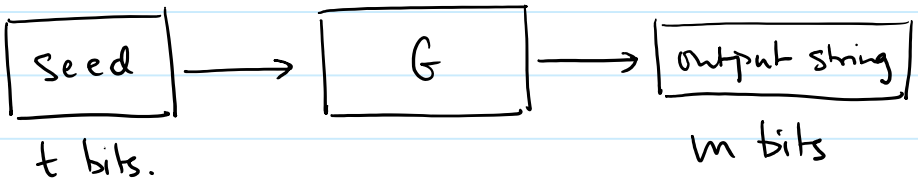
Thursday, April 26, 2018 8:26 PM

Next Goal: de-randomize BPP using pseudo-random generators. ↳ Simulate BPP in sub-exponential time or better.

Pseudo-random Generator (PRG)



G must be efficiently computable

Stretches t bits into m bits.

"fools" small circuits. For all C of size $\leq s$

G is (s, ϵ) indistinguishable.

$$\left| \Pr_{y}[C(y)=1] - \Pr_{z}[C(G(z))=1] \right| \leq \epsilon$$

y: ↳ m bits; z: ↳ t bits.

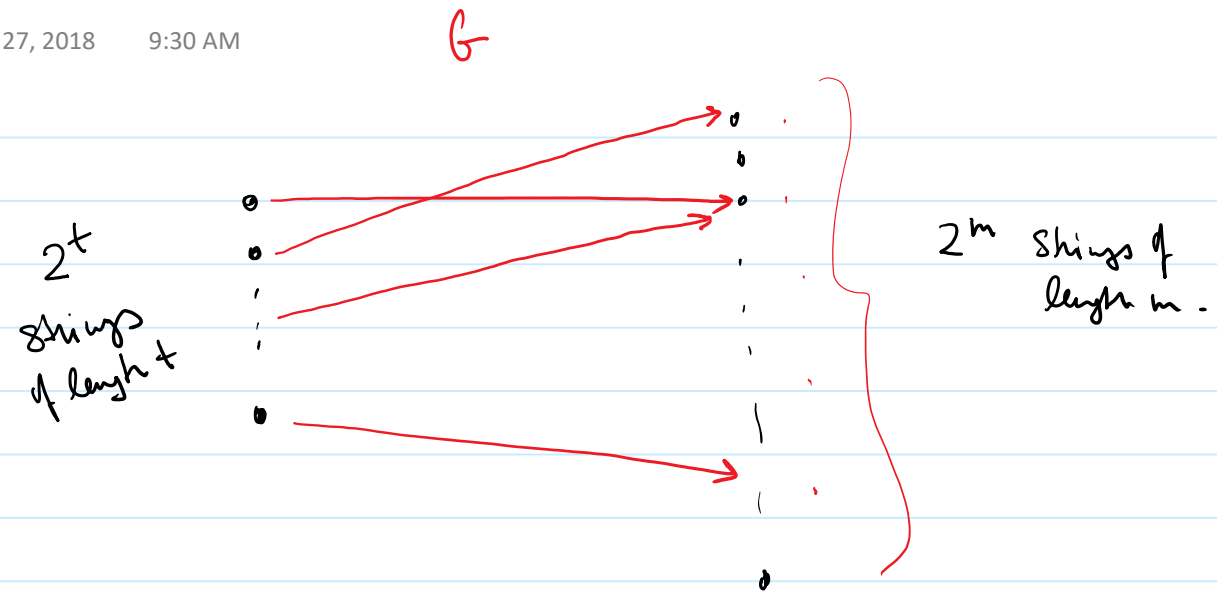
How to simulate BPP w/ a PRG.

Recall $L \in BPP \Rightarrow \exists$ p.p.t. TM M

$x \in L \Rightarrow \Pr_{y}[M(x,y) \text{ accepts}] \geq 2/3.$

$x \notin L \Rightarrow \Pr_{y}[M(x,y) \text{ ~~accepts~~ rejects}] \geq 2/3$

Friday, April 27, 2018 9:30 AM



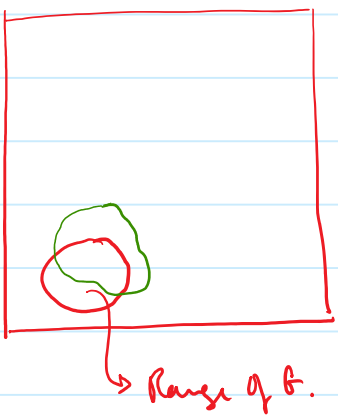
$|\text{Range of } G| \leq 2^t << 2^m.$

For circuit C (takes m-bit binary string as input).

$$\left| \Pr_{y \leftarrow U_m}[C(y)=1] - \Pr_{z \leftarrow U_t}[C(G(z))] \right| \leq \epsilon$$

Suppose G is 1-1

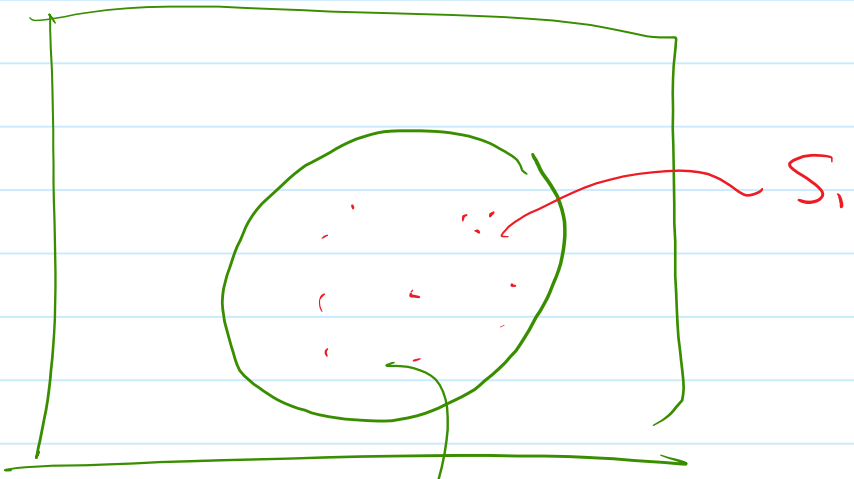
Let S be the set of strings which cause C to output 1.



$\frac{|S|}{2^m}$ $\frac{|S \cap \text{Range of } G|}{2^t}$

$\frac{2^t}{2^m} - 1$

2^m m-bit strings.



D D'

$\Pr_{y \leftarrow D}[y \in S]$

$\Pr_{y \leftarrow D'}[y \in S]$.

Thursday, April 26, 2018 8:26 PM

Convert M into a circuit
Hard-wire x into the circuit : $C_x(y)$. (y = random string.)

Both $|C_x|$ and $|y|$ are $\leq |x|^k$ for some k.

$$\Pr[C_x(y) = 1] \geq 2/3 \quad \text{for } x \in L$$
$$\Pr[C_x(y) = 1] \leq 1/3 \quad \text{for } x \notin L$$

Suppose we have a PRG with:
output length $= m$.
seed length $t << m$
error $\epsilon < 1/6$
fooling size s

(s, ϵ) - indistinguishable.

Want: $s(t)$ is superpolynomial.

Compute $\Pr_z[C_x(G(z)) = 1]$ exactly
evaluate $C_x(G(z))$ for every $z \in \{0,1\}^t$

running time: $(|C_x| + \text{time to compute } G(z))\, 2^t$

$|C_x|$ = time to simulate C_x on input $G(z)$

Can we distinguish between the two cases
$x \in L$ or $x \notin L$?

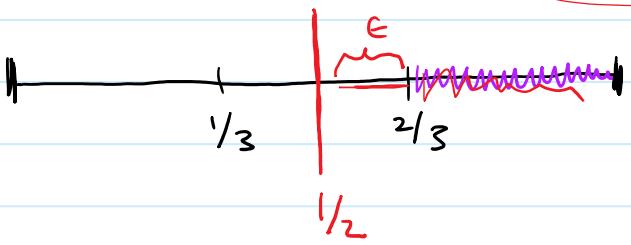
Thursday, April 26, 2018 8:26 PM

If C_x is poly-sized, smaller than fooling size of G.

$x \in L \quad \Pr_y[C_x(y)=1] \geq 2/3 \qquad \Pr_z[C_x(G(z))=1] \geq 2/3 - \epsilon$

since $\epsilon < 1/6$

Prob $> 1/2$



$x \notin L \quad \Pr[C_x(y)=1] \leq 1/3 \qquad \Pr_z[C_x(G(z))=1] \leq 1/3 + \epsilon$

since $\epsilon < 1/6$

Prob $< 1/2$.

If we can compute $\Pr_z[C_x(G(z))=1]$ exactly, we can determine if $<$ or $>$ $1/2$.

Goal: For all $0 < \delta < 1$ build a family of PRG's $\{G_m\}$ with

output length $= m$
seed length $= t = m^\delta$
error $\epsilon < 1/6$

fooling size: super-poly
running time: m^c

Length of random string n^k

Implies $BPP \subset \bigcap_\delta TIME(2^{n^\delta}) \not\subseteq EXP$.

Simulation runs in time

$O(\text{poly}(m) \cdot 2^{m^\delta}) = O(2^{m^{2\delta}}) = O(2^{n^{2k\delta}})$

For any γ, Pick δ so that: $2k\delta < \gamma$

Thursday, April 26, 2018 8:26 PM

In order to get BPP $\subseteq$ P, need $t = O(\log m)$

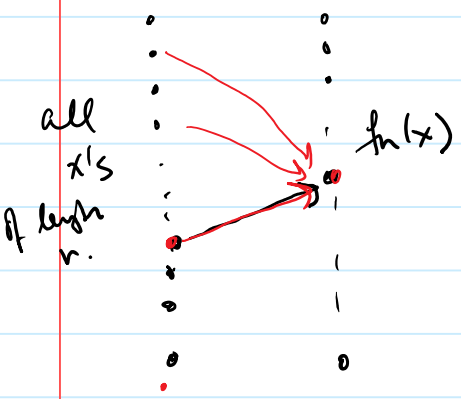
The existence of PRGs requires a complexity assumption.
Assumes that it's hard to distinguish between truly random & pseudo-random strings.

Definition: One Way Function (OWF)

function family $f = \{f_n\}$ $f_n : \{0,1\}^n \to \{0,1\}^n$

f is $(s(n), \epsilon(n))$ one-way. where $s(n)$ super-poly.

- f_n computable in polynomial time
- for every circuit family $\{C_n\}$ $|C_n| \le s(n)$.
- $\Pr_x [C_n(f_n(x)) \in f^{-1}(f(x))] \le \epsilon(n)$



$\epsilon(n) = o(n^{-c})$ for all c.

This requires hardness on average which is a stronger assumption than worst-case hardness.

It is generally believed that one-way functions exist (integer mult, discrete log, etc.)

Widely used in cryptography.

Thursday, April 26, 2018 8:26 PM

PRG $\Rightarrow$ OWF

PRG

seed length = t

output length = $2t$

Fooling size: $S(t)$

Error: $\epsilon(t)$

Computation time: $O(t^c)$

$(S(t), \epsilon(t))$ – indistinguishable.

$\Rightarrow$

OWF

$f: \{0,1\}^{2t} \to \{0,1\}^{2t}$

No circuit of size $S(t) - O(t^{2c})$ can invert w.p. $\epsilon(t) + 1/2^t$

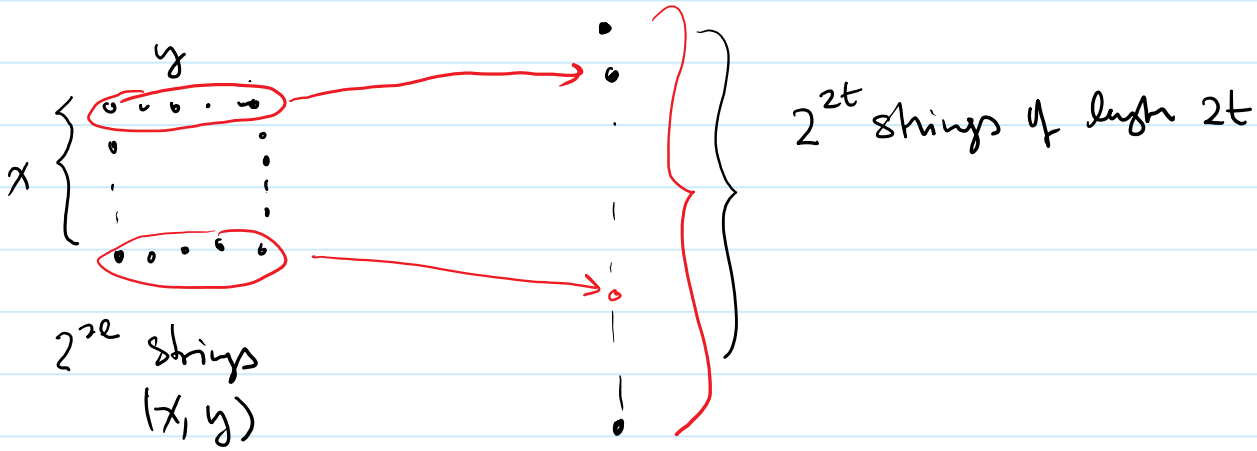
$(S(t) - O(t^{2c}), \epsilon(t) + 1/2^t)$ – one way

$G: \{0,1\}^t \to \{0,1\}^{2t}$

$f: \{0,1\}^{2t} \to \{0,1\}^{2t}$ $\quad f(x,y) = G(x)$

Show: If a circuit of size S can invert f w.p. ϵ then

There is a circuit of size $S + O(t^{2c})$ that can distinguish between random strings and outputs of G w.p. $\epsilon - 1/2^t$



Thursday, April 26, 2018 8:26 PM

Suppose C inverts f w.p. ϵ

Here is a distinguisher for G:

Given $z \in \{0,1\}^{2\ell}$
$(x,y) = C(z)$
if $G(x) = z$ accept.
O.W. reject.

If the input is a random $z \in \{0,1\}^{2\ell}$ Can only invert if z is in the range of G prob $\leq \frac{1}{2^\ell}$

→ Circuit to compute
$|C| + O(t^{2c}) = S + O(t^{2c})$
t^c is the computation time for G.

If given $G(x)$ for a random x

w.p. ϵ C finds (x', y') $f(x', y') = G(x)$
$G(x') = G(x)$

procedure will accept.

$$\left| \Pr_{z \in \{0,1\}^{2\ell}}[\text{Procedure Accepts}] - \Pr_{x \in \{0,1\}^{\ell}}[\text{Procedure accepts } G(x)] \right| \geq \epsilon - \frac{1}{2^\ell}$$

$\leq 1/2^\ell$ $\geq \epsilon$