CS 710: Complexity Theory 4/29/2010

Lecture 28: Harmonic Analysis

Instructor: Dieter van Melkebeek Scribe: Nathan Collins and Andrew Bolanowski

In the last lecture we discussed probabilistically checkable proofs and the PCP theorem (which we did not prove). Today we give an alternate (equivalent) version of the PCP Theorem that is more useful for making hardness of approximation arguments. We also give two examples of hardness of approximation results, for MAX-3-SAT and MAX-IND-SET. Finally, we prove one inclusion on a weaker version of the PCP Theorem stated in the last lecture. One part of this proof uses discrete harmonic analysis, so we introduce enough discrete harmonic analysis to present that part of the argument.

1 PCP Theorems and Hardness Approximation

Recall the PCP Theorem stated last time:

Theorem 1 (The PCP Theorem)**.** NP $=$ PCP$(O(\log n), O(1))$

A PCP$(r(n), q(n))$ for the language L is a probabilistic oracle TM V (the verifier) s.t.

- If $x \in L$ then $(\exists \Pi) \Pr[V^{\Pi}(x)] = 1$.
- If $x \notin L$ then $(\forall \Pi) \Pr[V^{\Pi}(x)] \leq 1/2$.

where the quantification is over proofs and the probability is over random coins flipped by the verifier. The verifier uses $r(n)$ random coins and makes $q(n)$ queries to the proof Π.

1.1 An Approximation Equivalent to the PCP Theorem

To see the relationship between the PCP Theorem and hardness of approximation we prove that the following theorem is equivalent to the PCP Theorem.

Theorem 2. *There exists $\alpha < 1$ and a poly-time computable f from 3-SAT to 3-SAT s.t. if a 3-CNF x is* not *in 3-SAT then the fraction of clauses of the 3-CNF $f(x)$ that can be simultaneously satisfied is less than α, and if x is in 3-SAT, then so is $f(x)$.*

Theorem 3. *The* PCP *Theorem is equivalent to Theorem 2.*

Proof. $(\Downarrow)$: Assuming the PCP Theorem, there exist constants q (the $O(1)$ number of queries to the proof) and c (the constant in the $O(\log n)$ number of random bits used), and a poly-time verifier V, such that for each 3-CNF x

- If $x \in$ 3-SAT then there exists a proof Π

$$\Pr_{\rho \in \{0,1\}^{c \log n}}[V^{\Pi}(x)] = 1. \tag{1}$$

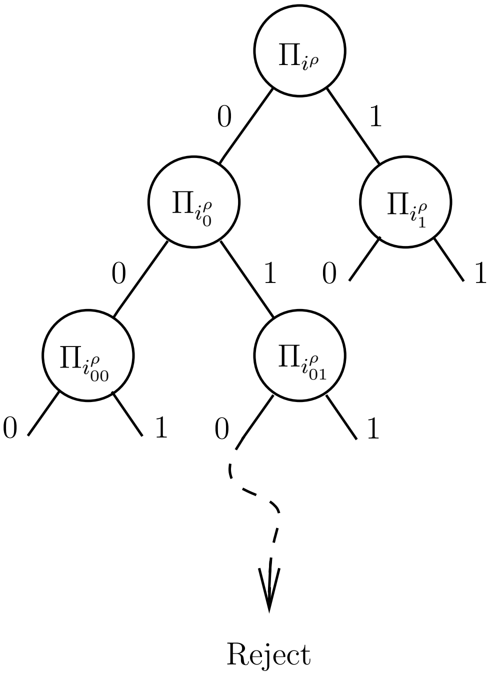


Figure 1: The bits in Π queried by V_ρ^Π depend only on ρ and bits of Π previously queried. The nodes are labeled by the queried proof bit and the outgoing edges are labeled by the nodes value.

- If $x \notin$ 3-SAT then for all proofs Π

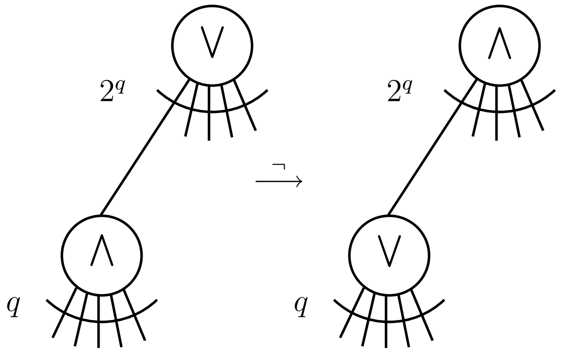
$$\Pr_{\rho \in \{0,1\}^{c \log n}} [V^\Pi(x)] \leq 1/2. \tag{2}$$

Let ρ denote a random bit string in $\{0,1\}^{c \log n}$ and V_ρ denote V supplied with random bits ρ. Once ρ is fixed, the value of $V_\rho(x)$ is completely determined by the (up to) q bits of the supplied proof Π that $V_\rho(x)$ queries. The first bit that $V_\rho(x)$ queries is independent of Π, and in general the k^{th} bit queried is determined by the values of the previous $k-1$ bits queried. Figure 1 illustrates a decision tree for this procedure, in which the index $i^\rho_{b_1,\ldots,b_k}$ is the $k+1^{st}$ proof bit that is queried when the first k bits queried had the values $b_1, \ldots, b_k$.[1]

Now we construct a CNF representation of the decision tree. Let the variable x_i represent the value of the i^{th} bit in Π. Then the disjunction of all rejecting path "signatures" gives a DNF that is true iff $V_\rho^\Pi(x)$ rejects, where by path signature we mean the conjunction of variables and variable negations that correspond to the query indices and values corresponding to that path. As Figure 2 illustrates, negating that DNF gives a CNF which is false iff $V_\rho^\Pi(x)$ rejects, and is hence equivalent to $V_\rho(x)$. Since there are at most 2^q paths, and each path has length at most q, the CNF constructed has at most 2^q clause, each of length at most q.

Let C_ρ denote the 3-CNF corresponding to the CNF constructed in the last step, gotten by expanding each $(\leq q)$-clause into at most q 3-clauses. Then C_ρ has at most $q2^q$ clauses and we

[1] The indices that are queried for a fixed Π can be written as $i^\rho, i^\rho_{\Pi_{i^\rho}}, i^\rho_{\Pi_{i^\rho}\Pi_{i^\rho_{\Pi_{i^\rho}}}}, i^\rho_{\Pi_{i^\rho}\Pi_{i^\rho_{\Pi_{i^\rho}}}\Pi_{i^\rho_{\Pi_{i^\rho}\Pi_{i^\rho_{\Pi_{i^\rho}}}}}}$, etc.



Path on which
V_ρ^Π rejects

Negation of
rejecting path

Figure 2: Since the set of paths is prefix free, V_ρ^Π rejects iff the query indices and values of a rejecting path are realized by Π. We form a DNF of rejecting path "signatures" and negate it to get a CNF that is true precisely when no rejecting path is realized.

define $f(x) = \bigwedge_\rho C_\rho$, where the top two levels of ANDs are contracted to form a CNF. Then $f(x)$ has at most $n^c q 2^q$ 3-clauses and either

- $x \in$ 3-SAT: Then $f(x) \in$ 3-SAT too, since the PCP system has perfect completeness (Equation (1)), and hence there exists Π that makes $V_\rho^\Pi(x)$ accept and C_ρ be satisfied for all ρ.
- $x \notin$ 3-SAT: Then $f(x) \notin$ 3-SAT, since the PCP system has soundness 1/2 (Equation (2)), and hence for all Π the computation $V_\rho^\Pi(x)$ rejects and C_ρ is unsatisfied for at least one half of all ρ. If C_ρ is unsatisfied then at least one of its clauses is unsatisfied, and so the proportion of $f(x)$'s clauses that are unsatisfied in this case is at least $(1/2)(1/q2^q) > 0$, giving $\alpha = 1 - 1/q2^{q+1} < 1$

($\Downarrow$, alternative construction): Let ρ denote a random bit string in $\{0,1\}^{c \log n}$ and V_ρ denote V supplied with random bits ρ. For each $\pi \in \{0,1\}^q$ simulate V_ρ on x, supplying bits from π as responses to V_ρ's proof queries, and keeping track of the proof bit indices that V_ρ queries. I.e., the first time V_ρ queries a proof bit, record the index queried and supply the first bit in π. On each subsequent query to the proof, check if the queried index has been queried previously and if so supply the same bit of π as before. Each time a *new* proof index is queried, supply the next unused bit in π. Let $V_\rho^{[\pi]}$ denote the simulated machine.

Each proof index queried by $V_\rho^{[\pi]}$ can depend on the values of the previously queried proof bits, but in any case $V_\rho^{[\pi]}$ queries at most q proof bits, so a q-bit π is sufficient for our simulation. Let the variable y_i represent the i^{th} bit in a hypothetical proof Π. For each $\pi \in \{0,1\}^q$ we can form a disjunction $\psi_{\rho,\pi}$ that is false iff the variables corresponding to the proof indices that $V_\rho^{[\pi]}$ queries agree with the values in π. E.g., if $q = 5$, $\pi = 01101$, and $V_\rho^{[\pi]}$ queries indices i_1, i_2, and i_3, then $\psi_{\rho,\pi} = y_{i_1} \vee \overline{y}_{i_2} \vee \overline{y}_{i_3}$.

We can now form a CNF φ that is satisfied iff its variables are set in such a way that they correspond to a proof that makes V_ρ accept for each $\rho \in \{0,1\}^{c \log n}$. I.e.,

$$\varphi = \bigwedge_{\neg V_\rho^{[\pi]}(x)} \psi_{\rho,\pi},$$

where the conjunction is taken over all $\rho \in \{0,1\}^{c \log n}$ and $\pi \in \{0,1\}^q$ that cause $V_\rho^{[\pi]}$ to reject x.

If x is in 3-SAT, then Equation (1) says that there exists a satisfying assignment for φ, since a proof Π that makes V_ρ accept x for all ρ can't include the bits that make any *included* $\psi_{\rho,\pi}$ fail. If x isn't in 3-SAT, then Equation (2) says that any assignment will correspond to a proof Π that makes V_ρ reject for at least $1/2$ of all ρ's. In terms of our $\psi_{\rho,\pi}$ defined above, this says that we fail to satisfy some $\psi_{\rho,\pi}$ included in the conjunction φ, for at least $1/2$ of all ρ's. Now, there are at most $2^q n^c$ clauses in φ, so the fraction of clauses that are unsatisfied is at least $\frac{(1/2)n^c}{2^q n^c} = 1/2^{q+1}$

So far, we have $\leq_m^p$ reduced 3-SAT to SAT with "$\alpha = 1 - 1/2^{q+1}$." To finish this direction of the proof, we note that we can efficiently convert φ into an equivalent 3-SAT ${}_3\varphi$, where each clause C in φ corresponds to at most q 3-clauses in ${}_3\varphi$, and whenever an assignment fails to satisfy C it also fails to satisfy at least one of the q 3-clauses corresponding to C. So, completeness is preserved, and whenever x is not in 3-SAT, the fraction of clauses that any assignment fails to satisfy is at least $(1/q)(1/2^{q+1})$, and so we have $\alpha = 1 - 1/(q2^{q+1}) < 1$.

($\Uparrow$): This direction is easier. Suppose that f is a reduction and $\alpha < 1$ as in Theorem 2. Given a 3-CNF x let V^Π calculate $f(x)$ and treat Π as a Boolean assignment for $f(x)$. Since V can only access $O(\log n)$ proof bits, it can't simply check all of $f(x)$'s clauses. However, with $O(\log n)$ random bits, V can choose a clause of $f(x)$ at random and then check if Π satisfies it, accessing at most 3 bits of Π. This procedure has perfect completeness, since if $f(x)$ is satisfiable then Π can be a satisfying assignment. If $f(x)$ is unsatisfiable, then with probability α the clause V chooses of $f(x)$ is not satisfied by Π, giving soundness $\leq \alpha$. Our definition of PCP requires the soundness be $\leq 1/2$, so can we repeat this procedure $\lceil \frac{\log 1/2}{\log \alpha} \rceil$ times (the error is one-sided). This shows that 3-SAT $\in$ PCP$(O(\log n), O(1))$ and the PCP Theorem follows from the NP-completeness of 3-SAT. □

1.2 Implications of the PCP Theorem on Hardness of Approximation

Theorem 2 can be used to derive approximation bounds on certain NP optimization problems. Examples include

- MAX-3-SAT: If a poly-time approximation algorithm A for MAX-3-SAT existed which gave approximations to a factor better than α, then we could determine membership in 3-SAT by running A on $f(x)$ and accepting iff $A(f(x)) \geq (1-\alpha)\#f(x)$,[2] giving P = NP. We conclude that approximating MAX-3-SAT to within α is NP-hard.

  In fact, it turns out that for all $\varepsilon > 0$, it is NP-hard to $(7/8+\varepsilon)$-approximate MAX-3-SAT in poly-time. This follows from the existence of a PCP$(O(\log n), 3)$ for the problem E-3-LIN (exactly 3 variables, linear equations) that has completeness $c = 1 - \varepsilon$ and soundness $s = 1/2 + \varepsilon$. An instance of E-3-LIN is a collection of linear equations: $x_{i,1} \oplus x_{i,2} \oplus x_{i,3} = b_i$ where the b_i are constants (either 0 or 1). The instance is in the language if there is a setting of the variables that satisfies each equation. The PCP for E-3-LIN works by generating three

[2] $\#f(x)$ denotes the number of clauses in $f(x)$.

indices i_1, i_2, and i_3, and a bit b, and verifying that $\Pi_{i_1} \oplus \Pi_{i_2} \oplus \Pi_{i_3} = b$. With this PCP system $s \geq 1/2$ since for random values $\Pi_{i_1} \oplus \Pi_{i_2} \oplus \Pi_{i_3} = b$ is satisfied half of the time. Also, it's unlikely to have $c = 1$ with this system, since deciding whether the system of equations has a solution can be solved in polynomial time using Gaussian elimination.

The approximation result for MAX-3-SAT follows from the PCP for E-3-LIN by converting each linear equation from an E-3-LIN instance into four clauses such that if the original equation is not true with a given assignment then at least one of the four clauses must be false. [3]

- MAX-IND-SET: Given a 3-CNF x, form a graph G corresponding to $f(x)$ as follows: For each 3-clause c in $f(x)$ add 7 nodes to G, where the nodes are labeled by c and one of the 7 possible satisfying assignments to c's 3 variables. Add an edge to G between any two nodes that together correspond to a conflicting variable assignment. Then for any clause c the 7 nodes corresponding to c form a clique, and so any anti-clique in G includes at most 1 node from each of these 7 node clusters. When $f(x)$ is satisfiable, we can choose 1 node from each cluster, so the max independent set has size the number m of clauses in $f(x)$. When $f(x)$ is not satisfiable, we can satisfy at most αm of $f(x)$'s clauses, and so the maximum independent set will have size at most αm. So, an α-approximation of MAX-IND-SET would imply P = NP.

  Using more "technology" we can get a $\frac{1}{n^{1-\varepsilon}}$-approximability bound, for some $\varepsilon > 0$, where n is the number of nodes in the graph. Note that this is a strong result since we can trivially get a $1/n$ approximation by picking a single node.

That's all we are going to say about hardness of approximation as it relates to PCPs.

2 A Weaker PCP Theorem

The proof of "the" PCP Theorem is quite elaborate, and we won't see it in this class, but we will prove a weaker PCP Theorem today. This result gives a flavor of the stronger PCP theorem - we are able to verify a proof by querying only a constant number of bits in the proof. The result we prove is weaker than "the" PCP theorem as we use a polynomial number of random bits.

Theorem 4. *NP $\subseteq$ PCP($poly(n), O(1)$).*

Notice that since PCPruns in polynomial time, limiting it to a polynomial number of random bits doesn't give anything useful. In the proof, we will create a PCP with perfect completeness for 3-CNF. Then we can use mapping reductions and the fact that 3-CNF us NP-complete. We want a randomized polytime oracle Turing Machine V, such that

$$y \in 3 - CNF \Rightarrow \exists \Pi \Pr(V^{\Pi}(y) accepts) = 1 \tag{3}$$

$$y \notin 3 - CNF \Rightarrow \forall \Pi \Pr(V^{\Pi}(y) accepts) \leq \frac{1}{2} \tag{4}$$

And, V uses $O(1)$ oracle calls. A natural proof Π would be the satisfying assignment, but that appears to need $O(n)$ oracle calls. So we need some method to put more information into the bits. We use the Hadamard code.

[3] See lecture 3, page 6, from CS 880 in 2004 for a more detailed explanation of the hardness of approximation result for MAX-3-SAT.

Proof. The idea is to somehow convert certificates of membership in an NP language into something easily verified by a PCP system. To do this we'll use quadratic equations and Hadamard coding. Given some 3-CNF φ we form an equivalent collection of quadratic equations and ask if they can be simultaneously satisfied. E.g., we translate the clause $c = x \vee \overline{y} \vee z$ into the polynomial equation $(1-x)y(1-z) = 0$ and introduce a new variable $\xi = xy$ to reduce the degree from 3 to 2. Working over $\mathbb{Z}_2$ we have an equivalent system of two equations: $y \oplus yz \oplus \xi \oplus \xi z = 0$, and the new equation $\xi \oplus xy = 0$.

So, suppose we have converted φ into a system of N quadratic equations in n variables $x_1, \ldots, x_n$. We want to convert this into a single equation. For a fixed assignment a to x, we could pick just 1 of the equations at random, but this will give terrible soundness. It could be the case that only 1 of the clauses were unsatisfiable. So we instead take a random linear combination of the equations. Notice that all the equations will have zero on the right hand side, and denote the k^{th} left hand side by Q_k. Then for each $k \in [N]$ we have $Q_k = \bigoplus_{i,j\in[n]} q_{kij}x_ix_j$ for some coefficients q_{kij}. The satisfiability of φ is thus reduced to finding a solution to the system $Q_k = 0$, for $k \in [N]$.

Given a candidate assignment $a = (a_1, \ldots, a_n)$ of the x_is we have

$$\Pr_{\rho\in\{0,1\}^N}[\underbrace{\bigoplus_{k\in[N]} \rho_k Q_k(a) = 1]}_{(*)} = 1/2, \tag{5}$$

if a fails to satisfy $Q(a) \equiv 0$, since $\bigoplus_{k\in[N]} \rho_k Q_k(a)$ is the inner product of $Q(a)$ (a *non-zero* vector) with a random vector ρ. If a is a satisfying assignment, then $Q_k(a) = 0$ for all $k \in [N]$. Thus, since $\bigoplus_{k\in[N]} \rho_k Q_k(a) = 0$, so the probability in (5) is zero.

Define $b_{ij} = a_ia_j$ for all $i, j \in [n]$. Then any quadratic in $\{a_i\}$ is linear in $\{b_{ij}\}$. Let the proof π be a Hadamard encoding of b. Then checking $(*)$ amounts to querying one position in π: the position corresponding to the vector with ij^{th} entry given by $\bigoplus_{k\in[N]} \rho_k q_{kij} b_{ij}$. For soundness we need to be able to reject π's that don't Hadamard encode a b as defined above. To probabilistically check that a given π is such a Hadamard code we perform the following tests

- Test 1: Probabilistically check that π is close to a Hadamard encoding of some b. Once we know π is close to a Hadamard code we can use local decodability.
- Test 2: Probabilistically check that $b_{ij} = b_{ii}b_{jj}$. This is true for b as above since $x = x^2$ in GF(2).

To perform Test 1 we choose $x, y \in \{0,1\}^{n^2}$ at random and verify that $\pi(x) \oplus \pi(y) = \pi(x \oplus y)$. Completeness of this test is clearly 1. If π is *not* close to any valid Hadamard code, i.e.

$$\forall b \in \{0,1\}^{n^2} \Pr_{x\in\{0,1\}^{n^2}}[\pi(x) \neq \langle b, x\rangle] \geq \gamma \tag{6}$$

for some $\gamma > 0$, then

$$\Pr[\text{Test 1 fails}] \geq \gamma. \tag{7}$$

To prove that (6) implies (7) we will use harmonic analysis, and we postpone that proof until the end of the current proof. So, assume Test 1 passes with high probability. We can run the test multiple times to boost confidence. Note that if $\gamma \leq \frac{1}{4}$ then the b is unique and we can use local decoding. Each bit b_{ij} can be determined by

$$b_{ij} = \langle b, x\rangle + \langle b, x + e_i\rangle \tag{8}$$

Since each of the inner products gives the correct value with probability $\geq 1-\gamma$, by the union bound, the equality holds with probability $\geq 1-2\gamma$, which is ¿$\frac{1}{2}$

To perform Test 2, we define matrices A and B by $A_{ij} = b_{ii}b_{jj}$ and $B_{ij} = b_{ij}$, choose $x, y \in \{0,1\}^{n^2}$ at random, and check that

$$x^\top Ay = x^\top By.$$

If $A = B$, i.e. if b is of the desired form, then Test 2 passes with probability 1. If $A \neq B$, then the probability that Test 2 fails is at least 1/4, since if $A \neq B$, then $x^\top A \neq x^\top B$ with probability at least 1/2, since A and B differ in at least one column (think inner products again), and whenever $x^\top A \neq x^\top B$, we have $x^\top Ay = x^\top By$ with probability exactly 1/2. This test requires only three queries to the proof, since

$$x^\top Ay = \bigoplus_{i,j} x_i A_{ij} y_j = (\bigoplus_{i,j} x_i b_{ii})(\bigoplus_{i,j} b_{jj} y_j)$$

and

$$x^\top By = \bigoplus_{i,j} x_i y_j b_{ij},$$

and so we can query π at the positions corresponding to the vectors with ij^{th} entry $\delta_{ij}x_i$, $\delta_{ij}y_j$, and x_iy_j, respectively, where $\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$ is a Kronecker delta function. □

3 Harmonic Analysis

We now develop enough harmonic analysis to prove the implication (6) implies (7) in the proof of the weak PCP Theorem. Hopefully people have seen some form of *continuous* harmonic analysis on $\mathbb{R}$ or $\mathbb{C}$, where functions are approximated using sin and cos (which are called *harmonics*). Today we describe a *discrete* theory for functions

$$f : G \to \mathbb{C},$$

where G is a group. In this discrete theory the harmonics are *characters* of the group, i.e. homomorphisms from G into the multiplicative group of complex numbers $\mathbb{C}^\times$. Recall that a *homomorphism* is a function that respects the group operation, i.e. $\phi : G \to \mathbb{C}^\times$ is a character of G if for all $g, h \in G$ we have $\phi(gh) = \phi(g)\phi(h)$. Notice that if $g \in G$ has finite order, which is true of all $g \in G$ if G is finite, then $\phi(g) \in e^{i\mathbb{R}}$ the unit circle, since $\phi(g)^{\mathrm{ord}(g)} = 1$.

3.1 Properties of Group Characters

Before we get to our application of discrete harmonic analysis, we list without proof a few general properties of group characters for finite groups

- The group characters form an orthonormal set with respect to the inner product

  $$\langle f_1, f_2 \rangle = \frac{1}{|G|} \sum_{g \in G} \overline{f_1(g)} f_2(g) = \mathop{\mathbb{E}}_{x \in G}(\overline{f_1(g)} f_2(g)),$$

  where $x \mapsto \overline{x}$ is complex conjugation.

- There at most $|G|$ characters, since the space of functions $G \to \mathbb{C}$ is a $\mathbb{C}$-vector space of dimension $|G|$. For many groups the number of characters is equal to the size of $|G|$, in which case we biject G with its characters and write χ_g for the character corresponding to $g \in G$, and then the characters form an orthonormal basis for the space of $G \to \mathbb{C}$ functions. I.e., given $f : G \to \mathbb{C}$, there exist *Fourier coefficients* $\{\hat{f}(g)\}_{g \in G}$ such that

  $$f = \sum_{g \in G} \hat{f}(g)\chi_g.$$

  The Fourier coefficients are easy to calculate since the χ_g are orthonormal, namely

  $$\hat{f}(g) = \langle \chi_g, f \rangle.$$

- Let $\hat{f} = g \mapsto \hat{f}(g)$ denote the vector of f's Fourier coefficients. Then the map $f \mapsto \hat{f}$ "respects" the inner product, i.e. given $f_1, f_2 : G \to \mathbb{C}$ we have

  $$\langle f_1, f_2 \rangle = |G| \langle \hat{f}_1, \hat{f}_2 \rangle,$$

  which yields a Parseval identity

  $$\|f\|_2^2 = |G| \|\hat{f}\|_2^2. \tag{9}$$

- For the *convolution* of f_1 and f_2, defined by

  $$(f_1 * f_2)(z) = \frac{1}{|G|} \sum_{x+y=z} f_1(x) f_2(y),$$

  we get the relation

  $$\widehat{f_1 * f_2} = \hat{f}_1 \hat{f}_2.$$

  This last equality allows for the reduction of some $O(n^2)$ computations to $O(n \log n)$ computations in signal processing.

3.2 Completing the Proof of the Weak PCP Theorem

That's all we are going to say about harmonic analysis in general. For $G = (\{0,1\}^n, \oplus)$, or equivalently $(\{\pm 1\}^n, \cdot)$, where equivalence comes from the mapping $x \mapsto -1^x$, all elements are order 1 or 2, and so the characters are of the form $G \to \{\pm 1\} \subseteq \mathbb{C}$. In this case, we have characters

$$\chi_g(x) = (-1)^{\langle g, x \rangle}, \tag{10}$$

where the inner product in the exponent is given by

$$\langle x, y \rangle = \bigoplus_{i \in [n]} x_i y_i.$$

Such χ_g are characters since $\langle g, x \oplus y \rangle = \langle g, x \rangle \oplus \langle g, y \rangle$. The fact that $\langle g, x \rangle$ is the x^{th} bit of the Hadamard code for g provides the connection with our previous work. Since our characters take values in $\{\pm 1\} \subseteq \mathbb{R}$, we can drop the complex conjugation in the definition of inner product for functions $G \to \mathbb{C}$, and we get

$$\begin{aligned}\hat{f}(g) = \langle \chi_g, f \rangle &= \mathop{\mathbb{E}}_{x \in G}[\chi_g(x) f(x)] \\ &= \Pr_x[f(x) = \chi_g(x)] - \Pr_x[f(x) \neq \chi_g(x)] \\ &= 1 - 2 \Pr_x[f(x) \neq \chi_g(x)], \end{aligned} \tag{11}$$

and more generally

$$\delta_{gh} = \langle \chi_g, \chi_h \rangle = \mathop{\mathbb{E}}_{x \in G}[\chi_g(x) \chi_h(x)]. \tag{12}$$

We now complete the proof of the Weak PCP Theorem, by proving that if for all $b \in \{0,1\}^{n^2}$

$$\Pr_{x \in \{0,1\}^{n^2}}[\pi(x) \neq \langle b, x \rangle] \geq \gamma$$

for some $\gamma > 0$, then

$$\Pr[\text{Test 1 fails}] \geq \gamma.$$

We prove the contrapositive via a calculation, where we use the $\{\pm 1\}$ domain for the values taken by π

$$\begin{aligned}&1 - 2\Pr[\text{Test 1 fails}] \\ &= \Pr[\text{Test 1 passes}] - \Pr[\text{Test 1 fails}] \\ &= \mathop{\mathbb{E}}_{x,y}[\pi(x)\pi(y)\pi(x \oplus y)] \\ &= \mathop{\mathbb{E}}_{x,y}[(\sum_{g_1} \hat{\pi}(g_1)\chi_{g_1}(x))(\sum_{g_2} \hat{\pi}(g_2)\chi_{g_2}(y))(\sum_{g_3} \hat{\pi}(g_3)\chi_{g_3}(x)\chi_{g_3}(y))]\end{aligned}$$

by switching to the Fourier domain and using the fact that χ_{g_3} is a homomorphism

$$= \sum_{g_1, g_2, g_3} \hat{\pi}(g_1)\hat{\pi}(g_2)\hat{\pi}(g_3) \mathop{\mathbb{E}}_x[\chi_{g_1}(x)\chi_{g_3}(x)] \mathop{\mathbb{E}}_y[\chi_{g_2}(y)\chi_{g_3}(y)]$$

by linearity of expectation, the independence of x and y, and the fact that $\mathbb{E}[\chi_g(x)] = 1$

$$= \sum_g (\hat{\pi}(g))^3$$

by (12)

$$\begin{aligned}&\leq \max_g \hat{\pi}(g) \sum_g (\hat{\pi}(g))^2 \\ &= \max_g \hat{\pi}(g)\end{aligned} \tag{13}$$

since $\sum_g (\hat{\pi}(g))^2 = 1$, by (9)

$$= 1 - 2 \Pr_x[\pi(x) \neq \chi_{g_M}(x)]$$

by (11), where g_M maximizes (13). Rearranging, we get

$$\Pr_x[\pi(x) \neq \chi_{g_M}(x)] \leq \Pr[\text{Test 1 fails}] \tag{14}$$

which finishes the proof, since by (10) we can rewrite (14) as

$$(\exists g_M \in \{0,1\}^{n^2}) \Pr_x[\pi(x) \neq \langle g_M, x\rangle] \leq \Pr[\text{Test 1 fails}],$$

if we again interpret π to take values in $\{0, 1\}$, and this is the contrapositive of what we set out to prove.