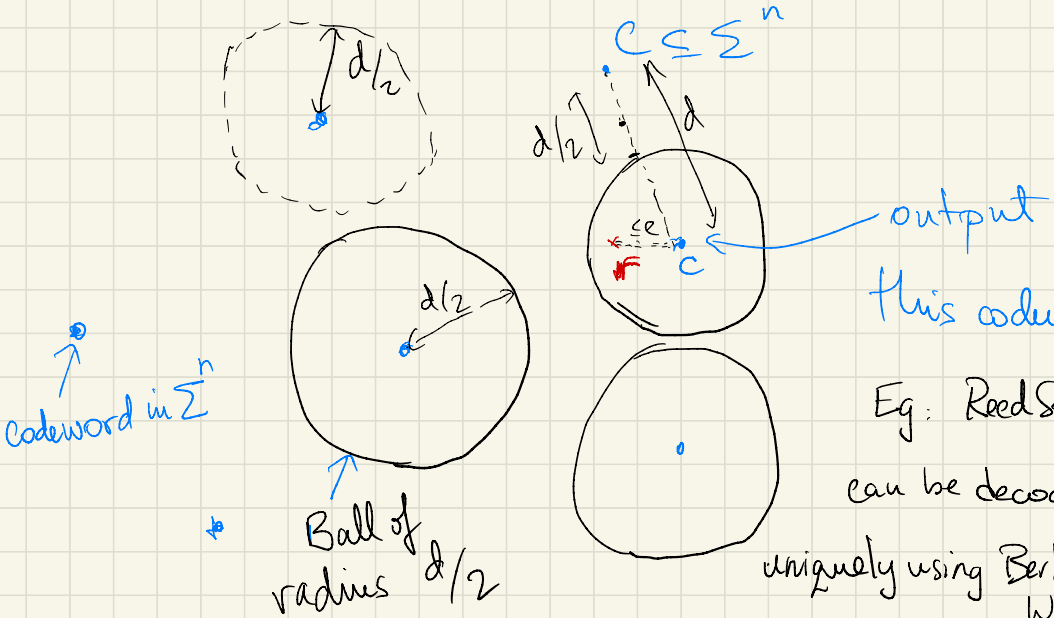
Local list decoding of the Hadamard code
(Goldreich Levin Theorem)

Unique decoding: (global)

Given code $C \subseteq \Sigma^n$ of min dist. d.
and a received word $r \in \Sigma^n$ s.t.
r has $\leq e$ errors, where $e < d/2$

Goal: output unique $c \in C$ s.t.

$$\mathrm{dist}(r, c) \leq e.$$



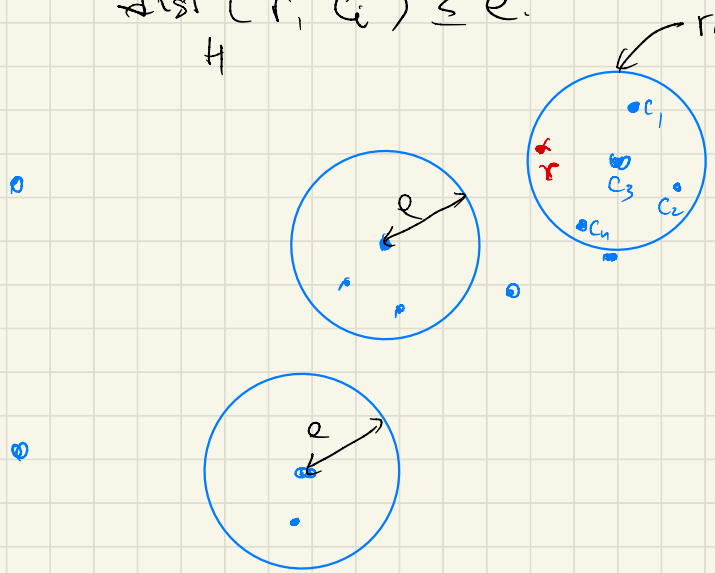
Eg: Reed Solomon can be decoded uniquely using Berlekamp Welch

Global list decoding.

- Given received $r \in \Sigma^n$ and error bound e with $\frac{d}{2} \leq e$

- Goal: output list of <u>all</u> codewords $c_1, c_2, \ldots c_L \in C \subseteq \Sigma^n$ s.t.

$$\mathrm{dist}_H(r, c_i) \leq e.$$



Output: c_1, c_2, c_3, c_4

Goal: efficient list decoding.

Def: C is (e, L) list decodable if given received $r \in \Sigma^n$ and error bound e

① combinatorial: # of codewords of C within e errors from r is $\leq L$.

② algorithmic: can output the list in ① efficiently in n & L.

- List decoding was first defined by Sudan who shows that $RS(k, n)$ of dist $d = n-k$ is list decodable from

$$\frac{d}{2} < e < n - \sqrt{2n(n-d)}$$

Improved by Guruswami - Sudan to

$$e < n - \sqrt{n(n-d)} \leftarrow \text{Johnson bound}$$

(Technique: generalization of Berlekamp Welch)

Interesting questions

- list decoding radius of a code (largest ρ for which list is poly size)

- tradeoffs for list size & params of code

Obs: Many natural fams of codes are list decodable.

Some generalizations

- List learning

- List decodable lattices

- List decodable group homomorphisms.

Above is also explicit: each codeword is explicitly specified.

Implicit local list decoding.

Def: A probabilistic alg M γ-computes a funct f if $\forall x$ in the domain

$$\Pr[M(x) = f(x)] \geq \gamma$$

Def: Local list decoding (implicitly)

A prob. oracle A is (e, L)-local list decodable if $\forall f$ st $\text{dist}(f, C) < e$ if, when A is given oracle access to f.

- A^f outputs a list of prob. oracle machines $M_1, M_2, \ldots, M_L$ st. $\forall c \in C$ with $\text{dist}(f, C) < e$ wp $> \frac{1}{2}$ over randomness of A^f, $\exists$ machine M_i that $7/8$-computes c.
- A & A^f run in time poly$(\log n, L)$

Ideally: poly$(\log n)$

We'll show Goldreich Levin's theorem: local list decoding of Hadamard code

$n = 2^k$

$$\mathrm{Had} = \left\{ l_a(x) = \sum a_i x_i \bmod 2 \quad l_a : \{0,1\}^k \to \{0,1\} \right\}_{a \in \{0,1\}^k}$$

Goldreich-Levin: $\exists$ probabilistic alg A st.

given $f : \{0,1\}^k \to \{0,1\}$ with $\mathrm{dist}(f, \mathrm{Had}) < \frac{1}{2} - \varepsilon$ relative

runs in time $\mathrm{poly}(1/\varepsilon, k)$ & outputs a list

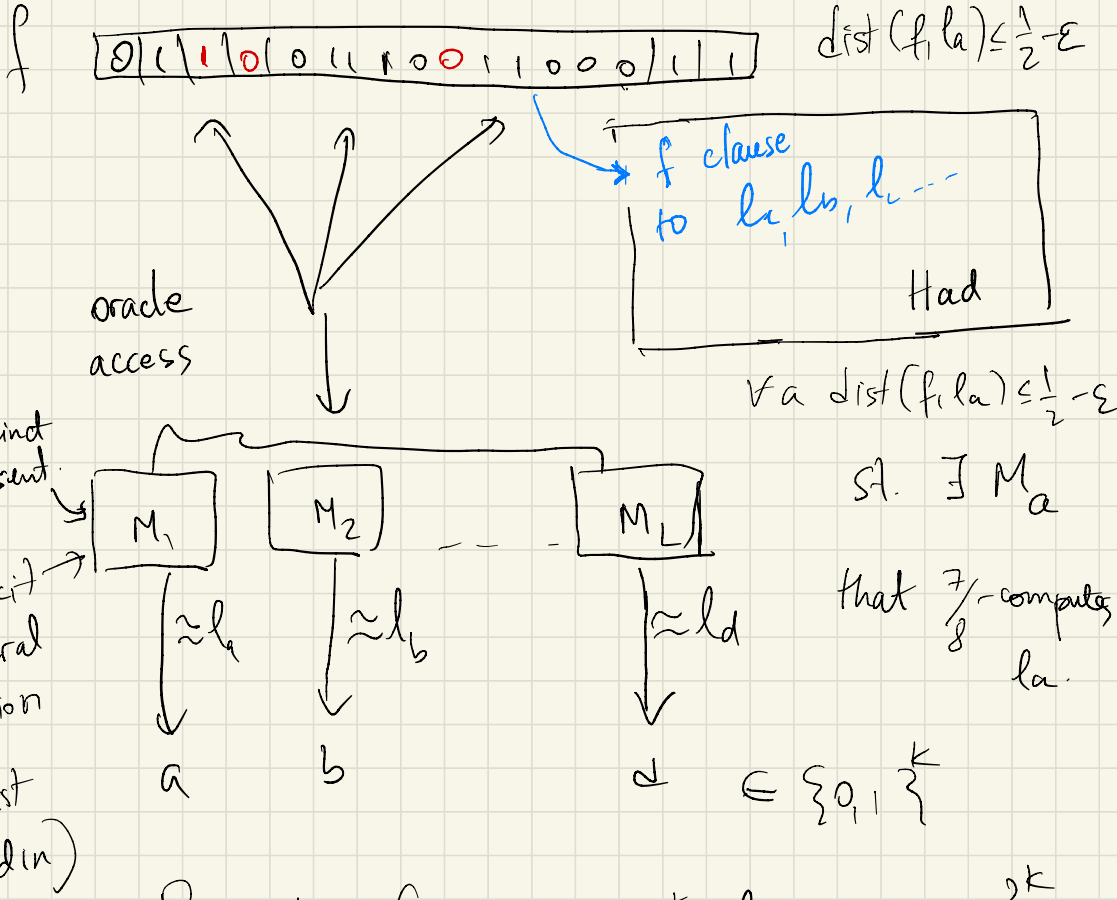
L st. if $\mathrm{dist}(f, l_a) \le \underbrace{\frac{1}{2} - \varepsilon}_{\text{fraction of error}}$ then

$l_a \in L$ wp $> 1/2$.

(explicit: a is output)

Obs: $\mathrm{dist}(\mathrm{Had}) = \min \mathrm{dist}(l_a, l_b) = \frac{1}{2}$

So unique decoding is at $e < \frac{1}{4}$.



Remark: Can represent $l_a \in \{0,1\}^{2^k}$

by $a \in \{0,1\}^k$. which

we can learn explicitly for Had. whp.

(as we'll see soon)

<u>Simple case</u>: $e \le 1/8 < \frac{1}{4}$ (unique decoding)

Claim 9: If $e \le 1/8$ $\exists$ $O(k \log k \log 1/\delta)$ alg that outputs a s.t. $\mathrm{dist}(f, l_a) \le 1/8$ wp $1-\delta$.

Proof: Unique decoding:

For each bit $i \in [k]$

for $j = 1$ to t

pick $x \in \{0,1\}^k$ u.a.r.

compute $a_{ij} = f(x+e_i) - f(x)$

elementary unit vectors in $\mathbb{F}_2^k$

Output $a_i = \mathrm{maj}_j \; a_{ij}$

ie. output $a = (a_1, a_2, \ldots a_k) \in \{0,1\}^k$

Remark

$l_a(x+e_i) - l_a(x) = l_a(e_i) = a_i$

Proof:

For j: $a_{ij} = a_i$ wp $1 - \frac{1}{4} = 3/4$

$\Pr[f(x+e_i) \ne l_a(x+e_i)] \le 1/8$

$\Pr[f(x) \ne l_a(x)] \le 1/8$

since $x+e_i$ is u.a.r if x is u.a.r

$\forall i \quad \Pr[\text{ maj of } a_{ij} \text{ is incorrect }] \leq e^{-\Omega(t)}$

(Chernoff bound)

$\Pr[\text{ all } a_i\text{'s are correct } i \in [k]] \geq (1 - e^{\Omega(t)})^k > 1 - \delta$

$t = O(\log k, 1/\delta)$

Overall for all k bits of a we need $O(k \log k)$ time.

(assume $\delta = \frac{1}{100}$)

So we output the correct a w.p. $99/100$

Here : the machine M output is

s.t. $M(x) = l_a(x) \quad \forall x.$

Case: large distance but with assumptions

oracle access

Claim: If $\mathrm{dist}(l_a, f) < \frac{1}{2} - \varepsilon$ then $\exists$ alg.

s.t. w.p $> \frac{1}{2}$ outputs M_a s.t. $\Pr_z[M_a(z) = l_a(z)] \geq 7/8$.

in time $O(\frac{1}{\varepsilon^2} k \log k)$

Proof: Given oracle access to f.

Alg: Pick $x^{(1)}, x^{(2)}, \ldots x^{(s)} \in \{0,1\}^k$ i.a.r

(Assume can) Obtain $l_a(x^{(1)}), l_a(x^{(2)}) \ldots l_a(x^{(s)})$

$f(z+x^{(1)}) \approx l_a(z+x^{(1)})$

Define func $A_{x^{(1)} x^{(2)} \ldots x^{(s)}}$ as

$$A_{x^{(1)} x^{(2)} \ldots x^{(s)}}(z) = \mathrm{maj}\Big[\underbrace{f(z+x^{(1)}) - l_a(x^{(1)})}_{\approx l_a(z)},$$

$$\vdots$$

$$f(z+x^{(s)}) - l_a(x^{(s)})\Big]$$

We'll show: A is $\frac{1}{8}$ close to l_a.

each diff. supposed to be equal to $l_a(z)$

don't actually know these values

Analysis: Define rand vars $E_1, \ldots E_s$

$$E_i = \begin{cases} 1 & \text{when } f(z + x^{(i)}) = \ell_a(z + x^{(i)}) \\ 0 & \text{ow} \end{cases}$$

Note $\mathbb{E}[E_j] = \Pr[f(z + x^{(i)}) = \ell_a(z + x^{(i)})]$

agreement of f and ℓ_a

by assumption on f.

$$\geq \frac{1}{2} + \varepsilon$$

We'll bound $\Pr_{x^{(1)}, \ldots x^{(s)}, z}\left[A_{x^{(1)}, \ldots x^{(s)}}(z) \neq \ell_a(z)\right] =$

$$= \Pr\left[\sum_j E_j < s/2\right]$$

$$\leq \Pr\Big[\Big|\sum E_j - \underbrace{\mathbb{E}\left[\sum E_j\right]}_{\geq (\frac{1}{2}+\varepsilon)s}\Big| \leq \varepsilon s\Big]$$

By chebyshev's ineq. $\leq \dfrac{\mathrm{Var}\left[\sum_j E_j\right]}{\varepsilon^2 s^2}$

$\mathrm{Var}[X] = \mathbb{E}\left[(X - \mathbb{E}[X])^2\right]$

(7)

Claim : If X and Y are pairwise independend (meaning $\mathbb{E}[XY] = \mathbb{E}[X]\,\mathbb{E}[Y]$) then

$$\mathrm{Var}[X+Y] = \mathrm{Var}[X] + \mathrm{Var}[Y].$$

Claim : $\mathrm{Var}[E_i] < 1/4$ for $\varepsilon > 0$, $\forall i \in [s]$

(Prove above claims at home)

$$(*) \leq \frac{s}{4\varepsilon^2 s^2} = \frac{1}{4\varepsilon^2 s}$$

Take $s = 4/\varepsilon^2$

So we proved that

$$\Pr_{x^{(1)}\, x^{(2)} \ldots z}\left[A_{x^{(1)}}(z) \neq l_a(z) \right] \leq \frac{1}{16}$$

By Markov $\Pr_{x^{(1)} \ldots x^{(s)}}\left[\Pr_z\left[A_{x^{(1)} \ldots x^{(s)}}(z) \neq l_a(z) \right] \geq \frac{1}{8} \right] \leq \frac{1}{2}$

$$\Rightarrow \Pr_{x^{(1)} \ldots x^{(s)}}\left[\Pr_z\left[\overset{\to f}{A}_{x^{(1)} \ldots x^{(s)}} \neq l_a(z) \right] \leq \frac{1}{8} \right] \geq \frac{1}{2}$$

Remarks about above algorithm

- do not have access to $l_a(x^{(i)})$ (that's what we need to find)

- Running time $O\left(\frac{1}{\varepsilon^2} k \log k\right)$ (if we knew $l_a(x^{(i)})$)

- Can run through all possible vals

$$\underset{\overset{\parallel}{b_1}}{l_a(x^{(1)})}, \underset{\overset{\parallel}{b_2}}{l_a(x^{(2)})} \cdots \underset{\overset{\parallel}{b_s}}{l_a(x^{(s)})}$$

So 2^s possibilities.

Each gives a machine $A^{b_1 \ldots b_s}_{x^{(1)} \ldots x^{(s)}}$

$s = O(1/\varepsilon^2) \Rightarrow 2^{O(1/\varepsilon^2)}$ machines

$\Rightarrow O(2^{1/\varepsilon^2} k \log k)$ alg.

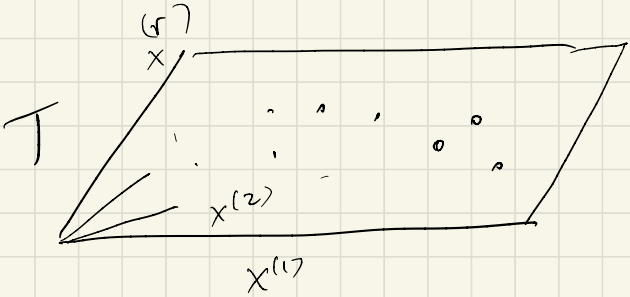
Can do better: use pseudorandomness

Recall $x^{(1)} \dots x^{(s)}$ are independent above but we only use pairwise independence in claim 10.

New idea 1: pick a subspace of small dim (dim $\log s$)

$r = \log s$ i.e. pick $x^{(1)}, \dots, x^{(r)} \xleftarrow{\text{uar}} \{0,1\}^k$

$\forall S \subset [r]$ consider $x^S = \sum_{i \in S} x^{(i)}$



There are 2^r x^S's that belong to the subset in $\{0,1\}^k = \mathbb{F}_2^k$ generated by $x^{(1)}, \dots, x^{(r)}$

New idea 2: Fix value of possible l_a at $x^{(1)}, \dots, x^{(r)}$. Say $l_a(x^{(i)}) = b_i \in \{0,1\}$

Then l_a is fixed on entire supspace T;

i.e. $l_a(x^S) = \sum_{i \in S} l_a(x^{(i)})$

$\alpha_i \in \{0,1\}$ coeffs

linear

$= \sum_{i \in S} b_i$

Only need to guess l_a at $\log s$ many positions. Everything else in the subsp gen by $x^{(1)} \dots x^{(\log t)}$ is obtained by taking lin comb of b_i's.

So, We can assume know val of l_a at s posns, namely at all x^S, $S \subseteq [\log s]$

Recall $s = O(1/\varepsilon^2)$ (# of machines output)

Let $E_i = \begin{cases} 1 & \text{if } f(z + x^{(i)}) = l_a(x^{(i)}) \\ 0 & \text{o.w.} \end{cases}$

Claim: Variables E_i, $i = 1, \dots, s$ gen as above are pairwise independent. (Show at home)

Final alg: (local list decoder for Had_k)

outputting oracles

Pick $x^{(1)} \dots x^{(r)}$ i.u.r in $\{0,1\}$

$\forall (b_1 \dots b_r) \in \{0,1\}^r$ $\qquad r = \lceil \log 4/\varepsilon^2 \rceil$

$\forall S \subseteq \{1, \dots r\}$. $\quad S \neq 0$

Assign $x^S = \sum_{i \in S} x^{(i)}$

$b^S = \sum_{i \in S} b^{(i)}$

the subspace gen by $x^{(1)} \dots x^{(r)}$

Define $A^{b_1 \dots b_r}_{x^{(1)} x^{(2)} \dots x^{(r)}}(z) = \mathrm{Maj}_{S \subset \{1..r\}} f(x^S + z) - b^S$

correcting above oracles

For $i = 1$ to k

for $j = 1$ to $c \log k$

pick $z \in \{0,1\}^k$ u.a.r, $a_{ij} = A(z + e_i) - A_z$

output $a_i = \mathrm{maj}(a_{i1} \dots a_{i\, c\log k})$

Proof: By Claim 10

Machine $A^{b_1 \dots b_r}_{x^{(1)} \dots x^{(r)}}$ is $7/8$-close to l_a.

By Claim 9 can correct/decode l_a.

Running time: $O\left(\frac{1}{\varepsilon^2} \cdot \frac{1}{\varepsilon^2} \cdot k \log k\right)$

List size: $s = O\left(1/\varepsilon^2\right)$

$\Downarrow$

List decoding radius for Had is $\frac{1}{2} - \varepsilon$.

(exp many codew at dist $\frac{1}{2}$)

Applications to crypto / pseudorandomness

Boolean circuit : inputs $x_1 \dots x_n$

outputs $y_1 \dots y_m$

gates : AND, OR, NOT

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

$m = 1$ then f is called predicate

Pseudorandom generator (PRG) [Blum Micali Yao type PRG]

Rand variable X over $\{0,1\}^n$ is (S, ε)-pseudorandom

if $\forall$ circuit C of size $\leq S$

$$\left| \Pr_{x \sim X}\left[C(x) = 1 \right] - \Pr_{y \sim U_n}\left[C(y) = 1 \right] \right| < \varepsilon$$

To C it is indistinguishable

if $x \sim X$ or $x \sim U_n$

uniform over n bits

Def: One-way permutation:

A bijective fnc $f: \{0,1\}^n \to \{0,1\}^n$ is

(S, ε) - one way permutation if $\forall$ circuit S of size $\leq S$ C cannot succeed in inverting f except on an $\leq \varepsilon$ fraction of inputs, ie.

$$\Pr_{\substack{x \sim \{0,1\}^n \\ U_n}} \left[C(f(x)) = x \right] \leq \varepsilon$$

Def: Hard core predicates

A function $B: \{0,1\}^n \to \{0,1\}$ is (S, ε)-hardcore predicate for perm. $f: \{0,1\}^n \to \{0,1\}^n$ if

$\forall$ circuit C of size $\leq S$. (the circuit cannot compute a function of the unknown x except with bias ε)

$$\Pr_{x \sim \{0,1\}^n} \left[C(f(x)) = B(x) \right] \leq \frac{1}{2} + \varepsilon$$

Thm: (Corollary to Goldreich Levin alg we saw above)

\label{PRG-GL}

Let $p:\{0,1\}^n \to \{0,1\}^n$ be an (s, ε) one way permutation computed by circuits of size t.

Let $g:\{0,1\}^{2n} \to \{0,1\}^{2n}$ be $g(x,y) = (p(x), y)$.

Then ① g is a one-way permutation

② $B(x,y) = x \cdot y \bmod 2$ is a (s', ε')-hard core predicate for $g(x,y)$.

③ $s' = \text{poly}(s, t, n)$

Punchline / Thm PRG-GL $\Rightarrow$ PRG $\Rightarrow$ many extensions to obtain pseudorand bits

Thm: (Blum Micali Yao): If B is a hard core pred. for f, then $(f(x), B(x))$ is a PRG

obtains one extra pseudorand bit

Proof of Thm. PRG - GL from the local list decoder GL.

① easy to check.

②+③ Assume for sake of contradiction that $B(x,y)$ is not (s', ε') hard core for g, so $\exists$ a circuit C of size $\leq s'$ s.t.

$$\Pr_{x,y \sim \{0,1\}^n}\left[C(p(x), y) \neq B(x,y) \right] < \frac{1}{2} - \varepsilon'$$

(for some s', ε' to be decided later)

By Markov:

$$\Pr_x\left[\Pr_y\left[C(p(x), y) \neq B(x,y) \right] > \frac{1}{2} - \varepsilon'' \right] < \frac{\frac{1}{2} - \varepsilon'}{\frac{1}{2} - \varepsilon''} = 1 - \varepsilon'''$$

Hence

$$\Pr_x\left[\Pr_y\left[C(p(x), y) \neq B(x,y) \right] < \frac{1}{2} - \varepsilon'' \right] > \varepsilon'''$$

Hence for an ε'' fraction of x's we have

$$\Pr_y\left[\; C(p(x), y) = B(x,y)\right] \geq \frac{1}{2} + \varepsilon''$$

For such an x, define $f_x(y) = C(p(x), y)$

$$= x \cdot y \quad \text{wp} > \frac{1}{2} + \varepsilon''$$

$$\Downarrow \; = \ell_x(y)$$

We can now show that local list decoder above implies we can find x & hence invert $p(x)$

For such x for which $\Pr_y\left[C(p(x), y) = B(x,y)\right] \geq \frac{1}{2} + \varepsilon''$

Use GL alg to output list L of all a's that agree with f_x on $> \frac{1}{2} + \varepsilon''$ fraction of inputs.

Given value of $p(x)$.

Compare $p(x)$ with $p(a)$ for all $a \in L$ & output a if $p(x) = p(a)$

we succeeded to invert $p(x)$ $\Leftarrow$

Parameters:

fraction of x's for which we can invert $p(x)$

To get contradiction, need $\varepsilon''' > \varepsilon$ and size of new circuit $< S$.

So p is not (S,ε) one way permut.

E.g. take $\varepsilon''' = 2\varepsilon$, $\varepsilon'' = \varepsilon$ so

$$\varepsilon' = \frac{1}{2} - \left(\tfrac{1}{2} - \varepsilon\right)(1-2\varepsilon) = 2(\varepsilon - \varepsilon^2)$$

- Note: When we make a query to f_x in the GL alg above, we need to replace it with a circuit computation.

So total comput time / circuit size

$$T = O\Big(\underbrace{S' \frac{1}{\varepsilon^4} n \log n}_{\text{for the list}} + \underbrace{\frac{1}{\varepsilon^2} t}_{\text{for evaluating } 1/\varepsilon^2 \text{ many } p(a)'\text{s}}\Big)$$

Choose $S' = \mathrm{poly}(S, t, n, 1/\varepsilon)$ s.t. $T < S$, concluding the contradiction ✓