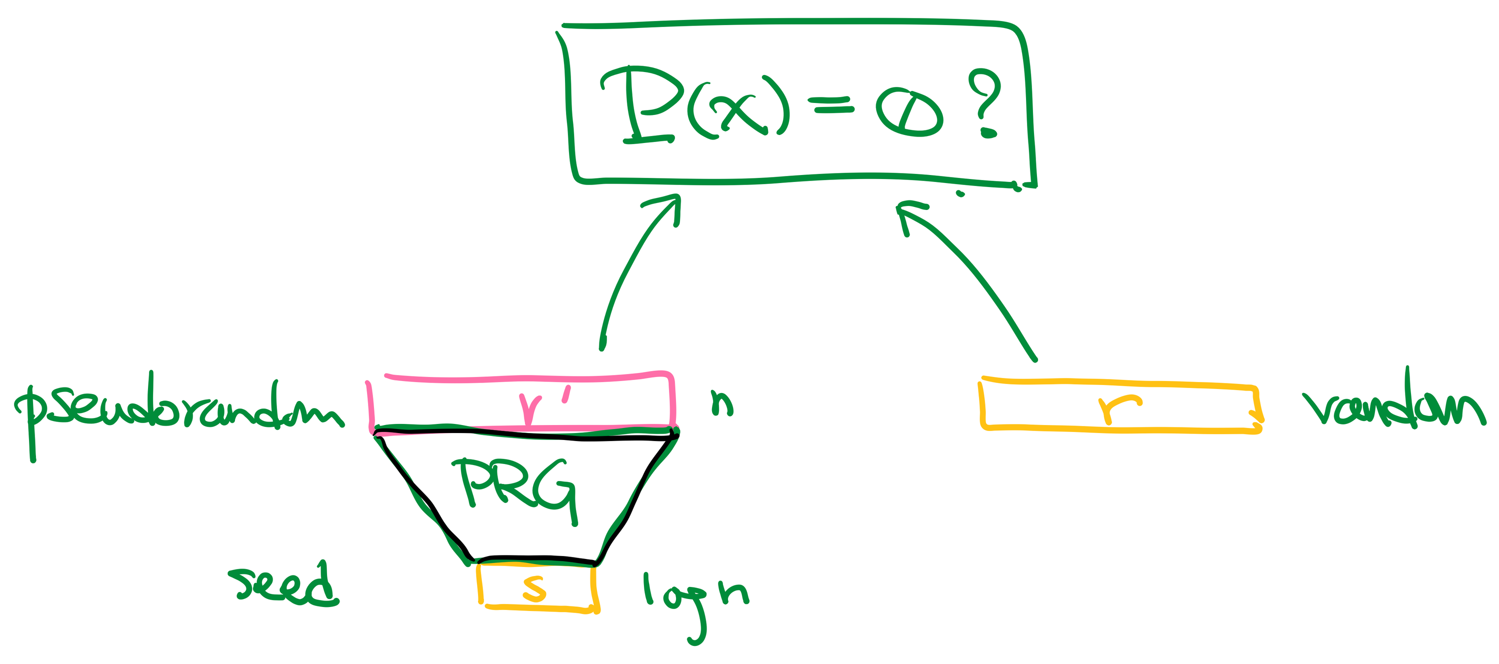
Pseudorandom generators and derandomization

2/24/21 10:47 AM

Question. Can we fake randomness?

	HC	TM	CS	ET	AP
From Me to Everyone: 0010010010111010100011111010001010010111 (20)	X	O	X	O	X
From Tracey Mills to Everyone: 0010100111010111001010011010001111100101 (21)	X	X	O	X	X
True random — From Connor Spencer to Everyone: 0110011010011000111111010100011010111111 (#1s : 24)	X	X	O	X	X
From Ethan Trep... to Everyone: 0111000010100111010100111010001011010100 (19)	X	O	O	X	X
From Aditya Prasad to Everyone: 0110101000110100101110100100100101011001 (19) — no long 0s or 1s.	X	X	O	X	X

Observation. If we can fake randomness, then BPP = P !



Pseudorandom generator (S,ε)-PRG.

$G:\{0,1\}^\ell \longrightarrow \{0,1\}^{S(\ell)}$

(1) stretching: s ℓ bits $\longmapsto$ $G(s)$ $S(\ell)$ bits.

(2) pseudorandom: $G(s)$ "looks" random to ANY "human" A.

$$\left| \Pr_{s\sim U_\ell}[A(G(s)) \text{ accepts}] - \Pr_{r\sim U_{S(\ell)}}[A(r) \text{ accepts}] \right| < \varepsilon$$

$\forall A:\{0,1\}^{S(\ell)} \to \{\text{acc}, \text{rej}\}$ efficiently checkable.

(3) efficient: G runs in $O(\text{poly}\, S(\ell))$ time.

Prop. If $\exists (2^{\varepsilon\ell}, 0.1)$-PRG, then BPP = P.

Pf. (sketch) Replace random n bits r used w/ $G(s)$. s $O(\log n)$ bits.

- BPP A accepts w.p. $\geq 3/4$
- BPP A won't notice r being replaced w.p. ≥ 0.9.
- New deterministic algorithm: enumerate $s\in\{0,1\}^{O(\log n)}$ and feed $G(s)$ to A.

Time: $O(2^{O(\log n)} \cdot \text{poly}\, n)$ ☒

Cor. If $\exists (\text{poly}\,\ell, 0.1)$-PRG. BPP $\subseteq$ SUBEXP $= \text{TIME}[2^{n^{o(1)}}]$

How to stretch random bits? s $\quad s \circ y$

Cool idea. HARD fcn $f(s)=y$

Nisan-Wigderson generator.

If $\exists$ fcn f (SAT) $: \{0,1\}^\ell \longrightarrow \{0,1\}$. $\text{SAT}(\phi) = \text{yes/no}$

- computable in $2^{O(\ell)}$ time.
- f can't be computed by $\text{poly}\, S(\ell)$-size circuits. (poly-time alg.) ($2^{\ell/100}$)

Then $\exists (S, 0.1)$-PRG.

Pf. (stretchy sketch) Worst-case hardness $\to$ Avg-case hardness

error-correcting code. '50 – [RM'54] [Yao'82][I'95] [IW'97]

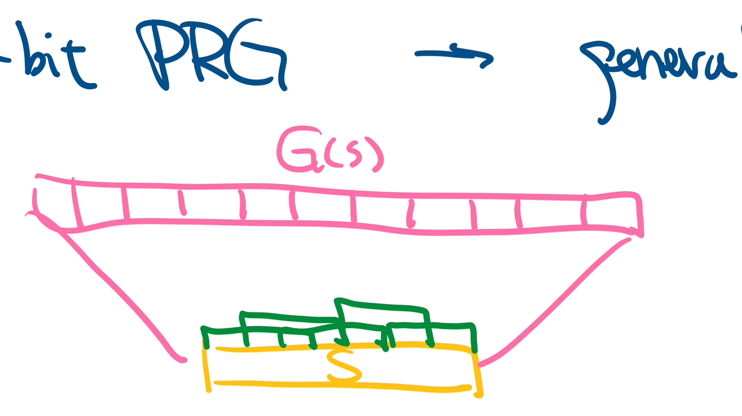
Avg-case-hard fcn $\to$ $s\circ f(s)$ pseudorandom.

Yao Theorem / hybrid argument. [Yao'82]

one-bit PRG $\to$ general PRG

combinatorial design [NW'88]

pseudorandom objects: expanders.



☒

Cor. If $\exists$ fcn $f\in \text{TIME}[2^{O(n)}]$, not solvable by ~~$2^{n/100}$-size~~ poly-size circuits

Then BPP = ~~P~~ SUBEXP

Further results:

[IW'98] BPP $\subseteq$ i.o.-SUBEXP unless BPP = EXP.

[KI'04] PIT $\in$ P $\Rightarrow$ some problems are hard.

What have we learned?

Hardness $\Leftrightarrow$ Derandomization

Either you believe some problems are hard, or random. poly-time algorithms really need dices.