# Lecture 9

More applications of pairwise independence

- Interactive proofs.

  Public coins vs. private coins

Derandomization via method of conditional expectations

Another setting in which k-wise independence is useful:

## Interactive Proofs

NP = all decision problems for which "Yes" answers can be verified in polytime by a deterministic TM ("verifier")

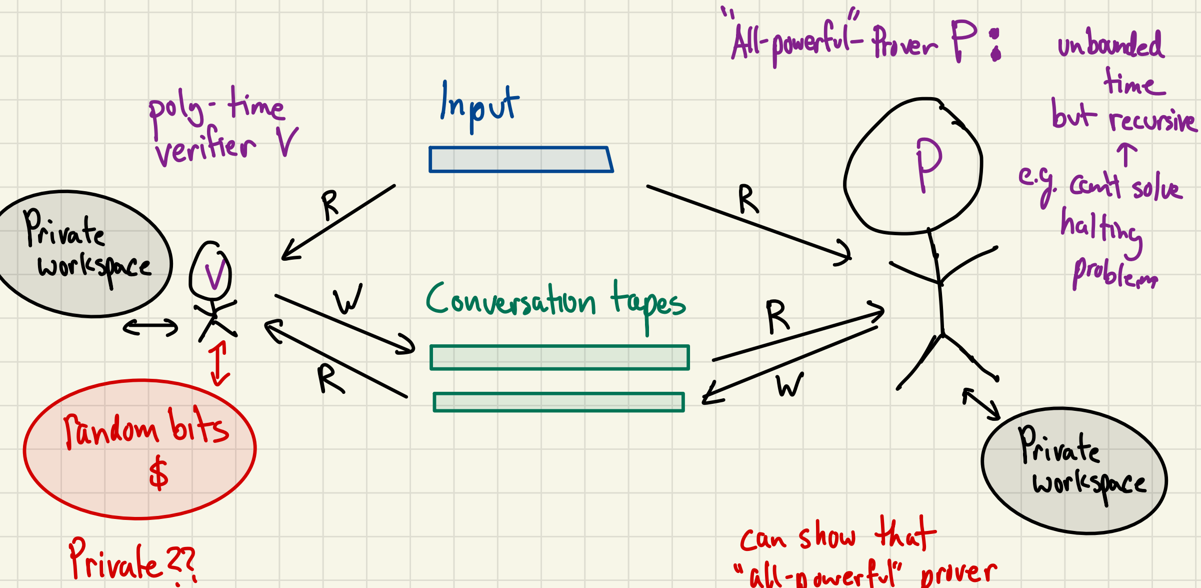
IP:

generalization of NP

Short proofs $\Longrightarrow$ short interactive proofs

"conversations that convince"

# IP Model




can show that "all-powerful" prover doesn't need random coins
(i.e. anything it can do with coins, it can also do without coins)

**def.** [Goldwasser Micali Rackoff]

An Interactive Proof System (IPS) for language $L$ is protocol s.t.

- if $x \in L$ & both $V, P$ follow protocol then

$$\Pr_{V\text{'s coins}}[V \text{ accepts } x] \geq 2/3$$

- if $x \notin L$ & $V$ follows protocol then (no matter what $P$ does)

$$\Pr_{V\text{'s coins}}[V \text{ rejects } x] \geq 2/3$$

**Thm** [Goldwasser Sipser]

$$IP_{\substack{\text{private}\\\text{coins}}} = IP_{\substack{\text{public}\\\text{coins}}}$$

GS's Answer: NO!
anything that has protocol with private coins also has (possibly different) protocol with public coins.

today we will see a building block for theorem:

Informally:

- Given set $S$ s.t. $S \in IP$ ← interesting even if $S \in P$
- Protocol in which P can convince V that size of set $S$ is "big"

Let $S_\phi \equiv \{x \mid x \text{ satisfies formula } \phi\}$ } can be replaced by any $L \in IP$

(note $S_\phi \in P$)

**Claim** $\exists$ protocol s.t. on input $\phi$

- if $|S_\phi| > k$ & if V, P follow protocol
  then $\Pr[V \text{ accepts}] \geq 2/3$
- if $|S_\phi| < \frac{k}{\Delta}$ & if V follows protocol ← even if P cheats!
  then $\Pr[V \text{ accepts}] < 1/3$

for now assume $\Delta = 4$

Note:

Can use protocol to show that # random strings which cause algorithm $\mathcal{A}$ to accept on input $x \geq 2/3$

First idea Random Sampling

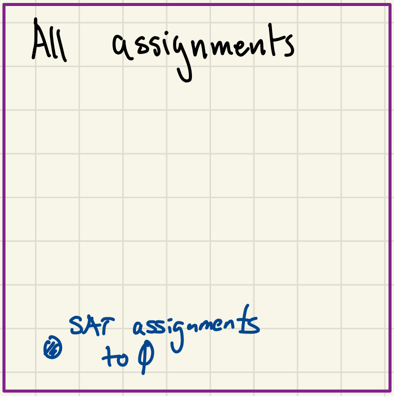
Repeat ? times:

V picks random assignment x
& evaluates $\varphi(x)$

Output $\dfrac{\text{\# satisfying } x\text{'s}}{\text{total \# repetitions}}$

how many repetitions?

$\Omega\left(\dfrac{\text{\# total assignments}}{\text{\# satisfying assignments}}\right)$ could be $\Omega(2^n)$




Problem: what if $S_\phi$ is small?

## Fix: Universal hashing

Recall:

Family of fctns $\mathcal{H} = \{h_1, h_2 \dots\}$

for $h_i : [N] \to [M]$ is

"pairwise independent" if

when $h \in_u \mathcal{H}$

(1) $\forall x \in [N], \quad h(x) \in_u [M]$ — any locn $x$ is mapped uniformly

(2) $\forall x_1 \neq x_2 \in [N], \quad (h_1(x), h_2(x)) \in_u [M]^2$ — any pair of locns $x_1 \neq x_2$ mapped uniformly & independently

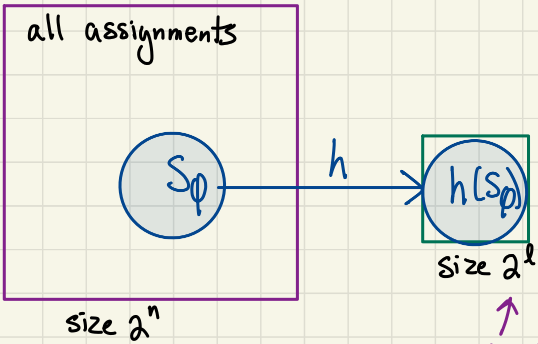
equivalently:

$\forall x_1 \neq x_2 \in [N]$

$\forall y_1, y_2 \in [M]$

$$\Pr_{h \in \mathcal{H}}[h(x_1) = y_1 \ \& \ h(x_2) = y_2] = \frac{1}{M^2}$$

## How does it help?




pick $\ell$ s.t.

$$2^\ell > k \geq \frac{2^\ell}{2}$$

Need:

1. $|h(S_\varphi)| \approx |S_\varphi|$
2. $h$ computable in poly time

<u>idea</u>

- Clearly $|h(S_\varphi)| \leq |S_\varphi|$
- hopefully $|h(S_\varphi)|$ is not too much smaller than $|S_\varphi|$

this is a very nice property of p.i. hash fctns.

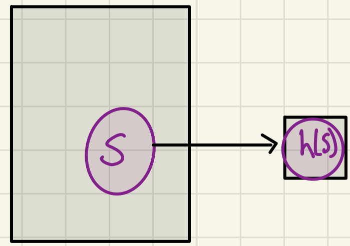
(we will show that whp $|h(S_\varphi)| > \frac{|S_\varphi|}{\Delta}$)

$\Rightarrow$ if $\ell$ s.t. $2^\ell$ is roughly $|h(S_\varphi)|$

then most of $1..2^\ell$ gets mapped to by $h(S_\varphi)$

(uses that $\mathcal{H}$ is p.i.)

# A comment about p.i. hash fctns




typical use:

- map set $S$ into smaller "space"
- good for storage, reducing size of "names" of elements ...
- need property of "few collisions"

  since collisions cause problems, so need to minimize
  (e.g. in hash tables, collisions $\Rightarrow$ chaining length)

- here "few collisions" $\Rightarrow$ $|h(S)|$ is not too much smaller than $|S|$

Why is that good?

- pick any pt in range, say $0^\ell$
- if $h(S)$ big, it will probably hit $0^\ell$

  uses that $h(x)$ is unif dist

**Protocol:** for distinguishing set of size $K$
from set of size $K/\Delta$

Given $\mathcal{H}$ (p.i. fctns mapping $\{0,1\}^n \to \{0,1\}^\ell$)

1. $V$ picks $h \in_R \mathcal{H}$
2. $V \to P$: $h$
3. $P \to V$: $x \in S_\varphi$ s.t. $h(x) = 0^\ell$
4. $V$ accepts iff $x \in S_\varphi$

**Idea:** hope: $h(S_\varphi)$ fills "random" portion of range, so can distinguish $|h(S_\varphi)|$ large or small.

Case 1 $|S_\varphi| > K$:

hopefully $|h(S_\varphi)| \approx K$ so $0^\ell$ is "hit" with reasonable ($\geq \frac{1}{2}$?) probability.
Then all-powerful $P$ can find preimage in $S_\varphi$

Case 2 $|S_\varphi| < \frac{K}{\Delta}$:

$|h(S_\varphi)| < K/\Delta$ so less likely $0^\ell$ hit.
if not hit, $P$ can't find preimage.
If $P$ sends $V$ a fake preimage, $V$ will detect.

if $U = S_\phi$ & $h$ maps $S_\phi$ 1-1 (which is unlikely) then $a$ is fraction mapped to

**Lemma** $\mathcal{H}$ is p.i., $U \subseteq \{0,1\}^n$, $a = \frac{|U|}{2^\ell}$

then $a - \frac{a^2}{2} \leq \Pr_h[0^\ell \in h(U)] \leq a$

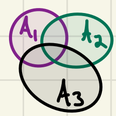
**Proof**

RHS:

$\forall x \ \Pr_{h \in H}[0^\ell = h(x)] = 2^{-\ell}$ since $H$ is p.i.

so $\Pr_h[0^\ell \in h(U)] \underset{\uparrow \text{ union bnd}}{\leq} \sum_{x \in U} \Pr[0^\ell = h(x)] = \frac{|U|}{2^\ell} = a$

LHS: $\Pr[\cup A_i] \underset{\uparrow \text{ inclusion exclusion}}{\geq} \sum_i \Pr[A_i] - \sum_{i \neq j} \Pr[A_i \cap A_j]$




$$\Pr_{h \in \mathcal{H}}[0^\ell \in h(U)] \geq \sum_{x \in U} \underbrace{\Pr[0^\ell = h(x)]}_{2^{-\ell}} - \sum_{x \neq y \in U} \underbrace{\Pr[0^\ell = h(x) = h(y)]}_{2^{-2\ell} \text{ if pairwise indep}}$$

$$= \frac{|U|}{2^\ell} - \binom{|U|}{2} \frac{1}{2^{2\ell}} \geq \frac{|U|}{2^\ell} - \frac{|U|^2}{2} \cdot \frac{1}{2^{2\ell}}$$

$$\geq a - a^2/2$$

$\blacksquare$

# Finishing up:

Pick $\ell$ s.t. $2^{\ell-1} \leq k \leq 2^{\ell}$

let $a = \dfrac{|S_\varphi|}{2^\ell}$

If $|S_\varphi| > k$ then $a \geq 1/2$

so $\Pr[0^\ell \in h(S_\varphi)] \geq a - \dfrac{a^2}{2} \geq 3/8$

if $|S_\varphi| < k/\Delta$ then $a < \dfrac{k/\Delta}{2^\ell} < \dfrac{1}{\Delta}$ ← assumption on $k$

so $\Pr[0^\ell \in h(S_\varphi)] \leq a < \dfrac{1}{\Delta}$

e.g. picking $\Delta = 4$ gives $\leq 1/4$

If repeat $O(\log 1/\beta)$ times,

Chernoff $\Rightarrow$ with prob $\geq 1-\beta$

if $|S_\varphi| \geq k$ then $P$ is successful $\geq 3/8 - o(1)$ of repetitions

if $|S_\varphi| \leq \frac{k}{\Delta}$ then $P$ is successful $\leq 1/4 + o(1)$ of repetitions

Comments

- Can improve so $\delta = 1 - \varepsilon$ (how??)

- Can use same idea to prove

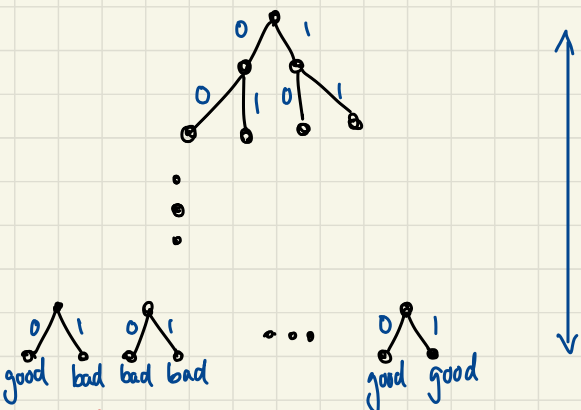
$$IP_{\substack{\text{private}\\ \text{coins}}} = IP_{\substack{\text{public}\\ \text{coins}}}$$

argue that l.b. protocol can be used to show size of accept region probability mass is large.

(need that V can verify a conversation / random coin flips transcripts falls into accept region).

# Derandomization via the method of conditional expectations

<u>idea</u>: view coin tosses of algorithm as path down tree of depth $m$
$\shortparallel$
\# coin tosses




depth $m$ = \# coin tosses

alternatively: $c_{00\ldots0}$ $c_{00\ldots1}$ $c_{00\ldots10}$ $c_{00\ldots11}$ $c_{11\ldots10}$ $c_{11\ldots1}$
cut values

good = correct / reach witness / good approximation / pass ...

good randomized algorithm $\Rightarrow$ most leaves good

<u>idea</u>: find a good path to leaf "bit-by-bit"

more formally:

Fix randomized algorithm $\mathcal{A}$

input $x$
$m = \#$ random bits used by $\mathcal{A}$ on $x$

for $1 \le i \le m$ + $r_1 \dots r_i \in \{0,1\}$

let $p(r_1 \dots r_i)$ = fraction of continuations that end in "good" leaf (good approx)

set 1st $i$ bits to $r_1 \dots r_i$
pick $(i+1)^{st}$ to $m^{th}$ bits randomly

$e(r_1 \dots r_i)$ = average cut value if set first $i$ nodes to $r_1 \dots r_i$

$$p(r_1 \cdots r_i) = \frac{1}{2} \cdot p(r_1 \cdots r_i 0) + \frac{1}{2} \cdot p(r_1 \cdots r_i 1)$$

by averaging, $\exists$ setting of $r_{i+1}$ to 0 or 1
s.t. $p(r_1 \dots r_{i+1}) \ge p(r_1 \dots r_i)$

can we figure out which one?

if $\quad p(r_1 \dots r_{i+1}) \geq p(r_1 \dots r_i) \quad \forall i$

then $\quad p(r_1 \dots r_m) \geq p(r_1 \dots r_{m-1}) \geq \dots \geq p(r_1) \geq$ fraction of good paths

$\geq \underbrace{2/3}$

arbitrary const $\geq 1/2$

this is a leaf so value is 1 or 0 but if $\geq 2/3$ it must be 1

<u>main issue</u>:

how do we choose best $r_{i+1}$ setting at each step?

<u>Example</u> Max cut (second way to derandomize)

recall algorithm:

flip $n$ coins $r_1 \dots r_n$

put node $i$ in $S$ if $r_i = 0$ & $T$ if $r_i = 1$

Output $S, T$

Recall from lecture 7:

Analysis:

$u,v$ on opposite sides of cut

Let $\mathbb{1}_{u,v} = \begin{cases} 1 & \text{if } r_u \neq r_v \\ 0 & \text{o.w.} \end{cases}$

$$\text{Cut size} = \sum_{(u,v)\in E} \mathbb{1}_{u,v}$$

$$E[\text{cut size}] = E\Big[\sum_{(u,v)\in E} \mathbb{1}_{u,v}\Big]$$

$$= \sum_{(u,v)\in E} E[\mathbb{1}_{u,v}] = \sum_{(u,v)\in E} \Pr[\mathbb{1}_{u,v} = 1]$$

$$= \sum_{(u,v)\in E} \Pr\big[(r_u = 1 \,\&\, r_v = 0) \text{ or } (r_u = 0 \,\&\, r_v = 1)\big]$$

$$= |E| \cdot \left[\tfrac{1}{4} + \tfrac{1}{4}\right] = \frac{|E|}{2}$$

So expect $\frac{1}{2}$ the edges to cross cut!

Note: $\underbrace{E[\text{cut size}]}_{\text{average cut size produced by algorithm}} = \frac{|E|}{2} \Rightarrow \exists \underbrace{\text{cut}}_{\text{must be one that is at least as big as the average}} \text{ of size } \frac{|E|}{2}$

derandomization:

$$e(r_1 \cdots r_i) = E_{R_{i+1} \cdots R_n}\left[\, |cut(S,T)| \text{ given } r_1 \cdots r_i \text{ choices made} \right]$$

$$e(\text{no choices fixed yet}) \geq \frac{|E|}{2} \quad \text{(previous lecture)}$$

how do we calculate $e(r_1 \cdots r_{i+1})$?

Let

$S_{i+1} = \{\text{nodes } j \mid j \leq i+1,\ r_j = 0\}$

$T_{i+1} = \{\text{nodes } j \mid j \leq i+1,\ r_j = 1\}$

$\Big\}$ S & T so far

$U_{i+1} = \{\text{nodes } j \mid j \geq i+2 \text{ \& } j \leq n\}$ $\Big\}$ undecided

So:

main insight #1

fact $e(r_1 \cdots r_{i+1}) = (\#\text{ edges between } S_{i+1} \text{ \& } T_{i+1})$

$+ \frac{1}{2}(\#\text{ edges touching } U_{i+1})$

(Follows from same reasoning as last lecture)

main insight #2

Note: don't need to calculate $e(r_1 \cdots r_{i+1})$

just need to figure out which is bigger

$e(r_1 \cdots r_i 0)$ or $e(r_1 \cdots r_i 1)$

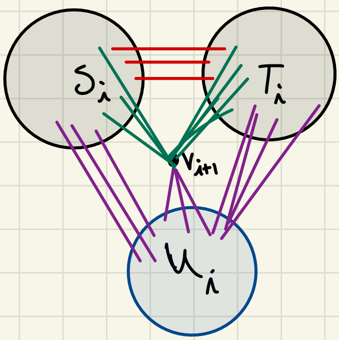
how do we do this?

# edges between $S_{i+1}$ + $T_{i+1}$ same for both

$U_i$ is same for both

$U_{i+1}$ differs only on edges to node i+1

to maximize this, place node i+1 to maximize cut size:




compare
#edges between $v_{i+1}$ + $S_i$
vs. " " " " $T_i$

$\Rightarrow$ can deterministically pick which choice gives bigger # edges touching $U_{i+1}$

$\Rightarrow$ if do this for each $i$, get solution which is $\geq$ expected value in deterministic way

yields:

## Greedy Algorithm

1) $S \leftarrow \varphi$, $T \leftarrow \varphi$

2) For $i = 0 \ldots n-1$

place $v_i$ in $S$ if #edges between $v_i$ + $T$
$\geq$ " " " " $S$

else place $v_i$ in $T$