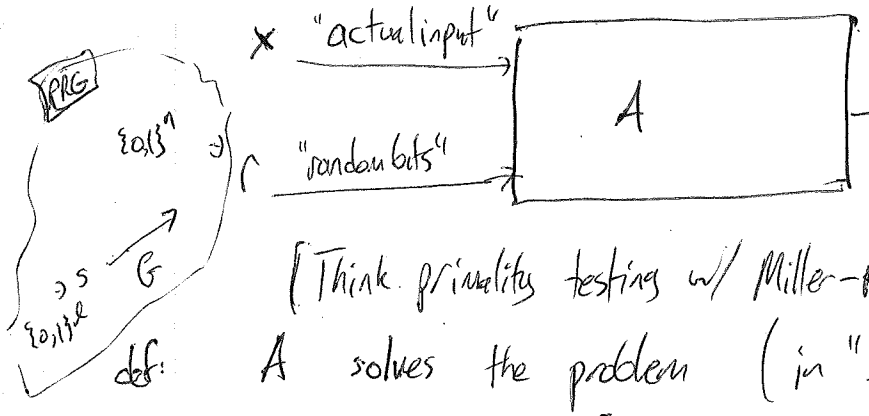
Lecture 11 - Derandomization

[Scribe] [Refs: Dieter's course, Salil's book]

[First, there's nothing wrong with randomized algs. "In practice" we have "random bits". If you make an eff. rand. alg for a problem — great! You're done IMO.
Still: • theoretically, don't really have rand bits; physics/philosophy
• interesting complexity-theoretic q's
⊛ Derand ≡ want "obj" with some "quality"
- have a huge search space where most objs have that quality
- but want a much smaller search space
↳ efficiency
Derand ideas often yield "efficient" algs/constructs where none known before]

[Terminology]: [Rand alg picture]



[For simplicity we focus on decision problems]

[Think primality testing w/ Miller-Rabin: x is the # to be tested]

def: A solves the problem (in "BPP") ("2-sided error") if

$$\forall x, \quad \Pr_{r}[A(x,r) \text{ correct}] \geq \frac{3}{4}$$

[← orbit, yes, but if you want more conf, run several times, take maj vote]

[Can we replace r by "pseudorand. bits"? How "good" must they be? Good enough so A_x's behavior doesn't change. Allow a few "truly" rand bits. Say $l=n$ for $A_x(r)$]

def: Let $\mathcal{C}$ be a class of fcns $f: \{0,1\}^n \to \{0,1\}$.
$G: \{0,1\}^l \to \{0,1\}^n$ is an ε-pseudorandom generator (PRG) for $\mathcal{C}$ with seed length l ($<n$) if

[BM, Y '82]

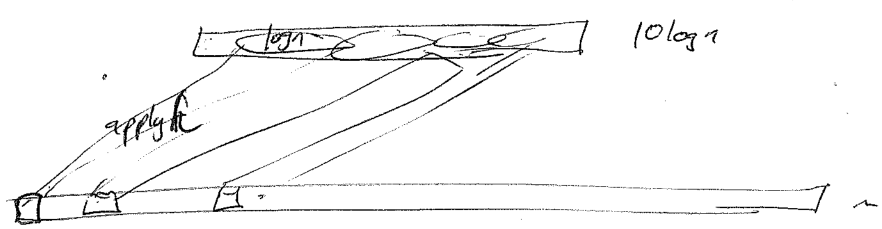
$$\forall f \in \mathcal{C} \quad \left| \Pr_{s \sim \{0,1\}^\ell}[f(G(s)) = 1] - \Pr_{r \sim \{0,1\}^n}[f(r) = 1] \right| \le \varepsilon.$$

{ "G ε-fools $\mathcal{C}$" Typically, want $G(s)$ computable in poly(n) time.

1 intuition: $\mathcal{C}$ is a class of "statistical tests" f, trying to check randomness of input.

Say A runs in $O(n^{10})$ time, uses n bits of randomness
$\forall x$, $A_x : \{0,1\}^n \to \{0,1\}$ computed by ckt of size $\tilde{O}(n^{10})$
Let $\mathcal{C} = \{f : \{0,1\}^n \to \{0,1\}$ computable by $\tilde{O}(n^{11})$-size ckts$\}$
If G .01-fools $\mathcal{C}$ w/ seed length ℓ,
$\Rightarrow$ can solve A's problem w/ just ℓ rand bits.
If $\ell = O(\log n)$ can solve A's prob. deterministically in poly(n) time! [Enumerate all seeds, one answer occurs for > .51 of them]

hardness vs. randomness idea: [Y; NW]: Let G do:



where $h : \{0,1\}^{\log n} \to \{0,1\}$ is "super-hard to compute"
→ computable in $2^{\log n}$ time, but any alg. using much less does terribly.

Impagliazzo-Wigderson '97 Theorem: Suppose $\exists (h_n)_n$
$h_n : \{0,1\}^m \to \{0,1\}$ s.t. • computable in time 2^{100m} by T.M.
• not computable by $2^{.001m}$-size circuit [$\forall n \ge n_0$]
[Stronger than P≠NP, sim. to ETH] (Pretty believable! eg SAT) Then BPP=P. [Basically, any rand. poly-time alg can be derand.]

[So, in some sense, if you believe ∃ explicit hard probs, you believe BPP = P. However, let's now talk uncondit. results. One famous one: instead of considering poly-time, say log space...]

Thm: [Nisan '92] ∃ PRG G which ε-fools randomized algs using
n rand. bits, space $S = S(n) \geq \log n$
where $\varepsilon = 2^{-S}$, seed length $\ell = O(S \log n)$.
G computable in $O(\ell)$ space ($\Rightarrow 2^{O(\ell)}$ time)

e.g.: $S = O(\log n)$: $\varepsilon = \frac{1}{\text{poly}(n)}$ ☺, $\ell = O(\log^2 n)$ ☹, time $n^{O(\log n)}$.

pf 1: pairwise indep... pf 2 [INW]: expanders... [we'll see both techniques.]

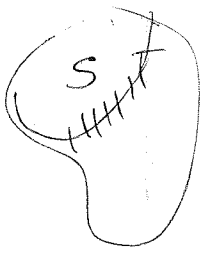
[Not all derand. is based on PRGs. Notice that PRGs hamstring you somewhat: can't look at the input!! Here's a very simple technique that uses knowledge of the input....]

Method of Conditional Expectations [An illustration.]

Max-Cut: Given $G = (V, E)$, n, m, find $S \subseteq V$ to maximize $|E(S, \bar{S})|$

$\Leftrightarrow$ find $f: V \to \{0,1\}$ to max $\text{val}(f) = \sum_{(u,v) \in E} 1[f(u) \neq f(v)]$

[1 if ≠, count 0 else. NB: could put in edge wts]



Rand alg: [Choose f: for $u = 1 \ldots n$ randomly [] set $f(u) = r_u$] Given rand $r \in \{0,1\}^n$, [uses n rand. bits] r_u, $u = 1 \ldots n$.

Analysis: $\mathop{E}_r[\text{val}(f)] = \sum_{(u,v) \in E} E[1[f(u) \neq f(v)]]$

$= \sum_{(u,v) \in E} \Pr[f(u) \neq f(v)] \geq \frac{1}{2}$

$= \frac{1}{2} m$. [Not bad. "$\frac{1}{2}$-approx."]

Aside: [Prob. method] $\forall G$, $\exists$ S cutting $\geq \frac{1}{2}$ edges.

[[Meth. of condit. expects uses the fact that bits are processed 1 at a time, and for each, you can "compute how well you're doing"]].

Derand: for $u = 1 \dots n$

- compute $E[\text{val}(f) \mid \text{choices so far} \ \& \ f(u) = b]$ for $b = 0, 1$ — if you were to do remaining choices randomly
- double det'ly, effic'ly
- choose $f(u) = b$ for whichever is bigger [[ties → arbitrary]]

[[Note: Though alg refers to randomness, it's determ.!]]

Analysis:

Invariant: after t iterations, $E[\text{val}(f) \mid \text{choices so far}] \geq \frac{1}{2} m$.

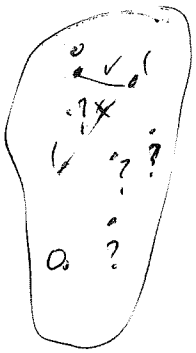
True at start? ✓

Maintained?

$$= \frac{1}{2} E[\text{val}(f) \mid \text{choices} \ \& \ f(t+1) = 0] + \frac{1}{2} E[\text{val}(f) \mid f(t+1) = 1]$$

← at least one $\geq \frac{1}{2} m$.

$\therefore E[\text{val}(f) \mid \text{all } n \text{ choices}] \geq \frac{1}{2} m$. □



rem: This alg. also describable as the greedy alg. [[!]]

[[Doesn't happen in more sophisticated cases]].

k-wise independence

def: $G : \{0,1\}^l \to \{0,1\}^n$ is pairwise indep if $\forall \ i \neq j \in [n]$

$$\Pr_{s \in \{0,1\}^l}[(G(s)_i, G(s)_j) = (0,0)] = \frac{1}{4}$$
$$= (0,1) = \frac{1}{4}$$
$$(1,1) = \frac{1}{4}.$$

I.e., for rand s, the r.v.'s $G(s)_i, G(s)_j$ are indep.

[[It's like you O-fool the class of all tests that only look @ 2 bits]].

- k-wise indep: $\forall$ distinct $i_1, \dots i_k \in [n]$
  $\Pr[(G(s)_{i_1}, \dots, G(s)_{i_k}) = (b_1, \dots, b_k)] = 2^{-k} \quad \forall b_1 \dots b_k \in \{0,1\}$
- Analogous defn for $G: \Sigma^\ell \to \Sigma^n$, $|\Sigma| = q \geq 2$.

thm: [ABI 85] $\forall k \leq n$, prime power q, $\exists$ poly(n)-computable k-wise indep. gen. with $\ell = \lfloor \frac{k}{2} \rfloor \log_q n + O(1)$

eg: $q=2$, $k = 2$ or 4 or 10: $O(\log n)$ seed length [[Great, can enum over in poly(n) time]]

[[What are they good for?]]

eg 1: Alternate Max-Cut derand:
- We only used $\forall u,v \in V$, $\Pr[r_u \neq r_v] = \frac{1}{2}$.
- Satisfied if r is ~~pairwise~~ gend pairwise-indep!
  → Enum. all $O(\log n)$ seeds.

eg 2: Say your rand algs. analysis needs the fact
$\Pr_r[r_1 + \dots + r_n \geq \frac{n}{2} + t\frac{\sqrt{n}}{2}] \leq \frac{1}{t^2}$

[[Chernoff tells you sthg much stronger, but often enough.]]
Provable by Chebyshev.
↳ Only req's pairwise indep. (Lec 3)

Lec 3: $\leq \frac{3}{t^4}$ by "4th mom. method,"
only needs 4-wise indep.

$q=2$, $k=2$:

$$G: \quad x \in \{0,1\}^\ell \longrightarrow \{0,1\}^{n = 2^\ell - 1}$$

$$x \longmapsto \begin{bmatrix} 0\,0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,1\,1 \\ \vdots \\ 1\,1\,1\,1\,1\,1 \end{bmatrix} \begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix}$$

[[Look familiar? Hadamard code (except no 1st 6-7)]]

Pairwise indep.? Need $\forall$ rows h, h':

$$\begin{bmatrix} \text{—} & h & \text{—} \\ \text{—} & h' & \text{—} \end{bmatrix} \begin{bmatrix} \\ x \\ \\ \end{bmatrix} : \begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$$

unif. on $\mathbb{F}_2^2$ when unif on $\mathbb{F}_2^\ell$ ✓

[[Several ways to see it. Certainly those rows must be lin indep else you'll be concd on a subsp OTOH iff they're lin. indep, it will be true. ...]]

ex:

$$x \in \mathbb{F}_q^\ell \longmapsto Gx \in \mathbb{F}_q^n \quad \text{is } k\text{-wise indep}$$

$\Leftrightarrow$ any k rows of G lin indep.
$\Leftrightarrow$ no k cols of G^T lin dep
$\Leftrightarrow$ dual code of G has min dist $\geq k+1$.

[[We used: [[Hamming code has min dist 3]].

ex. ABI Thm $\Leftarrow$ Reed-Solomon codes
with k instead of $\lfloor \frac{k}{2} \rfloor$

[to get $\frac{k}{2}$ which is sharp, use a tweak known as BCH codes]]

ε-biased generators

Def: $G: \mathbb{F}_2^\ell \to \mathbb{F}_2^n$ is an ε-biased generator if [it fools all deg-1/linear f]

$\forall w \in \mathbb{F}_2^n, w \neq 0$, $\Pr_{s \sim \mathbb{F}_2^\ell}[w \cdot G(s) = 1] \in [\frac{1}{2} - \frac{\varepsilon}{2}, \frac{1}{2} + \frac{\varepsilon}{2}]$

$\left| \cdot - \Pr_{x \sim \mathbb{F}_2^n}[w \cdot x = 1] \right| \leq \frac{\varepsilon}{2}$ ↘ $1/2$.

Thm: [NW93] $\ell = O(\log \frac{n}{\varepsilon})$ achievable if G poly-time comp'ble

(*) [AGHP92] $\ell = 2\log(\frac{n}{\varepsilon}) + O(1)$, $O(n^2/\varepsilon^2)$-time comp.

Also $\frac{1}{\varepsilon^3}$, $\frac{n^{5/4}}{\varepsilon^{7/2}}$

Eg. Applic: [Say your friend implements one of those crazy Matrix-Mult. algs running in time $n^{2.4}$ or whatever, but you're not sure did it correctly. Want to test his alg. In subcubic time, of course! For simplicity, assume over $\mathbb{F}_2$.]

Input: $A, B, C \in \mathbb{F}_2^{n \times n}$. Q: Is $AB = C$?

Goal: $O(n^2)$ [input-size] time alg.

Idea: Choose $y \sim \mathbb{F}_2^n$ uniformly, check if $(AB)y = Cy$

$A(By)$ — $O(n^2)$ — $O(n^2)$; Cy — $O(n^2)$

Analysis: $AB = C \Rightarrow \Pr[\text{equal}] = 1$.

$AB \neq C \Rightarrow D = AB - C$ has ≥ 1 nonzero row,

[— d —]

$\Rightarrow \Pr[d \cdot y = 1] = 1/2$.

$\Rightarrow ABy - Cy \neq 0$

[Can repeat w/ several y to gain high confidence]

Uses $O(n^2)$ time, $O(n)$ rand bits.

If y is output of a .1-biased gen, Pr still $\geq .45$.

$\rightarrow$ $O(n^2)$ time, $O(\log n)$ bits using [AGHP].

How to get a good ε-biased gen? Again, a coding connection!

Say G is ε-biased.
Let M be this $2^\ell \times n$ mtx over $\mathbb{F}_2$:

$$M = \begin{bmatrix} G(000\cdots0) \\ G(0\cdots01) \\ \vdots \\ G(111\cdots1) \end{bmatrix}$$

By def: $\forall w \in \mathbb{F}_2^n$, $w \neq 0$,
if $s \sim \mathbb{F}_2^\ell$,
$\Pr(G(s)\cdot w = 1] \in [\frac{1}{2} \pm \frac{\varepsilon}{2}]$

$\Leftrightarrow$ Mw has rel. Ham wt, in $[\frac{1}{2} \pm \frac{\varepsilon}{2}]$

$\Leftrightarrow$ M is generator of a code with min. (rel.) distance $\geq \frac{1}{2} - \frac{\varepsilon}{2}$ and max (rel.) distance $\frac{1}{2} + \frac{\varepsilon}{2}$!

For PRG we want a fat matrix $\Leftrightarrow$ good rate

$[N, n, (\frac{1}{2} - \frac{\varepsilon}{2})N]_2$ code with all codewords of wt $\leq (\frac{1}{2} + \varepsilon)N$

$\Leftrightarrow$ ε-biased generator, $(\log N) \longrightarrow n$.

Recall Lec. 10: RS $\diamond$ Hadamard: $[q^2, \underbrace{\varepsilon q^2 \log q}_{n}, (\frac{1}{2} - \frac{\varepsilon}{2})q^2]_2$

$\left(\frac{n}{\varepsilon}\right)^2$

$\Leftrightarrow$ $2\log(\frac{n}{\varepsilon})$ seed len.

$\therefore$ all nonzero Had codewords have rel. Ham wt $= \frac{1}{2}$, RS $\diamond$ H has all $\leq \frac{1}{2}$.

□