)| \in [0,1]$, we know $|E_1 f(x)| \in [0,1]$ and hence $L_1 f = f - E_1 f$ has range inside $[-2, 2]$. Thus, $|L_1 f(x)|^p \leq 2^p |L_1 f(x)|$, and so $\|L_1 f\|_p^p \leq 2^p \|L_1 f\|_1 = 2^p \inf_1(f)$.

(b) In general, Holder’s inequality says that whenever $\frac{1}{p} = \frac{1}{q} + \frac{1}{r}$, we have
\[
\|fg\|_r \leq \|f\|_q \|g\|_r
\] (31)

If $p \in (1, 2)$, then we can write $1/p = \theta/1 + (1 - \theta)/2$ for some $\theta \in (0,1)$, and then by applying Holder to $|f|^\theta$ and $|f|^{1-\theta}$ with $q = 1/\theta$ and $r = 2/(1-\theta)$:
\[
\|f\|_p^p = \|f\|_1^{p\theta} \|f\|_2^{p(1-\theta)}
\] (32)

Now replacing $f$ by $L_1 f$ and using $\|L_1 f\|_1 = \|L_1 f\|_2 = \inf_1(f)$, we conclude $\|L_1 f\|_p^p \leq \inf_1(f)$. \qed

8.10 (a) From it’s definition, we compute $\Pr_{y \sim N_\rho}(x)[y_i = \omega] = \rho \cdot 1_{x_i = \omega} + (1 - \rho)/m$.

(b) This defines a valid probability distribution as long as all of the probabilities lie in $[0,1]$, since the total mass is (by construction) independent of $\rho$. For this to be the case, we need only to make sure that $\rho \leq 1$ and $\rho \geq -\frac{1}{m-1}$, since then
\[
\rho + (1 - \rho)/m = \frac{1}{m} + \rho \frac{m - 1}{m} \geq 0.
\] (33)

(c) Since $\Pr_{x \sim \pi}[x_i = \omega] = 1/m$ for each $\omega \in \Omega$, plugging into (a) and averaging, we see that $y$ is uniformly distributed as well. It then follows from Bayes’ rule that $\Pr[x_i = \omega|y_i = \omega] = \Pr[y_i = \omega|x_i = \omega]$, and hence the distribution of $(x, y)$ is symmetric.

(d) When $\rho = -\frac{1}{m-1}$, we have equality in (33) and the result follows.

(e) For any $x \in \Omega^n$, $p_x(\rho) := \sum_{\alpha \in \mathbb{N}^n} \rho^{\# \alpha} \hat{f}(\alpha) \phi_\alpha(x)$ is a polynomial in $\rho$ which agrees with $T_\rho f(x)$ for all $\rho \in [0, 1]$. Moreover, from (a), we see that
\[
T_\rho f(x) = \mathbb{E}_{y \sim N_\rho(x)}[f(y)] = \sum_{y \in \Omega^n} f(y) \prod_i (\rho \cdot 1_{x_i = y_i} + (1 - \rho)/m)
\] (34)

is a polynomial for all $\rho \in [-\frac{1}{m-1}, 1]$. Hence, $p_x(\rho) = T_\rho f(x)$ for all $x \in \Omega^n$ and $\rho \in [-\frac{1}{m-1}, 1]$. \qed
8.11 Evidently, Inf_δ^{(b)}[f] is a continuous, monotone function of δ ∈ (0, 1]. Taking the limit as δ → 0^+, we obtain Inf_δ^{(0)}[f] = \sum_{\#\alpha = 1: \alpha \neq 0} \hat{f}(\alpha)^2. If Inf_δ^{(0)}[f] ≥ \epsilon, then also Inf_δ^{(1-\delta)}[f] ≥ \epsilon for every δ < 1. Hence, |J| ≤ \inf_{\delta < 1} \frac{1}{2\delta} = \frac{1}{2}, and so Prop 8.31 holds even when δ = 1. □

8.12 We have, by construction,

\[ \sum_{S \subseteq T} f = \sum_{S \subseteq T} \sum_{J \subseteq S} (-1)^{|S| - |J|} f^J \]  

(35)

Let’s examine the coefficient on \( f^J \), for a given subset \( J \subseteq T \). Firstly, if \( J = T \), then it occurs exactly once in the double sum, with coefficient 1; so we wish to show that everything else cancels out. For \( J \subsetneq T \), it appears with coefficient

\[ \sum_{T \supseteq S \supseteq J} (-1)^{|S| - |J|} = \sum_{U \subseteq T \setminus J} (-1)^{|U|} \]

(36)

= \sum_{k=0}^{\lfloor |T| - |J| \rfloor} \binom{|T| - |J|}{k} (-1)^k = (1 - 1)^{|T| - |J|} = 0. \]

(37)

\[ \square \]

8.13 Fix any product Fourier basis \( \{ \phi_\alpha : \alpha \in \mathbb{N}^n \} \) and consider the (linear) projection operator \( P_S \) onto the subspace of \( L^2 \) spanned by \( \{ \phi_\alpha : \text{supp}(\alpha) = S \} \). As in Prop 8.36, it follows from uniqueness that \( P_S(f) = f^S \), and hence the operator \( f \mapsto f^S \) is also a linear projection. □

8.14 If \( S \subseteq T \), then \( (f^S)^\subseteq T = \mathbb{E}_T[f = S] = f^S \), since \( f \) does not depend on any coordinates in \( T \). Now suppose \( S \cap T \neq S \). Then for any \( g \in L^2(\Omega^n, \pi \otimes m) \) which only depends on the coordinates in \( T \), we have on the one hand

\[ \langle f^S, g \rangle = \langle (f^S)^\subseteq T, g \rangle \]

(38)

but also

\[ \langle f^S, g \rangle = \langle f^S, g^{\subseteq S \cap T} \rangle = 0 \]

(39)

by the defining property (2) of \( f^S \). Thus, taking \( g = f^{\subseteq S \cap T} \), we conclude that \( f^S \subseteq T = 0 \) in this case. □

8.15 (a) We can write \( f^{\subseteq [t]} = f^{\subseteq [t-1]} + \sum_{S \subseteq [t-1]} f^{S \cup \{t\}} \). I claim that \( \mathbb{E}[f^{S \cup \{t\}} | \sigma_{t-1}] = 0 \), where \( \sigma_{t-1} \) is any \( \sigma \)-algebra inside of \( \sigma(x_1, \ldots, x_{t-1}) \). Indeed, whatever posterior distribution your information gives you about \( x_1, x_2, \ldots, x_{t-1} \), it must remain true that \( \Pr[x_t = \omega] = \pi(\omega) \), since these are independent. Hence, \( \mathbb{E}[\phi_t(x)g(x)|\sigma_{t-1}] = 0 \) for any \( g \) depending only on \( x_1, \ldots, x_{t-1} \). In particular,

\[ \mathbb{E}[f^{S \cup \{t\}}(x)|\sigma_{t-1}] = \sum_{\text{supp}(\alpha) = S \cup \{t\}} \hat{f}(\alpha)\mathbb{E}[\phi_t(x) \phi_{\alpha-t}(x)|\sigma_{t-1}] = 0 \]

(40)

which implies \( \mathbb{E}[f^{\subseteq [t]}(x)|f^{\subseteq [0]}(x), f^{\subseteq [1]}(x), \ldots, f^{\subseteq [t-1]}(x)] = f^{\subseteq [t-1]}(x) \).

(b) This was already done in (a). □
8.16 By parts 2) and 3) of Theorem 8.35, we know that $f^=S$ and $g^=T$ are orthogonal whenever $S \neq T$, since then either $S$ or $T$ must either be a proper subset of the other, or not a superset of it. This orthogonality gives us the identity

$$\langle f, g \rangle = \sum_{S \subseteq [n]} \langle f^=S, g^=S \rangle$$

(41)

Note that $E_i f^=S(x) = f^=S(x)$ if $i \notin S$, while $E_i f^=S(x) = 0$ otherwise, and so $L_i f^=S$ works the opposite way. It follows that

$$\text{Inf}_i[f] = \langle \sum_{S \subseteq [n]} f^=S, \sum_{S \supseteq i} f^=S \rangle = \sum_{S \supseteq i} \|f\|^2_2$$

(42)

and therefore

$$I[f] = \sum_{k=0}^n k \cdot W^k[f]$$

(43)

To see that $T_\rho(f^=S) = (T_\rho f)^=S$, observe that by linearity of $T$, $T_\rho f = \sum_S T_\rho(f^=S)$, and clearly $T_\rho(f^=S)$ only depends on coordinates in $S$, while if $T \subseteq S$ and $g$ is any function depending only on $T$, then self-adjointness gives

$$\langle T_\rho(f^=S), g \rangle = \langle f^=S, T_\rho g \rangle = 0$$

(44)

So, we conclude by the uniqueness part of Theorem 8.35 that $T_\rho(f^=S) = (T_\rho f)^=S$, which we can compute explicitly as

$$T_\rho(f^=S) = E_{J \sim \rho - \text{random subset}} [(f^=S) \subseteq J(x)] = \text{Pr}[S \subset J] f^=S(x) = \rho^{|S|} f^=S(x)$$

(45)

The formula for noise stability follows from this and orthogonality.

8.17 $\|f^=S\|_\infty \leq \sum_{J \subseteq S} \|f^=J\|_\infty \leq 2^{|S|} \|f\|_\infty$ (since $f^=J(x) = E_{\mathcal{F}}[f(x)] \implies \|f^=J\|_\infty \leq \|f\|_\infty$)

8.18 This is trivial to check, once the Fourier expansion has been computed, as in Ex. 5.

8.19 Fix any Fourier basis. For any set $S \subseteq [n]$ of size $s \geq 1$, we want to show

$$\sum_{\text{supp}(\alpha) \subseteq S} \hat{f}(\alpha)^2 \leq \frac{s}{s+1} \sum_{\text{supp}(\alpha) \subseteq S \cup \{i\}} \hat{f}(\alpha)^2$$

(46)

By symmetry, we have

$$\sum_{\text{supp}(\alpha) \subseteq S; \#\alpha = k} \hat{f}(\alpha)^2 = \binom{s}{k} \sum_{\text{supp}(\alpha) \subseteq S \cup \{i\}; \#\alpha = k} \hat{f}(\alpha)^2$$

$$\leq \frac{s}{s+1} \sum_{\text{supp}(\alpha) \subseteq S \cup \{i\}; \#\alpha = k} \hat{f}(\alpha)^2$$

(48)

and summing over $k$ gives the desired result.
8.20 Setting $X = \sum_{i=1}^n x_i$, we have $\mu = \mathbb{E}[X] = pn$. For $p = n/2 - C\sqrt{n}$, and $\delta = \frac{(2+o(1))C}{\sqrt{n}}$, the Chernoff bound says

$$\Pr[X > n/2] \leq P[X > (1+\delta)\mu] \leq e^{-\frac{\delta^2\mu}{2}} \leq e^{-\frac{C^2}{2} + o(1)}$$ (49)

and similarly, when $p = n/2 + C\sqrt{n}$,

$$\Pr[X < n/2] \leq e^{-\frac{C^2}{2} + o(1)}$$ (50)

and so it suffices to take $C = 2\sqrt{\ln(1/\epsilon)}$.

8.21 Just define $D_i f(x) = \sigma_i \cdot \frac{f(x^{(i+1)}) - f(x^{(i-1)})}{2}$, and the same arguments from Prop. 8.45 show

$$\inf_i [f] = \sum_{i=1}^n \sigma_i^2 \Pr[f(x) \neq f(x^{(i)})]$$ (51)

$$\inf_i [f] = \sigma_i \hat{f}(i) \quad (f \text{ monotone})$$ (52)

$$I[f] = \sum_{i=1}^n \sigma_i^2 \Pr[f(x) \neq f(x^{(i)})]$$ (53)

8.22 (a) This follows from using $D_{\phi_i} = \sigma_i D_{x_i}$ repeatedly.

(b) Since $f^{(p)}(S) = \mathbb{E}[D_{\phi_S} f]$, we can use part (a) to conclude

$$f^{(p)}(S) = \mathbb{E}[D_{\phi_S} f] = \prod_{i \in S} \sigma_i \cdot D_{x_i} f(\mu_1, \ldots, \mu_n)$$ (54)

(c) From the definition, we see $\|D_{x_i} f\|_\infty \leq \|f\|_\infty$, and hence by induction, $\|D_{x_S} f\|_\infty \leq \|f\|_\infty$. It follows that $|f^{(p)}(S)| \leq \prod_{i \in S} \sigma_i \|f\|_\infty$.

8.23 Clearly, $F(p) := \Pr_{x \sim \pi_p}[f(x) = \text{True}]$ is a continuous function which satisfies $F(0) = 0$ and $F(1) = 1$. Thus for $\alpha \in (0,1)$, there exists by the intermediate value theorem a value $p = p(\alpha) \in (0,1)$ such that $F(p) = \alpha$. Moreover, by the Margulis-Russo formula, $F$ is strictly increasing, and hence $p(\alpha)$ is unique.

8.24 WLOG suppose $p_c(n) \leq 1/2$ for all $n$. Since $(f_n)$ has a coarse threshold, there exists $c > 0$ and increasing sequence $\{n_i\}_{i \in \mathbb{N}}$ such that $\delta(n_i) > c \cdot p_c(n_i)$. If we let $F_i(p) := \Pr_{x \sim \pi_p}[f_{n_i}(x) = \text{True}]$, then by Ex. 8.27, $F_i^{-1}$ exists, and the above condition becomes

$$\frac{F_i^{-1}(1-\epsilon) - F_i^{-1}(\epsilon)}{1 - 2\epsilon} \geq (1 - 2\epsilon)^{-1} c \cdot p_c(n_i)$$ (55)

Moreover, by the mean value theorem, there is some $x_i \in (\epsilon, 1-\epsilon)$ with

$$F^{-1}(x_i) = \frac{F_i^{-1}(1-\epsilon) - F_i^{-1}(\epsilon)}{1 - 2\epsilon} \geq (1 - 2\epsilon)^{-1} c \cdot p_c(n_i)$$ (56)
Letting \( p_i = F^{-1}(x_i) \) and using Margulis-Russo, we obtain

\[
\frac{\sigma^2(p_i)}{I(f_{n_i}^{(p_i)})} = (F^{-1}(x_i))' \geq (1 - 2\epsilon)^{-1}c \cdot p_c(n_i)
\]

and hence

\[
I(f_{n_i}^{(p_i)}) \leq \frac{(1 - 2\epsilon)c^2(p_i)}{c \cdot p_c(n_i)}
\]

We want to conclude that the RHS is bounded by a constant independent of \( n \) – indeed, for this we use Ex. 8.25. More specifically, by extracting another subsequence if necessary, we can assume our \( p_i \) always lie on the same side of \( p_c(n_i) \) – WLOG, suppose \( p_i \leq p_c(n_i) \) for all \( i \). Then since \( F(p_i) > \epsilon \), we know by 8.25(c) that \( p_i \geq (2\epsilon)^2p_c \), and hence the RHS of (58) is bounded above by a constant depending only on \( \epsilon \).

8.25 (a) By Margulis-Russo, \( F'(p) = \frac{I(f^{(p)})}{4p(1-p)} \geq \frac{\text{Var}(f)}{4p(1-p)} = \frac{F(p)(1-F(p))}{p(1-p)}. \)

(b) For \( p \leq p_c \), we have both \( 1 - F(p) \geq 1/2 \) and \( p(1-p) \leq p \), hence from (a) we get \( F'(p) \geq \frac{F(p)}{2p} \), and therefore \( (\ln F)'(p) = \frac{F'(p)}{F(p)} \geq \frac{1}{2p} \).

(c) For \( p_0 \leq p_c \), we have

\[
- \ln(2F(p_0)) = \ln(F(p_c)) - \ln(F(p_0)) = \int_{p_0}^{p_c} (\ln F)'(p) \, dp \geq \int_{p_0}^{p_c} \frac{dp}{2p} = \ln\left(\frac{\sqrt{p_c/p_0}}{2\sqrt{p_c}}\right)
\]

and so \( F(p_0) \leq \frac{1}{2}\sqrt{p_0/p_c} \). In particular, if \( p_0 \leq (2\epsilon)^2p_c \), then \( F(p_0) \leq \epsilon \).

(d) Using (c), we know that for \( p \leq p_c \), we have

\[
\frac{F'(p)}{F(p)} \geq \frac{1 - \sqrt{p/p_c}}{p}
\]

and hence, using the FTC as before, we get

\[
- \ln(2F(p_0)) \geq \ln(p_c/p_0) + \sqrt{p_0/p_c} - 2 \implies F(p_0) \leq \frac{p_0}{2p_c}e^{2-\sqrt{p_0/p_c}}
\]

and hence for \( p_0 \leq \Theta(\epsilon) \), we have \( F(p_0) \leq \epsilon \).

(e) Applying the same kind of analysis to \( \ln(1 - F(p)) \), we obtain, for \( p \in [p_c, 1/2] \),

\[
F(p) \geq 1 - \frac{1}{2}\sqrt{p_c/p}
\]

and so if \( p_1 \leq 1/2 \) has \( p_c/p_1 \leq (2\epsilon)^2 \), we know \( F(p_1) \geq 1 - \epsilon \). In any case, we can still set \( p = 1/2 \) and obtain \( F(1/2) \geq 1 - \sqrt{p_c/2} \).

(f) I’m not sure what \( \delta \) is in this problem, but the problem’s claim certainly can’t be true unconditionally, since taking \( \delta \) very close to 1 clearly gives a contradiction. \( \Box \)
8.26 a) Let $x' = (x_3, \ldots, x_n)$. Letting $F(p) = \Pr_{x' \sim \nu_p}[\text{MAJ}_{n-2}(x') = \text{True}]$, we can compute

$$\Pr_{\nu_p}[f = \text{True}] = F(p)(1 - (1 - p)^2) + (1 - F(p))p^2 = p^2 + 2F(p)(p - p^2) \quad (63)$$

which yields $p_c = 1/2$ (since $F(1/2) = 1/2$). Monotonicity is obvious, as $f$ is the composition of monotone functions.

b) For large $n$, we know that $F(p)$ looks essentially like the step function $1_{p > 1/2}$, and so our probability curve looks a lot like the graph of

$$p^2 \quad \text{if } p \leq 1/2$$

$$2p - p^2 \quad \text{if } p > 1/2$$

c) Since $f$ is monotone, it suffices to compute the sum of its level 1 Fourier coefficients. Using the Fourier expansion of $\text{MAJ}_3$, this is easily seen to be $1 + \frac{1}{2} \text{Inf}(\text{MAJ}_{n-2}) = \Theta(\sqrt{n})$.

d) By our reasoning in b), we know that for $\epsilon < 1/4$, $p_1(n) - p_0(n) = 1/2 - (\sqrt{\epsilon} + o(1)) \neq 0$, and thus $f_n$ has a coarse threshold.

\[\hfill \square\hfill\]

8.27 a) I claim that the distribution (ii) assigns equal probability to all strings of weight $k$ – this is obvious from symmetry and the uniform choice of $\pi$. Also, since $\pi^{\leq k}$ always has weight $k$, and $k$ is chosen uniformly, we see that (ii) matches (i). Now consider choosing $n + 1$ points (which we’ll call $p, y_1, y_2, \ldots, y_n$) uniformly and independently in $[0, 1]$, and define the string $x \in \{0, 1\}^n$ by

$$x_i = 1 \iff y_i \geq p$$

Conditioned on $p$, the distribution of $x$ is visibly $\pi_p^{\otimes n}$, and since $p$ is uniform in $[0, 1]$, we see that $x$ is distributed according to (iii). However, if we fix a string $x$ with weight $k$, then the probability that $x = x$ is that for the $k$ indices $i$ in which $x_i = 1$, we have $y_i \geq p$, and for all other indices, $y_j < p$. Since any ordering of $p, y_1, \ldots, y_n$ is equally likely, this occurs with probability $\frac{1}{n+1} \cdot \frac{1}{(k)}$ (since $1/(n+1)$ of the orderings have $p$ as the $k + 1$st largest element, and $1/(k)$ of those have the right $k$ indices for the $k$ largest elements.) Thus, (i) (ii) and (iii) all define the same distribution.

b) To see this, it is perhaps best to use the characterization (iii): indeed this is just $\pi_p^f$, for $p \sim [0, 1]$ uniform, which is exactly $\nu^n$.  

c) Let $y$ have weight $1 \leq k \leq n - 1$. Then $f(y)$ appears in the sum $\sum_{i=1} \sum f(x^{(i \to 0)})$ with coefficient

$$\begin{cases} 
1 & \text{if } x \text{ is one of the } k \text{ vectors obtained by zeroing a non-zero coordinate of } y \\
2k - n & \text{if } x = y \\
-1 & \text{if } x \text{ is one of the } n - k \text{ vectors obtained by adding 1 to a zero coordinate of } y 
\end{cases}$$

and thus, using the (i) characterization of $\nu^n$, we conclude $f(y)$ appears in $\sum_{i=1}^n \text{Shap}_i(f)$ with coefficient

$$\frac{1}{n+1} \left( \binom{k}{n-1} + \frac{2k-n}{\binom{n}{k}} - \frac{n-k}{\binom{n}{k+1}} \right) = 0$$
8.29 a) Let \( \phi \) be a product Fourier basis for \( L^2(\Omega^n, \mathbb{R}, \pi) \), and for \( f : \Omega^n \rightarrow V \), define its Fourier coefficients via
\[
\hat{f}(\alpha) = \mathbb{E}_{x \sim \pi^n}[\phi_\alpha(x)f(x)] \in V
\] (66)
Then if we define an inner product \( \langle f, g \rangle := \mathbb{E}_{x \sim \pi^n}[(f(x), g(x))_V] \), we can, as usual, plug in the fourier expansion \( f(x) = \sum_\alpha \phi_\alpha(x)\hat{f}(\alpha) \) and use orthogonality of the \( \phi \) basis to deduce Plancherel’s theorem:
\[
\langle f, g \rangle = \mathbb{E}_{x \sim \pi^n}[(f(x), g(x))_V] = \mathbb{E}_{x \sim \pi^n}\left[\sum_{\alpha, \beta} \phi_\alpha(x)\phi_\beta(x)\langle \hat{f}(\alpha), \hat{g}(\beta) \rangle_V\right]
\] (67)
\[
= \sum_\alpha \langle \hat{f}(\alpha), \hat{g}(\alpha) \rangle_V
\] (68)
The other statements in 8.16 can essentially be obtained from this one.

b) Stab\(\rho\)(\(f\)) = \(\mathbb{E}_{x, y \sim \pi^n}[(f(x), f(y))_V] = \Pr_{x \sim \pi^n}[f(x) = f(y)]\), since the inner product of two coordinate vectors is 1 if they are equal and zero otherwise.

8.30 (a) I believe this problem is incorrect as stated. Consider the case that \( \rho = 0 \) and \( \Omega = \{-1, 1\} \) with the uniform measure. According to this problem, if we first choose \( \omega^{(+1)}, \omega^{(-1)}, x, y \in \{-1, 1\}^n \) independently and uniformly, then the strings \((\omega^{(x)}, \omega^{(y)})\) should be independent and uniform. However, there is clearly a bias in favor of \( \omega^{(x)} = \omega^{(y)} \) (which happens with probability 1/2), then we are guaranteed to have \( \omega^{(x)} = \omega^{(y)} \), and otherwise it happens with probability 1/2; for a total probability of 3/4. In fact, this problem can be corrected by incorporating this correlation boosting:

**Claim:** Under the conditions of Ex. 8.30(a), \((\omega^{(x)}, \omega^{(y)})\) is a \((\frac{1+\rho}{2})\)-correlated pair.

**Proof:** Clearly \( \omega^{(x)} \) is \( \pi^{\otimes n} \) distributed, and the coordinates are independent. Moreover, conditioning on \( \omega^{(x)} \), we have
\[
\Pr[\omega^{(y)} = \omega | \omega^{(x)}] = 1_{\omega = \omega^{(x)}} \Pr[y_i = x_i] + \Pr[y_i \neq x_i] \Pr[\omega^{(y)} = \omega]
\] (69)
\[
= 1_{\omega = \omega^{(x)}} \left(\frac{1+\rho}{2}\right) + \left(\frac{1-\rho}{2}\right) \pi(\omega)
\] (70)
This is evidently the marginal distribution of a \( \left( \frac{1+\rho}{2} \right) \)-correlated pair.

(b) Let \( f(\omega) = \text{sgn}(\ell(\omega)) \in L^2(\Omega^n, \pi^n) \) where \( \ell(\omega) \) has degree \( \leq 1 \), and for \( \omega^{(+1)}, \omega^{(-1)} \in \Omega^n \), set \( g(x) = g_{\omega^{(+1)}}(\omega^{(-1)})(x) := f(\omega(x)) \). We can write \( \ell(\omega) = \alpha_0 + \sum_{i=1}^n \alpha_i(\omega_i) \) for some constant \( \alpha_0 \) and functions \( \alpha_i : \Omega \to \mathbb{R} \). Then

\[
g(x) = \text{sgn}(\alpha_0 + \sum_{i=1}^n \alpha_i(\omega_i(x)))
\]

which is clearly a linear threshold function in the usual sense.

(c) Using part (a) and the notation of (b), we find

\[
\text{Stab}_\rho[f] = \mathbb{E}_{\rho \text{- correlated}} \left[ f(\omega)f(\omega') \right]
\]

\[
= \mathbb{E}_{\omega^{(+1)}, \omega^{(-1)} \sim \pi^n \oplus \pi^n} \mathbb{E}_{2\rho-1 \text{- correlated}} \left[ g_{\omega^{(+1)}}(\omega^{(-1)})(x)g_{\omega^{(+1)}}(\omega^{(-1)})(y) \right]
\]

\[
= \mathbb{E}_{\omega^{(+1)}, \omega^{(-1)} \sim \pi^n} \left[ \text{Stab}_{2\rho-1}[g_{\omega^{(+1)}}(\omega^{(-1)})] \right]
\]

Using the definition \( \text{NS}_\delta = \frac{1}{2} - \frac{1}{2}\text{Stab}_{1-2\delta} \), we conclude

\[
\text{NS}_\delta[f] = \mathbb{E}_{\omega^{(+1)}, \omega^{(-1)} \sim \pi^n} \left[ \text{NS}_{2\delta}[g_{\omega^{(+1)}}(\omega^{(-1)})] \right]
\]

and since \( g_{\omega^{(+1)}}(\omega^{(-1)}) \) is a genuine LTF, we conclude

\[
\text{NS}_\delta[f] \leq \max_{\omega^{(+1)}, \omega^{(-1)}} \text{NS}_{2\delta}[g_{\omega^{(+1)}}(\omega^{(-1)})] \leq C \sqrt{2\delta}
\]

and hence Peres’s theorem holds with a constant at most \( \sqrt{2} \)-times bigger than the constant obtained in the hypercube case.

8.31 (a) Given \( \alpha \in G \), define \( \chi_\alpha(x) = e^{2\pi i(\sum_{j=1}^m \alpha_j x_j/m_j)} \), which is obviously a character of \( G = \mathbb{Z}_m \times \cdots \times \mathbb{Z}_m \). If \( \alpha_i \neq \alpha_i' \mod m_i \), then \( \chi_\alpha(0, \cdots, 1, 0, \cdots) = e^{2\pi i\alpha_i/m_i} \neq e^{2\pi i\alpha_i'/m_i} \), and hence \( \chi_\alpha \) and \( \chi_{\alpha'} \) are distinct characters. Hence they are orthogonal, and since \( \dim L^2(G) = |G| \), these \( \chi_\alpha \) form a Fourier basis for \( L^2(G) \).

(b) I claim the map \( \alpha \mapsto \chi_\alpha \) is group homomorphism. Indeed,

\[
\chi_{\alpha + \beta}(x) = e^{2\pi i(\sum_{j=1}^m (\alpha_j + \beta_j) x_j/m_j)} = e^{2\pi i(\sum_{j=1}^m \alpha_j x_j/m_j)}e^{2\pi i(\sum_{j=1}^m \beta_j x_j/m_j)} = \chi_{\alpha}(x)\chi_{\beta}(x)
\]

Moreover, by (a), it is injective, and hence an isomorphism. Thus, \( G \cong \hat{G} \).

8.32 Since \( (f * g)(x) = \mathbb{E}_{y \sim G}[f(y)g(x-y)] \), and the measure on \( G \) is translation invariant, we can instead consider \( z = x - y \sim G \) and write \( f * g(x) = \mathbb{E}_{z \sim G}[f(x-z)g(z)] = g * f(x) \). Associativity essentially follows from switching the order of summation and using the
fact that if $T_y$ is the “translation by $y$” operator, then $(T_y f) * (T_{-y} g)(x) = f * g(x)$. Explicitly,

$$
(f * (g * h))(x) = \mathbb{E}_{y \sim G}[f(y)\mathbb{E}_{z \sim G}[g(z)h(x - y - z)]]
$$

$$
= \mathbb{E}_{y \sim G}[f(y)\mathbb{E}_{z \sim G}[g(z - y)h(x - z)]]
$$

$$
= \mathbb{E}_{z \sim G}\mathbb{E}_{y \sim G}[(f(y)g(z - y))h(x - z)]
$$

$$
= \mathbb{E}_{z \sim G}[(f * g)(z)h(x - z)]
$$

$$
= ((f * g) * h)(x)
$$

Since $(f, g) \mapsto \hat{f} \hat{g}$ and $(f, g) \mapsto \hat{f} \hat{g}$ are both bilinear operators, it suffices to show that they agree on a basis – namely the characters $\chi \in \hat{G}$. If $f = \chi$ and $g = \chi'$, then the corresponding Fourier transforms are point-masses at $\chi$ and $\chi'$ respectively, and so $\hat{f}(\alpha)\hat{g}(\alpha) = 1_{\alpha = \chi = \chi'}$. On the other hand, $(\chi \chi')(x) = \mathbb{E}_g[\chi(y)\chi'(x - y)] = \chi'(x)\langle \chi, \chi' \rangle = \chi(x)1_{\chi = \chi'}$, and so it’s Fourier transform is either zero (if $\chi \neq \chi'$) or a point mass at $\chi$ if $\chi = \chi'$, which is of course the same thing. \hfill \Box

8.33 (a) Let $G = \text{Aut}(f) \leq S_n$. Then $G$ induces a faithful action on decision trees computing $f$; given such a tree $T$, let $T'$ be the randomized decision tree formed by the orbit $\{T^\pi\}_{\pi \in G}$ of $T$ under $G$ with the uniform distribution. Then

$$
\delta_i(T') = \frac{1}{|G|} \sum_{\pi \in G} \delta_{\pi(i)}(T)
$$

(76)

So, if we can show $|\{\pi \in G : \pi(i) = j\}|$ is the same for all $(i, j)$, it will follow that $\delta_i(T')$ is independent of $i$. Indeed, for any $i, j, k$, we can find $\pi_0 \in G$ such that $\pi_0(j) = k$, and hence $\pi \mapsto \pi_0 \pi$ gives a bijection between $\{\pi \in G : \pi(i) = j\}$ and $\{\pi \in G : \pi(i) = k\}$, so each set has size $|G|/n$, and therefore

$$
\delta_i(T') = \frac{1}{n} \sum_{j=1}^{n} \delta_j(T)
$$

(77)

for all $i$.

(b) By remark 8.63, we can always find a deterministic decision tree $T$ computing $f$ which attains $\Delta(T) = \Delta(f)$. Then by (a), we can find a randomized decision tree $T'$ with $\delta(T') = \frac{1}{n} \Delta(f)$. To see that this is optimal, note that for any (randomized) tree $T$, we have $\delta(T) \geq \frac{1}{n} \Delta(T) \geq \frac{1}{n} \Delta(f)$.

\hfill \Box

8.34 (a) To prove that $\text{DT(MAJ}_{3}^{\otimes d}) = 3^d$, it suffices to show that for any sequence of $3^d - 1$ (adaptive) queries to the input, there is always some sequence of answers which will leave the function undetermined. Indeed, we can arrange things so that at each “layer,” there is exactly one unknown bit, while the other two bits are unequal. This way, if the final un-queried bit is a 1, the whole function will evaluate to 1, while if it is a zero, the function will be 0.

To see that $\text{RD}\text{T(MAJ}_{3}^{\otimes d}) \leq (8/3)^d$, we use induction on $d$. The base case $d = 1$ is done as an example on page 222. For the inductive step, break the input $x$ into 3 chunks $x^1, x^2, x^3$ so that $\text{MAJ}_{3}(x) = \text{MAJ}_{3}(\text{MAJ}_{3}^{\otimes d-1}(x^1), \text{MAJ}_{3}^{\otimes d-1}(x^2), \text{MAJ}_{3}^{\otimes d-1}(x^3))$, and
suppose $T_1, T_2, T_3$ are randomized decision trees computing $\text{MAJ}^{\otimes d-1}_3$ with worst-case cost $(8/3)^{d-1}$. Then define the randomized tree $T$ as follows: choose 2 distinct indices $i, j \in \{1, 2, 3\}$ uniformly at random and then make the queries $T_i$ wants to make, followed by all the queries $T_j$ wants to make. If the value is still undecided, make the queries the remaining tree $T_k$ wants to make. With probability at least $1/3$, we don’t need to know the information from $T_k$, in which case we use an expected $2(8/3)^{d-1}$ queries; otherwise, with probability $2/3$, we need an expected $3(8/3)^{d-1}$ queries. Adding gives $(8/3)^d$.

A nearly identical argument shows $\Delta(\text{MAJ}^{\otimes d}_3) \leq (5/2)^d$.

(b) Consider the following randomized protocol (which can easily be written down in the form of a randomized decision tree). Writing $x = (x^1, x^2, x^3)$, as before, where each $x^i$ consists of 3 bits, we randomly pick 2 distinct indices $i, j \in \{1, 2, 3\}$ and then randomly sample 2 distinct bits from each of $x^i$ and $x^j$. If $i, j$ correspond to 2 majorities which agree, we call this “getting lucky in the first layer,” and otherwise we are “unlucky in the first layer.” Similarly, if the bits queried within $x^i$ agree, that is lucky in the second layer; so far, we can represent the luck pattern in our three choices by triples LLL, LUU, LUL, ULL, UUU, ULU, UUL. The probabilities of each of these events are computed by multiplying $(2/3)$ for each U and $(1/3)$ for each L. The next step in the protocol must be the same within information sets, which are \{LLL, \{LUU,UUU\}, \{ULL\}, \{LLU,ULU\}, \{LUL,UUL\}. In the first case we are done with a total of 4 queries; in the second case we read the remaining 2 bits from $x^i$ and $x^j$, and then, if necessary, query 2 bits at random from $x^k$ and then the third bit if necessary, for an expected total of either 6 or $6 + \frac{8}{3}$ bits; in the third case, we query 2 bits at random from $x^k$ and then the third if necessary, for an expected total of $4 + \frac{8}{3}$ bits; in the fourth and fifth cases we read the third bit from whichever of $x^i$ or $x^j$ is still undetermined, and then sample 2 random bits from $x^k$ if necessary, for a total of either 5 or $5 + \frac{8}{3}$ bits. Scaling by the appropriate probabilities and adding, we get $\frac{187}{27} = 6.8888 < (8/3)^2$. \(\Box\)

8.35 (a) Let $T$ be deterministic decision tree computing $\text{OR}_n$. No matter which sequence of bits $T$ chooses to query, it must be the case that it only outputs an answer when either a 1 has been found, or all bits have been queried. The probability that $T$ queries exactly $k$ bits is therefore $p(1 - p)^{k-1}$, unless $k = n$, in which case it is just $(1 - p)^{n-1}$. Hence $\Delta(p)(T) = n(1 - p)^{n-1} + p \sum_{k=1}^{n-1} k(1 - p)^{k-1}$. We can also compute this expectation as the geometric series $\sum_k p \text{Pr}[^{-\infty} \geq k] = \sum_{k=1}^{n} (1 - p)^{k-1} = 1 - (1 - p)^{n}$. Since $\text{Pr}_p[\text{OR}_n(x) = 1] = 1 - (1 - p)^{n}$, we can compute $1/2 = (1 - p_c)^{n}$, so that $p_c = 1 - \frac{1}{2^{1/n}}$. For large $n$, $\frac{1}{2^{1/n}} = e^{-\frac{\ln 2}{n}} \approx 1 - \frac{\ln 2}{n}$, and hence $p_c \approx \frac{\ln 2}{n}$, and thus $\Delta(p_c)(\text{OR}_n) \approx \frac{n}{2\ln 2}$. \(\Box\)

8.36 (a) When $d = 2$, $\text{NAND}^{\otimes 2}(x_1, x_2, x_3, x_4)$ is 1 iff at least one of the layer 1 NAND’s is zero, i.e. $(x_1 \land x_2) \lor (x_3 \land x_4)$. The result then follows for general $d$ by induction, using essentially the same argument.

(b) We prove the following claim by induction on $d$: If $T$ is a deterministic decision tree computing $\text{NAND}^{\otimes d}$, then $T$ must make $2^d$ queries on some input. When $d = 1$, this is clear: if whatever bit queried first is true, the second bit MUST be queried. Now suppose $\text{DT}(\text{NAND}^{\otimes d}) = 2^d$, and suppose $T$ computes $\text{NAND}^{\otimes (d+1)}(x) = \text{NAND}(\text{NAND}^{\otimes d}(x^1), \text{NAND}^{\otimes d}(x^2))$. For any $x^1$ which cases $\text{NAND}^{\otimes d}(x^1)$ to be TRUE, there must be some
$x^2$ for which $T$ makes $2^d$ queries to it. Indeed, if not, we could then hardcode $x^1$ into $T$ and get a better decision tree for $\mathrm{NAND}^{\otimes d}$. Moreover, we can pick $x^2$ such that $\mathrm{NAND}^{\otimes d}(x^2)$ is TRUE, since if all the bits are needed, toggling the last queried bit of $x^2$ must change the function value. Then we can swap the roles of $x^1$ and $x^2$ to get a pair for which both internal $d$-recursive NANDs must query all $2^d$ of their bits, meaning $T$ requires $2^{d+1}$ queries on some input, as desired.

(c) On the input $(x_1, x_2) = (1, 1)$, we must read both bits to conclude the value of NAND, so $\text{RDT}(\text{NAND}) = 2$.

(d) Since TRUE inputs have at least one false coordinate, we can query a coordinate at random and with probability $1/2$ (in the worst case), we’ll be done, and with probability $1/2$, we’ll need an extra bit, giving $\text{RDT}_{\text{True}}(T) = 3/2$.

(e) This follows from (c), (d) and the inequalities in (f) below.

(f) We build up the family $(\mathcal{T}_d)_{d \in \mathbb{N}}$ inductively. On input $x = (x^1, x^2)$, $\mathcal{T}_d$ will pick either $x^1$ or $x^2$ at random and run $\mathcal{T}_{d-1}$ on it; if the value is still undetermined, it will run $\mathcal{T}_{d-1}$ on the other $x^j$. If $x$ is a FALSE input, then $\mathcal{T}_{d-1}$ must output TRUE for both $x^1$ and $x^2$, and so $\text{RDT}_{\text{False}}(\mathcal{T}_d) \leq 2\text{RDT}_{\text{True}}(\mathcal{T}_{d-1})$. If $x$ is a TRUE input, then one $x^1, x^2$ must evaluate to false – with probability at least $1/2$, we select this $x^j$ and are done after running $\mathcal{T}_{d-1}$ on it; with probability at most $1/2$, we need to do an additional run of $\mathcal{T}_{d-1}$ on a true input. Hence, $\text{RDT}_{\text{True}}(\mathcal{T}_d) \leq \text{RDT}_{\text{False}}(\mathcal{T}_{d-1}) + \frac{1}{2}\text{RDT}_{\text{True}}(\mathcal{T}_{d-1})$.

(g) Using the two recursive inequalities, setting $T_d = \text{RDT}_{\text{True}}(\mathcal{T}_{d-1})$, we see

$$T_d \leq \frac{1}{2}T_{d-1} + 2T_{d-2} \tag{78}$$

and hence $T_d \leq c_1 \lambda_+^d + c_2 \lambda_-^d$, where $\lambda_{\pm} = \frac{1+\sqrt{33}}{4}$. A patient person could solve for $c_\pm$ exactly using the initial conditions from parts (d) and (e), but without doing that, one can already deduce (since $|\lambda_-| < |\lambda_+|$) the asymptotic behavior $\text{RDT}(\text{NAND}^{\otimes d}) = O\left((\frac{1+\sqrt{33}}{4})^d \right) = O(n^{-754})$. □

8.37 By the OS inequality, we know $I[f] \leq O(\sqrt{\Delta(f)}) \leq O(\sqrt{\text{DT}(f)}) \leq O(\sqrt{k})$, and hence $f$ is $\epsilon$-concentrated up to degree $O(\sqrt{k}/\epsilon)$, and therefore on a collection $\mathcal{F}$ of size $n^{O(\sqrt{k}/\epsilon)}$, and so we can apply the usual technique: estimate each of the Fourier coefficients with $\text{poly}(|\mathcal{F}|, n, 1/\epsilon)$ samples and output the sign of the resulting multilinear polynomial. By standard Chernoff bounds, this is indeed a PAC learner. □

8.38 Even though the coordinates are drawn in a dependent order, every draw is independent, and every bit is drawn from \pi, hence the resulting string is distributed according to $\pi^{\otimes n}$. □

8.39 Let $P$ be a random root-leaf path in $T$, so that

$$\Delta(T) = \mathbb{E}[|P|] = \sum_{\text{paths } P} |P|2^{-|P|} = \mathbb{E}[\log_2(2^{|P|})] \leq \log_2(\mathbb{E}[2^{|P|}]) = \log_2(s) \tag{79}$$

as desired. □
8.40 (a) By the OSSS inequality, we have \( \maxinf(f) \geq \frac{\sum \delta_i(T) \inf_i(f)}{\Delta_i(T)} \geq \frac{\var(f)}{\Delta_i(f)} \).

(b) Using Madrijanis’s upper bound \( \Delta(f) \leq DT(f) \leq \deg(f)^3 \) and part (a), we conclude \( \maxinf(f) \geq \var(f)/\deg(f)^3 \).

(c) Picking \( T \) such that \( \delta_i(T) = \delta_i(f) \), we have by the OSSS inequality that \( I(f) \delta_i(T) \geq \sum \delta_i(T) \inf_i(f) \geq \var(f) \) and hence \( I(f) \geq \frac{\var(f)}{\delta_i(f)} \).

8.41 (a) WLOG we may assume \( T = T \) is deterministic. Set \( J = [n] \setminus \{i\} \). If \( x' \) is such that there exists \( x_i, x_i' \in \Omega \) with \( f(x', x_i) \neq f(x', x_i') \), then whenever \( T \)'s queries are answered according to \( x' \), it must query \( x_i \). It follows from Proposition 8.24 that \( \inf_i(f) \leq \delta_i(T) \).

(b) Let \( T \) be a randomized decision tree with \( \delta(f) = \max_i \delta_i(T) \). Then by OSSS, part (a), and Ex. 8.33, we have, for transitive-symmetric \( f \):

\[
\var(f) \leq \sum \delta_i(T) \inf_i(f) \leq \sum \delta_i(T)^2 \leq \frac{\Delta(f)}{n} \tag{80}
\]

and hence \( \Delta(f) \geq \sqrt{n \var(f)} \).

8.42 (a) Let \( T_\omega \) be the depth \( k - 1 \) decision tree starting at the node obtained by following the outgoing edge labelled by \( \omega \) from the root of \( T \). The event that \( \{T_\omega(x) \neq f(x)\} \) is a subset of the union of the two events \( \{T(x) \neq f(x)\} \) and \( \{f(x) \neq f(x^{(\omega+\omega)})\} \). The probability of the first is \( \err(T) \), and the probability of the second is \( \frac{1}{2} \inf_i(f) \), and hence \( \err(T) \leq \frac{1}{2} \inf_i(f) \).

(b) A randomized decision tree of depth 0 takes values \(-1 \) and \( 1 \) with some fixed probability, independent of \( x \), and hence its error is a convex combination of \( \Pr[f(x) = 1] \) and \( \Pr[f(x) = -1] \), and is therefore at least the minimum of these two numbers.

(c) Since the error of a randomized decision tree is a convex combination of errors of deterministic trees, part (a) can be extended to the case that \( T \) is also randomized: namely, if \( T \) is a randomized decision tree with \( p_i := \Pr[i \text{ is the root of } T] \), and \( T' \) is the randomized decision tree rooted at a random child of the root of \( T \), then we have

\[
\err(T') \leq \err(T) + \frac{1}{2} \sum_i p_i \inf_i(f) \tag{81}
\]

The base case of the induction is (b), and (81) will help us complete the inductive step as follows. By construction, we have \( \delta_i(T') \leq \delta_i(T) - p_i \) and hence

\[
\frac{1}{2} \sum \delta_i(T') \inf_i(f) + \err(T') \leq \frac{1}{2} \sum \delta_i(T) \inf_i(f) + \err(T') - \frac{1}{2} \sum_i p_i \inf_i(f) \leq \frac{1}{2} \sum \delta_i(T) \inf_i(f) + \err(T)
\]

Since \( \text{depth}(T') < \text{depth}(T) \), the result follows by induction. This gives a version of the OSSS inequality with a slightly worse constant — indeed, if \( \err(T) = 0 \), i.e. \( T \) computes \( f \), then this says \( \sum_i \delta_i(T) \inf_i(f) \geq 2 \min\{\alpha, 1 - \alpha\} \), and since \( \var(f) = 4\alpha(1-\alpha) \),
we obtain $\sum_i \delta^{(\pi)}_i (\mathcal{T}) \text{Inf}_i(f) \geq \frac{1}{2} \text{Var}(f)$ in general (although when $\alpha = 1/2$, we indeed get the sharp constant 1). However, this proof has the advantage of robustness: for all decision trees which approximately compute $f$, say, with error $< (\frac{1}{2} - \epsilon) \text{Var}(f)$, we still have essentially the same inequality, with constant $\Omega(\epsilon)$.