PCP Theorem

Definition: Suppose r, q are functions. L is in $PCP(r, q)$, if there is a polynomial time verifier V and a constant c satisfying:

- V on input x of length n, a random string $\{0,1\}^{c*r(n)}$, and a 'proof', checks at most $c*q(n)$ bits of the proof (the bits checked depend on x and the random string), non adaptively, and accepts or rejects.
- If x is in L, there is a proof such that V accepts with probability 1 (note that this proof can be taken to be of length at most $cq(n)2^{cr(n)}$).
- If x is not in L, then for any proof, V accepts with probability at most $1/3$.

Theorem: $NP = PCP(\log n, 1)$
Clearly, $PCP(\log n, 1)$ is contained in NP. Other direction is difficult. We will show a weaker version of it.
Theorem: $NP \subseteq \bigcup_{c \in \mathbb{N}} PCP(n^c, 1)$.

Walsh Hadamard Codes

For x and y of same length (say n), let $x \ o \ y$ denote $(\sum_{i=1}^{n} x_i \cdot y_i) \bmod 2$.
For any n, and $k \in \{0,1\}^n$ let $W(y) \in \{0,1\}^{2^n}$ be defined as follows. For the i-th element x in $\{0,1\}^n$, i-th bit of $W(y)$ is $y \ o \ x$ (we sometimes also call it the x-th bit). Sometimes we denote $W(y)$ by W_y and treat W_y as a function from $\{0,1\}^n$ to $\{0,1\}$.
Below the operations are mod 2. Note that W is a linear function in the sense that
$W(x+y)(z) = W(x)(z) + W(y)(z)$, where $+$ is bit wise mod 2 addition.
$W(x \cdot y)(z) = W(y)(x \cdot z)$, where $\cdot$ is bit wise and.
$W(x)(y+z) = W(x)(y) + W(x)(z)$, where $+$ is bit wise mod 2 addition.

Theorem: Any function $f : \{0,1\}^n$ to $\{0,1\}$ is W_u for some u iff f is linear (mod 2).
Proof: Clearly each W_u is linear.
Suppose f is linear. Suppose e_i has all bits 0 except the i-th bit.
$f(x) = \sum_{i=1}^{n} f(x_i e_i)$
$= \sum_{i=1}^{n} x_i f(e_i)$,
$= W_u(x)$, where u has i-th bit $f(e_i)$.

Definition: f, g from $\{0,1\}^n$ to $\{0,1\}$ are ρ-close if they agree on at least ρ fraction of the inputs. f is ρ-close linear function if it is ρ-close to some linear function.

Lemma: Suppose f is a function from $\{0,1\}^n$ to $\{0,1\}$. If $prob(f(x+y) = f(x) + f(y)) \geq \rho \geq 1/2$, then f is a ρ-close linear function.
Note that one can do random verification for $f(x+y) = f(x) + f(y)$, using large enough number of trials.

Lemma: If f is ρ-linear for some $\rho > 3/4$, then there exists a unique linear function $\hat{f}$ such that f is ρ-close to $\hat{f}$.
Proof: Suppose there are two such $\hat{f}$ and $\hat{h}$. But then $\hat{f}$ and $\hat{h}$ are $> 1/2$ close to each other, which is not possible. Why?
$\hat{f} = W_u$ $\hat{h} = W_v$. Suppose, u and v are different on i-th bit.
Then consider any x and x' which differ on exactly i-th bit.
Now, exactly one pair:
$u \ o \ x$ and $v \ o \ x$
or
$u \ o \ x'$ and $v \ o \ x'$
are same.

QuadEQ

Definition: Instance: Given some quadratic equations over n boolean variables u_1 to u_n.
Question: is there assignment to the boolean variables so that all equations are satisfied.

QuadEQ is NP-complete.

Theorem: QuadEQ is in $\bigcup_{c\in\mathbb{N}} PCP(n^c, 1)$.
Consider the equations as
$AU = b$, where A is $m \times n^2$ matrix, b is $m \times 1$, and U is formed by using $U(i, j) = u_i u_j$. We view U as both a $n \times n$ matrix and a vector of length n^2 depending on context.
We need to verify if there is some vector u which satisfies the above.
What should now be the proof?
We use Walsh-Hadamard codes for U and u, that is $f = W(U)$ and $g = W(u)$ of 2^{n^2} and 2^n bits respectively. U can be considered as $u \otimes u$.

We need to verify that

1. f and g are indeed linear functions
2. Check that for some u, $g = W(u)$ and $f = W(u \otimes u)$
3. $AU = b$, U is the matrix obtained from $u \otimes u$.

1.

Use enough random pairs x, y and verify
$f(x) + f(y) = f(x + y)$,
so that if f is not 0.99-linear it will fail the test with 99% probability.

Same for g.

Thus, we have unique linear function $\hat{f}$ and $\hat{g}$ which is 0.99-close to f and g respectively.

How to get values of $\hat{f}$ and $\hat{g}$?

For any x, choose a random r and calculate $f(x + r) - f(r)$.

This will be $\hat{f}(x)$ with high probability (98%).

2. Pick random $\alpha, \beta \in \{0,1\}^n$ and calculate $\hat{f}(\alpha \otimes \beta)$ and $\hat{g}(\alpha)\hat{g}(\beta)$.

Note that $\hat{f}(\alpha \otimes \beta) = U \ o \ (\alpha \otimes \beta) = \alpha U \beta$

$\hat{g}(\alpha)\hat{g}(\beta) = (u \ o \ \alpha)(u \ o \ \beta) = \alpha B \beta$, where $B_{i,j} = u_i u_j$.

Thus, If $\hat{f}$ and $\hat{g}$ are indeed representing U and u respectively, then $\hat{f}(\alpha \otimes \beta)$ and $\hat{g}(\alpha)\hat{g}(\beta)$, must be same.

If U did not represent $u \otimes u$, then probability of above test succeeding is at most $3/4$.
Why? if $U \neq B$, then probablity of $\alpha U \neq \alpha B$ is at least $1/2$. If $\alpha U \neq \alpha B$, then probability of $(\alpha U)\beta$ being not equal to $(\alpha B)\beta$ is at least $1/2$.
Repeating the test a fixed number of times decreases the probability of passing the test for a wrong proof.

3.
Choose $r \in \{0,1\}^m$ at random and compute
$AU\ o\ r$ and $b\ o\ r$.
If $AU \neq b$, then $AU\ o\ r$ and $b\ o\ r$ will not be equal with probability 1/2.

How to compute $AU\ o\ r$:
Using linearity of U, can be done using one query.