Last time:

- Sketch pf that if $\mathcal{C}, \mathcal{D}$ is s.t. $\mathcal{C}$ contains many ($n^{\omega(1)}$ many) functions that are all pairwise uncorrelated under $\mathcal{D}$, then no SQ alg. can eff. learn $\mathcal{C}$ when dist. is $\mathcal{D}$.
  - HW problem 5: no eff. SQ alg. exists for parities, DNFs, DTs.
- Start unit on crypto. hardness of learning "rich" concept classes.

Today:

- computational hardness of learning $\mathcal{C}$ = all poly(n)-size Boolean circuits

  poly(n) many samples suffice, b/c $|\mathcal{C}| \le 2^{\text{poly}(n)}$

  $\frac{1}{\varepsilon}\left(\ln|\mathcal{C}| + \ln\frac{1}{\delta}\right)$

  based on existence of pseudorandom function families
- mapping the boundary of efficient learnability
- start hardness of learning based on public-key cryptography (trapdoor 1-way permutations)

Questions?

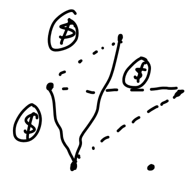
Note: average-case hardness assumptions are stronger than worst-case ('.

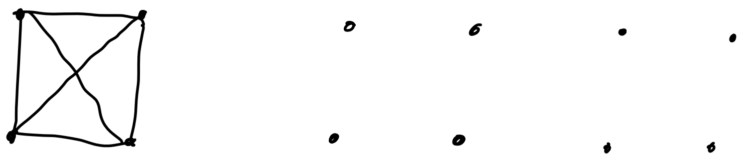
"G3COL has no worst-case poly(n) time alg"

but... there's a very easy alg for G3COL which succeeds on $\gg 1 - \frac{1}{n^{100}}$ frac. of all n-node graphs

/

alg: NO.





$\frac{1}{2^6} = \frac{1}{64}$

$\frac{n}{4}$ clusters;

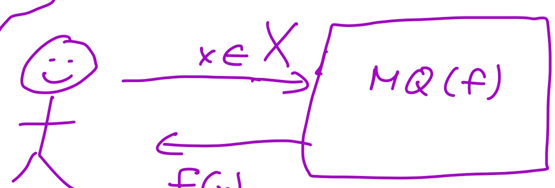
prob.(none gets all 4 edges) $\leq \left(\frac{63}{64}\right)^{\frac{n}{4}} = \frac{1}{2^{\Theta(n)}}$.

We'll make avg-case (strong) hardness assump. to get HoL.

Pseudorandomness + HoL

New learning setup: "membership queries"
"black-box"
"oracle access"

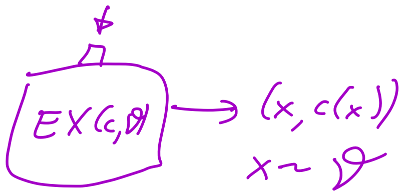
$\mathcal{D}$ = unif dist. $\{0,1\}^n$

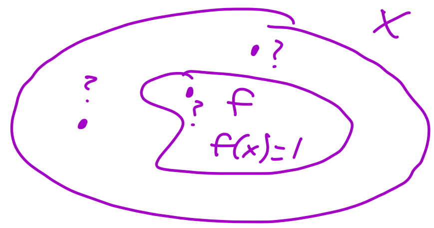


f = target function

$f \in \mathcal{C}$

A = alg; A^f: "A has MQ access to f"





(Truly) Random vs Pseudorandom functions

What's a "random Bool. fn" ?

$(X = \{0,1\}^n)$

$$\boxed{\mathcal{C}_{ALL} = \text{all } 2^{2^n} \text{ fns } f: \{0,1\}^n \to \{0,1\}}$$

"(Truly) Random Bool fn" : a $f \sim_{\text{unif}} \mathcal{C}_{ALL}$.

(Need to toss $\underline{2^n \text{ coins}}$ to pick such an f.)

What's a pseudorandom Bool. fn?
It's A uniform draw from a PRFF.

(n coin tosses needed!)

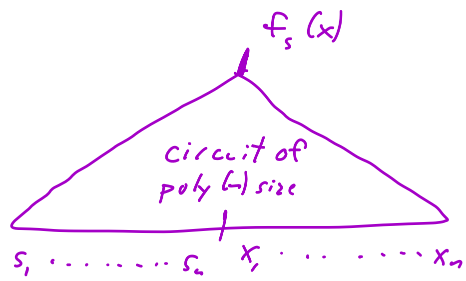
Def (PRFF) Let $\mathcal{F}$ be a set of 2^n Bool fns

$$\mathcal{F} = \{ f_s : s \in \{0,1\}^n \} \quad \text{each } f_s : \{0,1\}^n \to \{0,1\}$$
$$(s = \text{seed})$$

We say $\mathcal{F}$ is a pseudorandom function family (PRFF) if:

① (efficient computability)
Each f_s is poly(n)-time computable; in fact, there's a poly(n) time alg which, given s, x, outputs $f_s(x)$

($\mathcal{F} \subseteq$ class of all poly(n) size ckts)



② (indistinguishability, to poly(n)-time observers, from truly random fns)
Let DIST (distinguisher) be any poly(n)-time alg which gets oracle access to a Bool fn $\{0,1\}^n \to \{0,1\}$ + outputs either "PR" or "R". Then

$$\left| \Pr_{\substack{f \sim \mathcal{C}_{ALL} \\ \text{(truly random)}}}\left[\text{DIST}^{f} \text{ outputs "PR"}\right] - \Pr_{\substack{s \sim \{0,1\}^n \\ \text{(pseudo-random)}}}\left[\text{DIST}^{\overset{\in \mathcal{F}}{f_s}} \text{ outputs "PR"}\right] \right|$$

$< \frac{1}{p(n)}$ for all polynomials $p(n)$.

Major crypto hardness assumption:

$\exists$ PRFFs.

If one-way fns exist, .

If factoring is (avg-case) hard, $\exists$ PRFFs.

A PRFF is a hard-to-learn concept class:

Thm: Suppose $\mathcal{F}$ is a PRFF.
Then there is no poly(n)-time PAC learning alg A for $\mathcal{F}$, using any polynomially evaluatable $\mathcal{H}$, even if

i) only require alg to succeed under $\mathcal{U}$ = unif dist on $\{0,1\}^n$;

ii) alg gets MQ access to target fn.

$\boxed{MQ(f)}$ lets you sim. $\boxed{EX(f, \mathcal{U})}$

Idea: $\left.\begin{array}{l} A \text{ succeeds on } \mathcal{F} \\ A \text{ fails on } \mathcal{C}_{ALL} \end{array}\right\}$ distinguisher

Pf: Let A be eff PAC alg for $\mathcal{F}$ as in .

We can use A to get a distinguisher as follows:
Given oracle access to unknown c, $\leftarrow$ $c = f$, $f \in \mathcal{C}_{ALL}$
or
$c = f_s \sim \mathcal{F}$

- run A using MQ oracle, with $\varepsilon = 0.01$, $\delta = 0.01$

  get hyp $h : \{0,1\}^n \to \{0,1\}$ (poly(n)-time evaluatable)

- pick uniform $z \sim \{0,1\}^n$

  call $MQ(c)$ on z to get $c(z)$

  eval. $h(z)$ $\qquad h(z)$

- output "PR" iff $h(z) = c(z)$. (else output "R").

Claim: this viol. prop. 2 of PRFF def.

B/c:

- Suppose $c = f_s$, some $f_s \in \mathcal{F}$.

w.p. ≥ 0.99, h has error ≤ 0.01; hence

overall w.p. $\geq \boxed{0.98}$ $h(z) = c(z)$ & output "PR".

- Suppose $c = f$, $f \sim \mathcal{C}_{ALL}$.

Unless z is a point queried in execution of A (prob. $\leq \frac{\text{poly}(n)}{2^n} \ll .01$), $h(z) = \underbrace{c(z)}_{\text{coin toss!}}$

w.p. $= \frac{1}{2}$.

So, overall prob. $h(z) = c(z) \leq \frac{1}{2} + \ldots \leq \boxed{0.51.}$

Contradiction! So couldn't have had such an A.
((2) of PRFF def)

So, there are computational barriers to eff. learning!

EASY →   ← HARD

$\mathcal{C}$ = all conjunctions $\mathcal{C}$ = all LTFs   ? $\mathcal{C}_{\star}$ = all poly(n)-size ckts

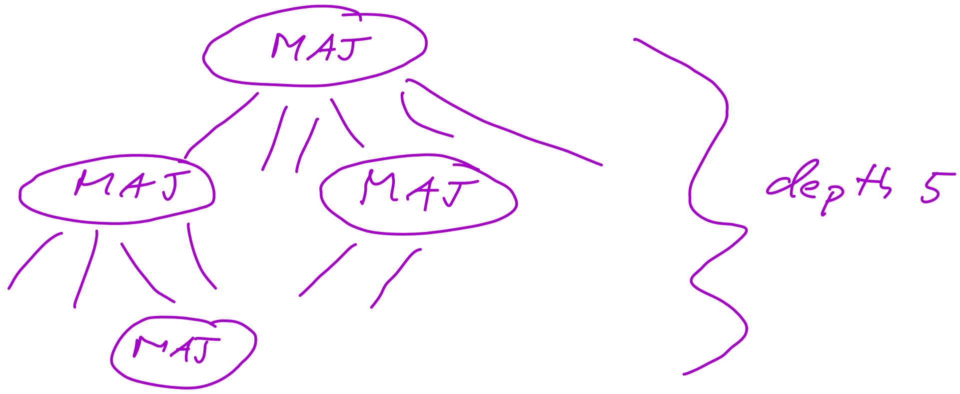
simpler functions   more cx functions

circuit complexity measures "simplicity" of fns...

One way to try to show that "simpler" classes than $\mathcal{C}_{\star}$:

design simple PRFFs.

There are PRFFs computable as depth-5 MAJ ckts.



Another way to get HoL:
diff arguments/crypto. objects give us HoL?

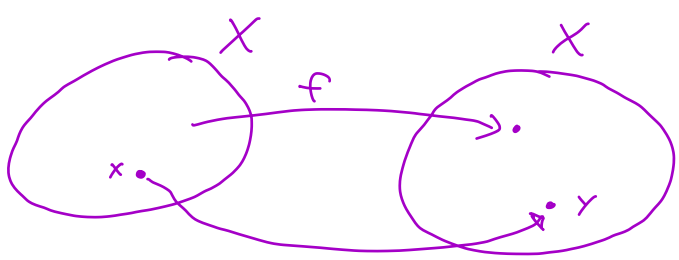
Yes

HoL based on Public-Key Crypto/Trapdoor 1-way Permutations

Def: A permutation of finite set X:

bijection $X \longrightarrow X$

one-to-one + onto



$\forall y \in X$, $\exists$ unique x s.t. $f(x) = y$

i.e. $f^{-1}: X \to X$ is well-defined

(Informal) "one-way permutation" on $X = \{0,1\}^n$

a perm. $f : \{0,1\}^n \to \{0,1\}^n$ s.t.

- there's a poly(n) time alg. to <u>compute</u> f, but
- any poly(n) time alg. can't compute f correctly even on a $\frac{1}{\text{poly}(n)}$ frac. of inputs.

Next time:

<u>trap-door</u> one-way permutations

+ HoL