Pseudorandomness: Lecture 15.

Instructor: Ramprasad.
Date:
Lecture #: 15.

Agenda:
- ▷ Randomised log space
- ▷ Nisan's PRG for RL.

We have some algo $A \in C$ for a decision problem.

$x \in L$: $\underset{\substack{\uparrow \\ m \text{ random bits.}}}{A}$: Yes or No

$$x \in L \qquad \Pr_r\left[A(x,r) = \text{Yes}\right] \geq 2/3$$

$$x \notin L \qquad \Pr_r\left[A(x,r) = \text{No}\right] \geq 2/3$$

If C is the class P, then this is BPP

Stupid derandomisation: Loop over all $\{0,1\}^m$ strings for r
Check if $\geq 2/3$ & then accept.

PRG for C:
$$G : \underset{\text{seed}}{\{0,1\}^d} \longrightarrow \{0,1\}^m \qquad d \to m \text{ stretch.}$$

is an ε-PRG for C if $\forall A \in C$ $\forall x \in \{0,1\}^n$.

$$\left| \mathop{E}_{r \sim \mathcal{U}_m}\left[A(x,r)\right] - \mathop{E}_{y \sim \mathcal{U}_d}\left[A(x, G(y))\right] \right| \leq \varepsilon.$$

Less-stupid derand: Loop over all seeds $y \in \{0,1\}^d$.
Run A on x & $G(y)$.
Acc if $\geq 2/3 - \varepsilon$ accept.

Time: $2^d \cdot$ "Computing $G(y)$" $\cdot$ "running time of A".

Recap:

- PRGs against a class $\mathcal{C} = \{ T : \{0,1\}^m \to [-1,1] \}$

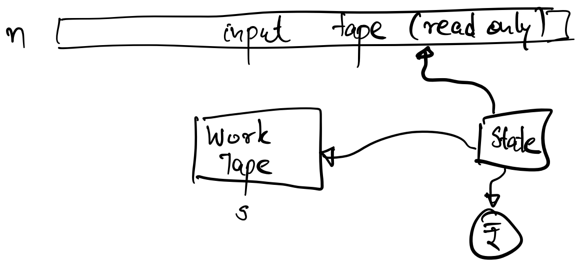
$$G : \{0,1\}^d \longrightarrow \{0,1\}^m \quad \text{s.t} \quad \forall\, T \in \mathcal{C}$$

$$\left| \mathbb{E}[T(G(\mathcal{U}_d))] - \mathbb{E}[T(G(\mathcal{U}_m))] \right| \leq \varepsilon.$$

- If $\mathcal{C} \subseteq$ circuits of size s, then the "right" $d = O(\log s + \log 1/\varepsilon)$.

- Goal: get as close to as possible, for some interesting class of algorithms.

Low space algorithms:



$|Q|$

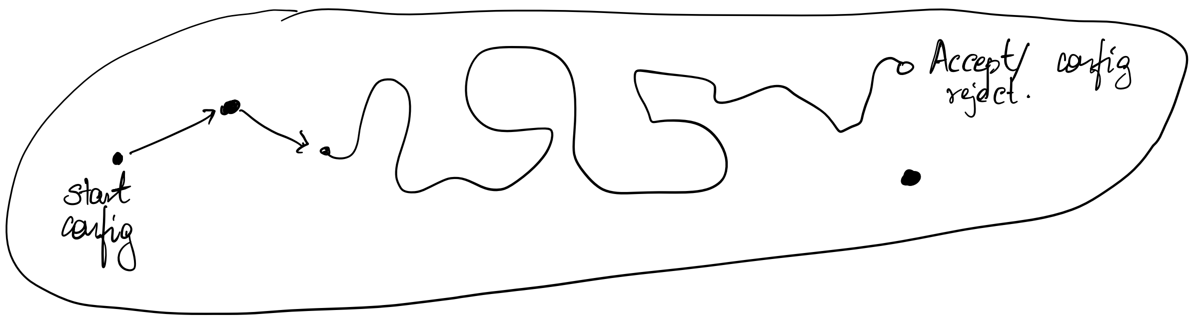
Think of input tape containing A_G, s, t.
Worktape has space $O(\log n)$ bits. Can only store const. many vertices at any point.

Fix an input.
A configuration:

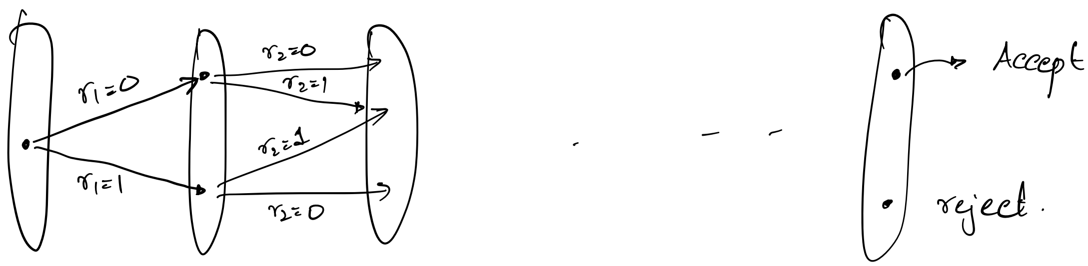
- What is the current state?
- What are the work tape contents?
- Where is the TM heads pointing?

$$\#\text{Configurations}: \; |Q| \cdot 2^s \cdot s \cdot n \;=\; \text{poly}(n)$$

if $s = O(\log s)$



What if we throw in randomness?



$x \in L$ $\quad\quad \Pr_r [A(x,r) = \text{Accept}] \geq 2/3$

$x \notin L$ $\quad\quad \Pr_r [A(x,r) = \text{Acc}] \leq 1/3$

Width = small (memory that the algo has)

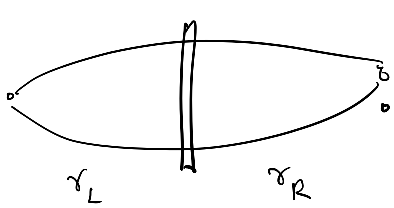
Length = T (time the algo runs for).

Theorem: [Nisan] There is an explicit PRG $G:\{0,1\}^d \to \{0,1\}^m$ for length T width W branching programs with

$$d = O\left(\log T \log\left(\frac{TW}{\varepsilon}\right)\right)$$

$$= O(\log^2 m) \quad \text{for } RL.$$

Idea:



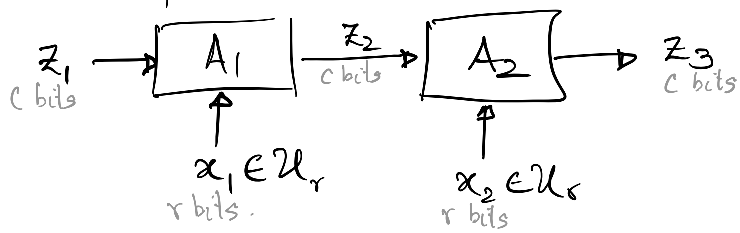
Does r_L & r_R need to be independent?

The algo can't remember a lot about r_L.

There is still $|r_L| - \log W$ "bits of entropy"

1'

A simpler abstraction.



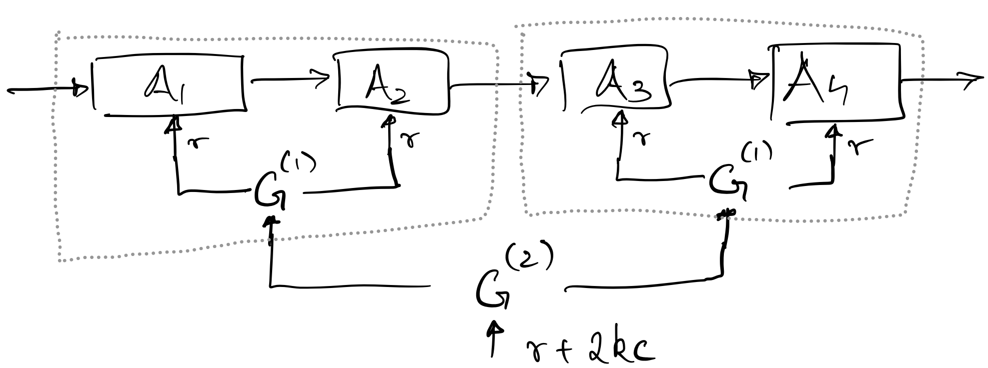
2-step comm-c Algo: $\{0,1\}^c \times \{0,1\}^r \times \{0,1\}^r \to \{0,1\}^c$.

Defn: A map $G: \{0,1\}^d \to \{0,1\}^r \times \{0,1\}^r$ is an ε-PRG for 2-step comm. c algos if for all $z_1 \in \{0,1\}^c$: the "output distribution" is "similar" to the "uniform behavior":

$$\sum_{z_3 \in \{0,1\}^c} \left| \Pr_{x_1,x_2 \sim \mathcal{U}} \left[A_2(A_1(z_1,x_1),x_2) = z_3 \right] - \Pr_{x_1,x_2 = G(y) \quad y \sim \mathcal{U}_d} \left[A_2(\qquad) \right] \right| \le \varepsilon$$

Lemma: There are explicit $G: \{0,1\}^d \to \{0,1\}^{2r}$ PRGs for 2-step c-comm. algos with $d = r + kc$ where k is a fn of just c & ε.

We will prove this later. Now what?



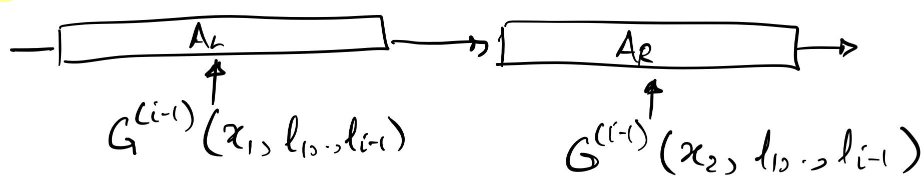
$$G^{(1)} = G : \{0,1\}^{r+kc} \longrightarrow \{0,1\}^r \times \{0,1\}^r$$
$$(x, l) \longmapsto (x_1, x_2)$$

$$G^{(i)} = \{0,1\}^{r+ikc} \longmapsto \{0,1\}^{2^i \cdot r}$$
$$(x, l_1, \dots, l_i) = G^{(i-1)}(x_1, l_1, \dots, l_{i-1}) \cdot G^{(i-1)}(x_2, l_1, \dots, l_{i-1})$$
$$\text{where } x_1 x_2 = G(x, l_i)$$

We need to analyse how well this does.

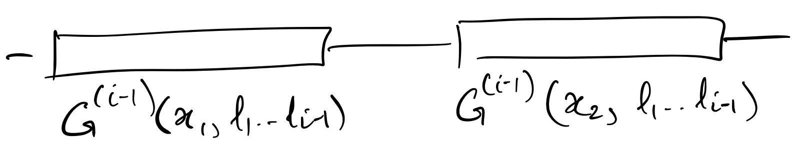
Lemma: Suppose base is an ε-PRG for 2-step comm. c algo. $G^{(i)}$ is an ε'-PRG for 2^i-step algo with $\varepsilon' \leq 3^i \cdot \varepsilon$

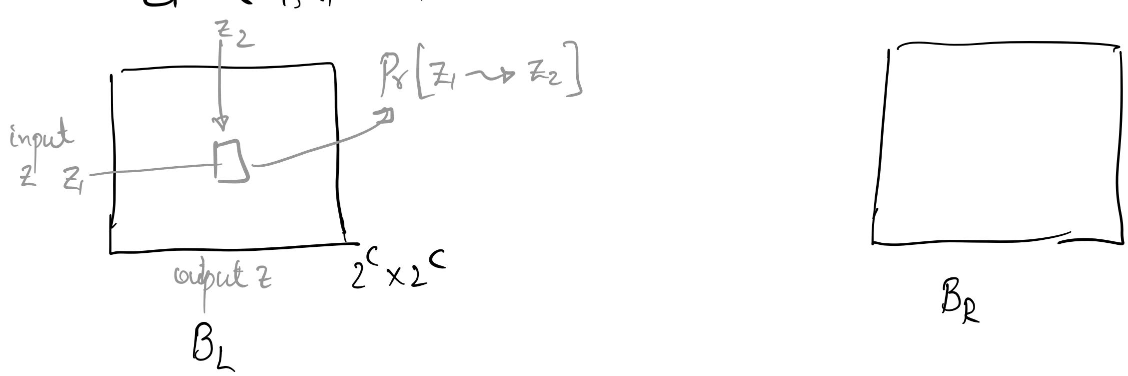
Pf:



where $x_1 x_2 = \underbrace{\cancel{G(x, l_i)}}_{\text{uniform}}$ At most ε.

$\therefore$ With just ε error, we may assume x_1 & x_2 are uniform random strings.





$\|A\|_\infty = \max_i \sum_j |A_{ij}|$

Facts: $\|A+B\|_\infty \leq \|A\|_\infty + \|B\|_\infty$

$\|AB\|_\infty \leq \|A\|_\infty \cdot \|B\|_\infty$

$$\begin{aligned}\|B_L B_R - \tilde{B}_L \tilde{B}_R\|_\infty &\leq \|B_L B_R - B_L \tilde{B}_R\| + \|B_L \tilde{B}_R - \tilde{B}_L \tilde{B}_R\|_\infty \\ &\leq \|B_L\|_\infty \cdot \|B_R - \tilde{B}_R\|_\infty + \|B_L - \tilde{B}_L\|_\infty \cdot \|\tilde{B}_R\|_\infty. \\ &\leq 2\,\varepsilon_{i-1}\end{aligned}$$

$\Rightarrow$ Total error $\leq 2\varepsilon_{i-1} + \varepsilon \leq 3^i \varepsilon$ $\square$

Building the base PRG.

$G(\text{seed}) = (x_1, x_2)$.

What do we want?

$$\begin{tikzcd} \arrow[r, "z_1"] & \boxed{A_1} \arrow[r, "z_2"] & \boxed{A_2} \arrow[r, "z_3"] & {} \\ & x_1 \arrow[u] & x_2 \arrow[u] & \end{tikzcd}$$

Fix z_1, z_3. For any $b \in \{0,1\}^c$

$S_b = \{x_1 : A_1(z_1, x_1) = b\}$.

$T_b = \{x_2 : A_2(b, x_2) = z_3\}$.

$$\Pr\left[\xrightarrow{z_1} \underset{\substack{\uparrow \\ x_1 \in \{0,1\}^r}}{\boxed{A}} \longrightarrow \underset{\substack{\uparrow \\ x_2}}{\boxed{A_2}} \longrightarrow z_3 \right] = \sum_b \Pr\left[x_1 \in S_b ,\ x_2 \in T_b \right] \qquad \underset{(z_1, z_3)}{(*)}$$

Candidate: $G : (x, \ell) = x, \Gamma_H(x, \ell)$

(Candidate: $(x, h) \mapsto (x, h(x))$
where $h \in \mathcal{H}$ - p.w.i.h.f)

EML: $\left| \Pr_{(x,y) \in G}\left[x_1 \in S, y \in T \right] - \alpha\beta \right| \leq \lambda \sqrt{\alpha\beta} \leq \lambda.$

$$\underset{z_1 z_3}{(*)} \leq \sum \left(\mu(S_b) . \mu(T_b) + \lambda \right) = \Pr\left[\begin{smallmatrix} \text{under} \\ \text{uniform} \end{smallmatrix} \right] + \lambda . 2^c.$$

$\Rightarrow$ PRG and uniform are within $\lambda . 2^c$ entry wise.

$\| \text{PRG} - \text{Uniform} \|_\infty \leq \lambda . 2^c . 2^c = \lambda . 4^c.$

$\Rightarrow$ If $\lambda \leq \varepsilon / 4^c$, we are done

If H is Ramanujan, then $\lambda \approx \frac{2}{\sqrt{D}} \Rightarrow D = 16^c / \varepsilon^2.$

$\Rightarrow kc = |\ell| = O(\log D) = O(c + \log 1/\varepsilon).$

$\square$

Instantiating for width W & length T BPs,

$c = \log W. \qquad i = \log T. \qquad 3^i \leq T^2.$

Need a base PRG with $\varepsilon_1 \leq \varepsilon / T^2$

Base needs $c + \log(T/\varepsilon) = O(\log TW/\varepsilon).$

Final seed length = $r + ikc$

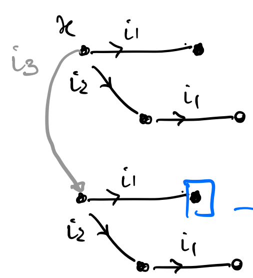
$$\Rightarrow O(\log T \cdot \log TW/\varepsilon).$$

$\square$

What does this generator look like?

$$G^{(1)}(x,i) = x, \Gamma(x,i)$$

$$G^{(2)}(x,i_1,i_2) = G^{(1)}(x,i_1) \quad G^{(1)}(\Gamma(x,i_2), i_1)$$
$$= x, \Gamma(x,i_1) \quad \Gamma(x,i_2) \quad \Gamma_{1,2}(x,(i_2 i_1))$$



A weird "parallelogram" in the graph H.
Remark: Any specific block can be computed efficiently.

$\therefore$ Suppose we have some language in BPL, then we can enumerate over all $O(\log^2 n)$-length seeds.
- enumerate in $2^{(\log^2 n)}$ time

$$\text{BPL} \subseteq \text{DSPACE}(\log^2 n).$$

Thm: [Nisan] $\text{BPL} \subseteq \text{TISP}(\text{poly}(n), O(\log^2 n))$

Next lecture:

[Saks-Zhou] $\text{BPL} \subseteq \text{SPACE}((\log n)^{3/2})$.

believed to be "yes".

Major open problem: Is $\text{BPL} = \text{L}$?