Today
- Hardness Amplification w/o XOR Lemma (Sudan-Trevisan-Vadhan)

Lecture 31
Computational Complexity (28 May 2020)
Instructor: Prahladh Harsha

Different notions of hardness:

Worst-case hardness:

f is w.c hard for ckts of size S if $\forall$ ckts C of size S, $\exists x$, $f(x) \neq C(x)$

average hardness:

f is (S,δ)-hard if for all ckts C of size S

$$\Pr_{x \sim \{0,1\}^n}[f(x) = C(x)] \leq \frac{1+\delta}{2} \qquad (\delta \in (0,1))$$

Mildly average: $\delta \sim$ close to 1. $\frac{1+\delta}{2} = 99\%$

Strongly average: $\delta \sim$ close to 0. $\frac{1+\delta}{2} \approx 51\%$.

Nisan-Wigderson Generator:

Hypothesis: $\exists$ a $f \in E = 2^{O(n)}$ s.t. f is $(2^{\delta n}, \frac{1}{2}+\varepsilon)$-hard for some $\delta \in (0,1)$ & negligible ε

Conclusion: $BPP = P$.

Qn: Can one weaken the hypothesis for NW from strongly average-case hard fn to mildly average case hard fn in E

$\to$ even to worst-case hard fn in E

Worst-Case hard fn in E ———— [IW]

$\Downarrow$

Mildly average case hard fn in E ———— [BFNW]

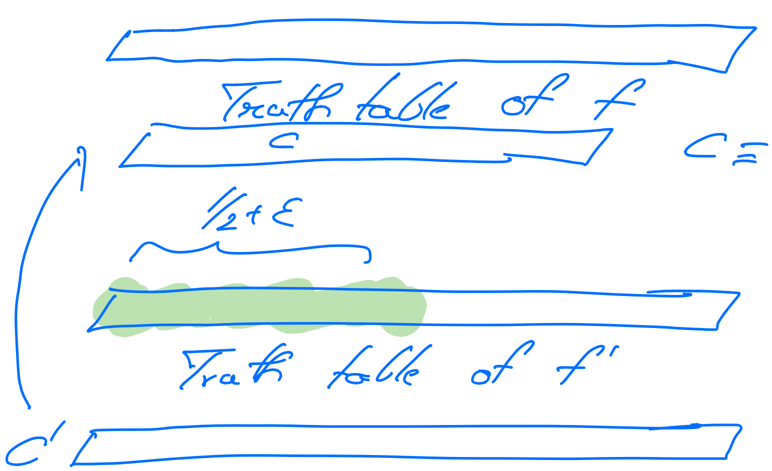
$\Downarrow$

Strongly average case hard fn in E ————→ [NW]

$\Downarrow$

BPP = P

Today: an alternate (more direct proof) of IW + BFNW result due to Sudan-Trevisan-Vadhan.



$f : \{0,1\}^n \to \{0,1\}$

$C \equiv f$

$f' : \{0,1\}^{n'} \to \{0,1\}$

$\Pr[C'(x) = f(x)] \geq \frac{1}{2} + \varepsilon.$

Suppose f' - encoding of f $\quad f' = \mathcal{C}(f)$ for some $\mathcal{C}$

& furthermore. C^* - is decodable even w/ $(\frac{1}{2}-\varepsilon)$-errors.

then $f' = C(f)$ is a candidate.

strongly-~~hard~~ average case hard fn.

<u>Differences from the usual coding setup</u>

1. f & f' are never written down at any point

We only have access to them either

— an alg in E that computes f

— a ckt of size $2^{\delta n}$ that approximates f'.

Suppose $C : \{0,1\}^{2^n} \to \{0,1\}^{2^{n'}}$

& C ran in time polynomial in its input length (i.e., 2^n)

$(2^{n'} = \mathrm{poly}(2^n)$ i.e., $n' = O(n))$

then $f \in E \Rightarrow f' = C(f) \in E$.

$f \in DTIME(2^{cn})$.

① Truth table of can be written down in time. $2^n \cdot 2^{cn} = 2^{(c+1)n}$

② Compute $C(f)$. — takes time. $\mathrm{poly}(2^n) = 2^{c'n}$

③ Alg to compute f' (read off the relevant bit from the tt of $f' = C(f)$).

Encoding requirement: C - poly time encoder

$(f \in E \Rightarrow f' = C(f) \in E)$.

Decoding Issue :

$$C' \; \Big\} \, 2^{\delta n} \qquad \Pr[f' = C'] \geq \frac{1+\varepsilon}{2} \quad \xRightarrow{\text{Decoding}} \quad C \; \Big\} \, 2^{\delta n} \qquad C \equiv f \text{ everywhere.}$$

$$f : \{0,1\}^n \to \{0,1\} \quad \xRightarrow[\text{XOR Lemma}]{\text{Yao's}} \quad f^{(k)} : \{0,1\}^{nk} \to \{0,1\}$$

$$f^{(k)} = \bigoplus_{i=1}^{k} f(x_i)$$

Req $\quad \underline{n' = O(n)} \qquad k = O(1)$.

$k = O(1)$ is not good enough.

So, Yao's XOR Lemma (as stated before) is insufficient for mildly average case hard to strongly average case hard. (for the parameters we seek).

One potential soln:

Prove Yao's XOR Lemma works even when the k inputs are not independent.

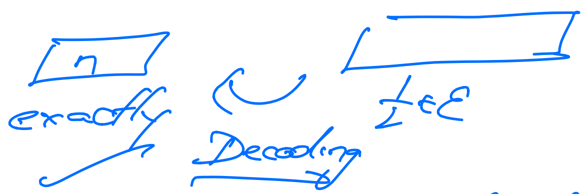
[Derandomizing Yao's XOR Lemma - BFNW; IW]

STV Soln: Completely avoids XOR Lemma.

Goldreich-Levin Alg:

Had: $\{0,1\}^n \to \{0,1\}^{2^n}$ $(n \mapsto 2^n)$ ✗

$(n \mapsto \text{poly}(n))$ ✓



Want a code C that has the decoding properties of Had but has poly rate instead of exponential rate.

Code:

$C = RM \circ Had$.

$RM_{\mathbb{F}}[m,d]: \mathbb{F}^{\binom{m+d}{d}} \longrightarrow \mathbb{F}^{\mathbb{F}^m} = \left(\{0,1\}^r\right)^{2^m}$

coeffts of m-multivariate degree d $\longrightarrow$ eval of the polynomial.

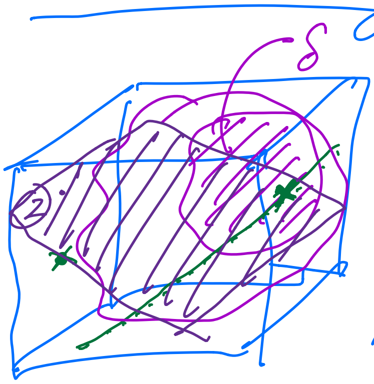
$\mathbb{F} = GF(2^n)$

Had: $\{0,1\}^n \longrightarrow \{0,1\}^{2^n}$

$\underline{RM \circledcirc Had}$: $\mathbb{F}^{\binom{m+d}{m}} \longmapsto \{0,1\}^{\overbrace{2^n \cdot \mathbb{F}^m}}$

$f \in \{0,1\}^{2^n} \longrightarrow f' \in \{0,1\}^{2^{n'}}$

List-decoding of RM codes:



$p : \mathbb{F}^m \to \mathbb{F}$

$C : \mathbb{F}^m \to \mathbb{F}$ (corruption of p)
(circuit)

Want to correct C to obtain p.

Suggestion:

To decode p at the pt x.
- choose a random line ℓ thru x
- Query C at $d+1$ pts on ℓ (other than x).
- Interpolate to o/p value at x.

$\Pr_\ell[C(1^{st} \text{ point on line}) \neq p(x)] \leq \delta$

$\Pr_\ell[\forall i \in [d+1],\ C(i^{th} \text{ point on line}) = p(x)] \geq 1 - (d+1)\delta$

We can decode if the fraction of errors $<< \frac{1}{d+1}$

Instead of requiring that all $(d+1)$ pts on line are uncorrupted.

Will only ask for 90% of pts to be uncorrupted & then use the unique decoding alg to RS code to obtain the poly p.

Alternate Alg:

Input: $C : \mathbb{F}^m \to \mathbb{F}$ (given as a ckt)
x

Output: $p(x)$

Guarantee: $\Pr[C(x) \neq p(x)] \leq \delta$

Alg:

(1) Choose a random line ℓ thru x.

(2) Query C on all pts of ℓ except x

(3) Unique decode the corrupted RS codeword to obtain the poly p restricted to line ℓ.

(4) Output $p_\ell(x)$.

If $\delta \leq \frac{1}{100}$, then can recover polynomial correctly everywhere?

What have we achieved

Increased the corruptions from

$$\frac{1}{d} \longrightarrow \frac{1}{100} \longrightarrow \frac{99}{100}$$

Unique decoding Algorithm for RS code
$\downarrow$
List decoding Algorithm for RS code

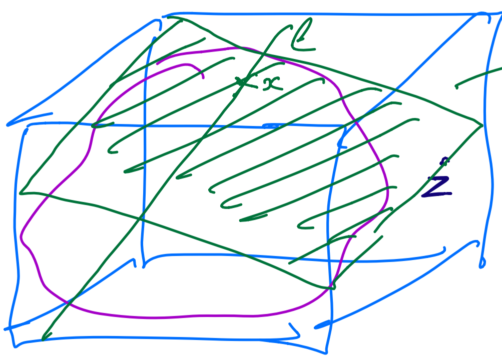
Modified Alg : Input x:

1. Choose a random line ℓ thru x
2. Query C on all pts of line ℓ except x
3. List-decode the "corrupted RS" codeword to output a list of poly $P_1 \ldots P_\ell$.
4. How does one disambiguate among the different poly $P_1 \ldots P_\ell$? Which one do I evaluate to output $P(x)$?

Trick : Guess / Hard wire the value of the poly P at a random point z in $\mathbb{F}^m$

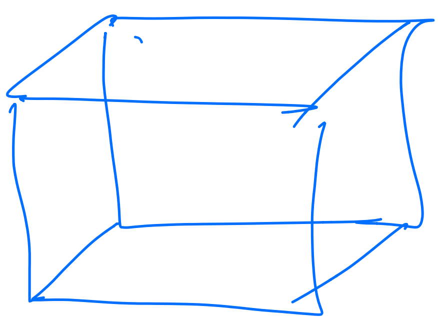
Input: $\begin{cases} x, & C \text{ (ckt that approximates } p) \\ z, & p(z) = a. \end{cases}$

Alg:
1. Choose a random line ℓ thru x.
2. Consider the plane ρ that contains ℓ & z.
3. Query C on all pts on the plane ρ.
4. List-decode bivariate RM on the plane to obtain poly $p_1, \ldots, p_\ell$.
5. Disambiguation: Find i s.t $p_i(z) = a$. If there is more than one, halt.
6. Output $p(x)$.



$p_1, \ldots, p_\ell$ - list of poly on plane.

Hopefully, there exists at most one poly p_i s.t $p_i(z) = a$.



$p : \mathbb{F}^m \to \mathbb{F}$ $\quad \mathbb{F} = GF(2^k)$

$'p' : \mathbb{F}^m \times \{0,1\}^k \to \{0,1\}$

$(x, \bar{a}) \longmapsto \langle p(x), a \rangle$

Suppose there is a ckt C'

$$\Pr_{x,a}\left[C'(x,\bar{a}) = p'(x,a)\right] \geq \frac{1}{2}+\varepsilon$$

For $\varepsilon/2$ - fraction of x's.

$$\Pr_a\left[C'(x,a) = p'(x,a) = \cancel{\langle p(x), a\rangle}\right] \geq \frac{1}{2}+\frac{\varepsilon}{2}$$

$\downarrow$ (GL algorithm) $\downarrow$

For $\varepsilon/2$ - fraction of x's

$C''(x)$ s.t. $C''(x) = p(x)$

$\downarrow$ RM decoding

To ~~obtain~~ a C''' that computes p everywhere.