Proof of a Weak PCP Theorem

$NP \subseteq PCP(poly(n),1)$

NP $\subseteq$ PCP(poly(n),1)

- On an input of length n, the verifier V can use poly(n) random bits, but still makes only a constant number of queries to the proof
- The total number of possible computation paths of V can be exponential in n, so the number of possible bits queried, over all computation paths, could be exponential in n
- So, the proof can have length exponential in n

Handy Notation and Properties

- Let $u = (u_1, u_2, \ldots, u_n)$, and similarly let v, x, and y be n-dimensional bit vectors

- *Inner product:* $u \odot x = \sum u_i x_i \pmod 2 = u\, x^T \pmod 2$

- *Tensor (or outer) product*:
  $u \otimes x = (u_1x_1, u_1x_2, \ldots, u_1x_n, u_2x_1, \ldots, u_nx_n)$

Handy Notation and Properties

- Let $u = (u_1, u_2, \ldots, u_n)$, and similarly let v, x, and y be n-dimensional bit vectors
- *Inner product:* $u \odot x = \sum u_i\, x_i \pmod 2 = u\, x^T \pmod 2$
- *Tensor (or outer) product*:
  $u \otimes x = (u_1x_1, u_1x_2, \ldots, u_1x_n, u_2x_1, \ldots, u_nx_n)$
- *Inner-Outer Property*:
  $(u \odot x)\,(u \odot y) = (u \otimes u) \odot (x \otimes y)$

Handy Notation and Properties

- Let $u = (u_1, u_2, \ldots, u_n)$, and similarly let v, x, and y be n-dimensional bit vectors
- *Inner product:* $u \odot x = \sum u_i x_i \pmod 2 = u\, x^T \pmod 2$
- *Tensor (or outer) product*:
  $u \otimes x = (u_1x_1, u_1x_2, \ldots, u_1x_n, u_2x_1, \ldots, u_nx_n)$

Handy Notation and Properties

- Let $u = (u_1, u_2, \ldots, u_n)$, and similarly let v, x, and y be n-dimensional bit vectors
- *Inner product:* $u \odot x = \sum u_i x_i \pmod 2 = u\, x^T \pmod 2$
- *Tensor (or outer) product*:
  $u \otimes x = (u_1x_1, u_1x_2, \ldots, u_1x_n, u_2x_1, \ldots, u_nx_n)$
- *Random Subsum Property*: If $u \neq v$ then
  $\Pr_x [u \odot x \neq v \odot x] \geq 1/2$

Walsh-Hadamard encodings

Let $u = (u_1, u_2, \ldots, u_n)$ be a bit vector.

The *Walsh-Hadamard encoding* WH(u) of u is the 2^n-dim. vector of values $u \odot x$ for all $x \in \{0,1\}^n$.

Walsh-Hadamard encodings

Let $u = (u_1,u_2,\ldots,u_n)$ be a bit vector.
The *Walsh-Hadamard encoding* WH(u) of u is the 2^n-dim. vector of values $u \odot x$ for all $x \in \{0,1\}^n$.

WH(u) is the truth table of the function $f(x) = u \odot x$

Walsh-Hadamard encodings

Let $u = (u_1,u_2,\ldots,u_n)$ be a bit vector.
The *Walsh-Hadamard encoding* WH(u) of u is the 2^n-dim. vector of values $u \odot x$ for all $x \in \{0,1\}^n$.

WH(u) is the truth table of the function $f(x) = u \odot x$

There is a 1-to-1 correspondence between WH encodings and linear functions $f:\{0,1\}^n \rightarrow \{0,1\}$
(linear means $f(x) + f(y) = f(x+y)$)

QUADEQ: an NP-Complete Problem

Instance:

- An $m \times n^2$ matrix A with entries in $\{-1,0,1\}$
- An m-dimensional bit vector b

Problem: Is there an n-dimensional bit vector u such that $A\,(u \otimes u)^T = b^T$?

A PCP for QUADEQ

An “NP certificate” that (A,b) is in QUADEQ is simply a bit vector u such that $A\,(u \otimes u)^T = b^T$

A “PCP certificate" that (A,b) is in QUADEQ is the Walsh-Hadamard encoding of both u and $u \otimes u$:

$$WH(u),\quad WH(u \otimes u)$$

The certificate has length $2^n + 2^{n^2}$

Tasks of PCP Verifier

Given a QUADEQ instance (A,b), and a PCP proof, i.e. truth tables of functions f and g

V checks:

- *Linearity*: f = WH(u) and g = WH(w) for some u, w
- *Consistency*: $w = u \otimes u$
- *Satisfiability*: If $g = WH(u \otimes u)$ then $A\,(u \otimes u)^T = b^T$.

Tasks of PCP Verifier

Given a QUADEQ instance (A,b), and a PCP proof, i.e. truth tables of functions f and g

V checks:

- *Linearity*: f = WH(u) and g = WH(w) for some u, w
- *Consistency*: $w = u \otimes u$
- *Satisfiability*: If $g = WH(u \otimes u)$ then $A\,(u \otimes u)^T = b^T$.

It's not possible for V to do this using a constant number of queries, so V will do somewhat weaker tests and have a low probability of error

Linearity Check: f, g are WH Encodings

Naive test:

- For all x, y $\in \{0,1\}^n$, check that $f(x+y) = f(x) + f(y)$
- Do a similar test for g

Problem: this requires too many queries

Linearity Check: f, g are WH Encodings

We say that f is $(1-\delta)$-*close to a linear function* f', where $\delta \in [0,1]$, if $\Pr_x [f(x) = f'(x)] \geq (1-\delta)$

Linearity Check: Given f, and $\delta \in (0,1/4)$
Repeat $\Theta(1/\delta)$ times:
    Choose x and y randomly and uniformly
    Reject if $f(x+y) \neq f(x) + f(y)$
Accept

Linearity Check: f, g are WH Encodings

We say that f is $(1–\delta)$-*close to a linear function* f′, where $\delta \in [0,1]$, if $\Pr_x [f(x) = f'(x)] \geq (1–\delta)$

Linearity Check: Given f, and $\delta \in (0,1/4)$
Repeat $\Theta(1/\delta)$ times:
    Choose x and y randomly and uniformly
    Reject if $f(x+y) \neq f(x) + f(y)$
Accept

Theorem: If f is not $(1–\delta)$-close to a linear function, the linearity test rejects with probability at least 1/2

Tasks of PCP Verifier

Given a QUADEQ instance (A,b), and a PCP proof, i.e. truth tables of functions f and g

V checks:

- *Linearity*: f = WH(u) and g = WH(w) for some u, w
- *Consistency*: $w = u \otimes u$
- *Satisfiability*: If $g = \mathrm{WH}(u \otimes u)$ then $A\,(u \otimes u)^T = b^T$.

Consistency:
If $f = WH(u)$ and $g = WH(w)$ then $w = u \otimes u$

Naive test: Check that

$$f(x)f(y) = g(x \otimes y) \text{ for all } x \text{ and } y$$

If $w = u \otimes u$ then

$$\begin{aligned} f(x)\, f(y) &= (u \odot x)\,(u \odot y) \\ &= (u \otimes u) \odot (x \otimes y) && \text{(by inner-outer property)} \\ &= g(x \otimes y) && \text{(since } w = u \otimes u\text{)} \end{aligned}$$

If $w \neq u \otimes u$ then for some x and y, $f(x)f(y) \neq g(x \otimes y)$

Problem: this requires too many queries

Consistency:
If f = WH(u) and g = WH(w) then $w = u \otimes u$

Naive test: Check that

$$f(x)f(y) = g(x \otimes y) \text{ for all } x \text{ and } y$$

If $w = u \otimes u$ then

$$\begin{aligned} f(x)\, f(y) &= (u \odot x)\, (u \odot y) \\ &= (u \otimes u) \odot (x \otimes y) && \text{(by inner-outer property)} \\ &= g(x \otimes y) && \text{(since } w = u \otimes u\text{)} \end{aligned}$$

Consistency:
If $f = WH(u)$ and $g = WH(w)$ then $w = u \otimes u$

Naive test: Check that

$$f(x)f(y) = g(x \otimes y) \text{ for all } x \text{ and } y$$

If $w = u \otimes u$ then

$$\begin{aligned} f(x)\, f(y) &= (u \odot x)\, (u \odot y) \\ &= (u \otimes u) \odot (x \otimes y) && \text{(by inner-outer property)} \\ &= g(x \otimes y) && \text{(since } w = u \otimes u\text{)} \end{aligned}$$

Theorem: If $w \neq u \otimes u$ then

$$\Pr_{x,y}\left[f(x)f(y) \neq g(x \otimes y) \right] \geq 1/4$$

Consistency:
If f = WH(u) and g = WH(w) then $w = u \otimes u$

Naive test: Check that

$$f(x)f(y) = g(x \otimes y) \text{ for all } x \text{ and } y$$

If $w = u \otimes u$ then

$$\begin{aligned} f(x)\, f(y) &= (u \odot x)\, (u \odot y) \\ &= (u \otimes u) \odot (x \otimes y) && \text{(by inner-outer property)} \\ &= g(x \otimes y) && \text{(since } w = u \otimes u\text{)} \end{aligned}$$

Theorem: If $w \neq u \otimes u$ then

$$\Pr_{x,y} [f(x)f(y) \neq g(x \otimes y)\ ] \geq 1/4$$

For proof, see Arora-Barak, Section 18.4

Consistency:

If $f = WH(u)$ and $g = WH(w)$ then $w = u \otimes u$

Consistency Test: Given $f = WH(u)$, $g = WH(w)$

Repeat a constant number of times:

Choose x and y randomly and uniformly

Reject if $f(x)f(y) \neq g(x \otimes y)$

Accept

Consistency:

If $f = WH(u)$ and $g = WH(w)$ then $w = u \otimes u$

Consistency Test: Given $f = WH(u)$, $g = WH(w)$

Repeat a constant number of times:

Choose x and y randomly and uniformly

Reject if $f(x)f(y) \neq g(x \otimes y)$

Accept

Theorem: If $w \neq u \otimes u$, the consistency test rejects with constant probability (close to 1)

Tasks of PCP Verifier

Given a QUADEQ instance (A,b), and a PCP proof, i.e. truth tables of functions f and g

V checks:

- *Linearity*: f = WH(u) and g = WH(w) for some u, w
- *Consistency*: $w = u \otimes u$
- *Satisfiability*: If $g = WH(u \otimes u)$ then $A\,(u \otimes u)^T = b^T$.

Satisfiability: If $g = WH(u \otimes u)$ then $A\,(u \otimes u)^T = b^T$

Recall: A is a $(m \times n^2)$ matrix and b is a m-dimensional vector representing m quadratic equations, each of the form

$$A_k\,(u \otimes u)^T = b_k\,,$$

where A_k is the kth row of A

Also, $A_k\,(u \otimes u)^T$ is exactly $g(A_k)$

Naive test: given A, b, and $g = WH(u \otimes u)$

check that for all k, $1 \le k \le m$, $g(A_k) = b_k$

Problem: the number of queries is linear in m

Satisfiability: If $g = WH(u \otimes u)$ then $A\,(u \otimes u)^T = b^T$

Satisfiability test: Given A, b, and $g = WH(u \otimes u)$

Repeat a constant number of times

Take a random subset of the equations

Compute their sum mod 2; let the result be

$$z\,(u \otimes u)^T = c,$$

where z is a n^2-dim. vector, c is a constant

Reject if $g(z) \neq c$

Accept

Theorem: If $A\,(u \otimes u)^T \neq b^T$ then each iteration of the test fails with probability at least 1/2

Proof: Apply the random subsum property

Tasks of PCP Verifier

Given a QUADEQ instance (A,b), and a PCP proof, i.e. truth tables of functions f and g

V checks:

- *Linearity*: $f = WH(u)$ and $g = WH(w)$ for some u, w
- *Consistency*: $w = u \otimes u$
- *Satisfiability*: If $g = WH(u \otimes u)$ then $A\,(u \otimes u)^T = b^T$.

Tasks of PCP Verifier

Given a QUADEQ instance (A,b), and a PCP proof, i.e. truth tables of functions f and g

V checks:

- *Linearity*: f and g are $(1-\delta)$-close to $f' = WH(u)$ and $g' = WH(w)$
- *Consistency*: $w = u \otimes u$
- *Satisfiability*: If $g' = WH(u \otimes u)$ then $A\,(u \otimes u)^T = b^T$.

Tasks of PCP Verifier

Given a QUADEQ instance (A,b), and a PCP proof, i.e. truth tables of functions f and g

V checks:

- *Linearity*: f and g are $(1-\delta)$-close to $f' = WH(u)$ and $g' = WH(w)$
- *Consistency*: $w = u \otimes u$
- *Satisfiability*: If $g' = WH(u \otimes u)$ then $A\,(u \otimes u)^T = b^T$.

Problem: Need to update the Consistency and Satisfiability checks, to account for the fact that f and g are close to, but may not equal, f′ and g′.

Next Time

- We'll finish the proof that NP $\subseteq$ PCP(poly(n), 1)
- We'll see one more application, to hardness of approximating the Clique problem