18.405J/6.841J Advanced Complexity Theory	April 1, 1999
Lecture 14	
Lecturer: Daniel A. Spielman	*Scribe: Prahladh Harsha*

1 Pseudo-Random Generators

In today's lecture we shall introduce the notion of a *Pseudo-Random Generator* (PRG). Roughly speaking, a PRG is a deterministic device that takes a fairly small (random) seed, blows it up in size and fools a randomised TM into believing that this blown-up string was a purely random string. In more formal terms, we have

Definition 1 *Let M be a randomised* TM *that on input w requires $f(|w|)$ random bits. The family $\{g_n\}_{n=1}^{\infty}$ of functions ($g_n : \{0,1\}^{s(n)} \rightarrow \{0,1\}^{f(n)}$) is an $\epsilon-$generator for M if*

$$\forall w \qquad \left| \underset{r \in \{0,1\}^{f(|w|)}}{\text{Prob}} [M(w,r)\text{accepts}] - \underset{z \in \{0,1\}^{s(|w|)}}{\text{Prob}} \left[M(w, g_{|w|}(z))\text{accepts}\right] \right| < \epsilon$$

Instead of a family of functions, it is more convenient to view g, the $\epsilon-$generator for M as a TM that on input a string of length $s(n)$ outputs a string of length $f(n)$. $s(n)$ is the length of the seed of the PRG.

2 Randomised Logspace TM

In this lecture, we shall present a PRG that fools all randomised logspace TMs. In the next lecture, we shall see a PRG for polynomial time TMs based on some unproven hardness assumptions. Before proceeding any further, we have to clarify a point regarding how the random bits are accessed in a randomised space bounded computation (logspace in our case). There are 2 varying definitions of randomised space bounded TMs based on the manner the random bits are accessed

- The TM obtains the random bits as and when required by it.
- The string of random bits is fed as an auxiliary off-line input (on a separate tape) in addition to the regular input to the TM. In this case, the head accessing the random bits on this tape can move back and forth on the tape.

It is to be noted that in the case of randomised time bounded computation it is immaterial which convention we observe. For randomised space bounded computation, we shall consider only TMs of the first kind or equivalently consider TMs of the second kind in which the head on the random tape is restricted in the sense that it can only move right along the tape (ie., the random tape is one-way read-only tape) We shall also assume that all randomised space S TMs halt in time less than 2^S.

3 Nissan's Generator

The PRG for randomised logspace TMs is as follows:

Theorem 2 *There exists a* SPACE(n) TM *G, $G : \{0,1\}^{O(\log^2 m)} \rightarrow \{0,1\}^m$ such that for all randomised logspace* TM *M, G is a $\frac{1}{n}-$generator for M.*[1]

[1]We shall infact prove a more general result that for all randomised space S TM M, G is a $\frac{1}{2^S}-$generator for M.

Before going into proving the existence of a PRG as mentioned in Theorem 2, we shall first study the structure of the computation tableau of a randomised space S TM and find how this structure can be exploited to reduce the randomness. The computation tableau for a randomised space S TM is as shown in Figure 1.

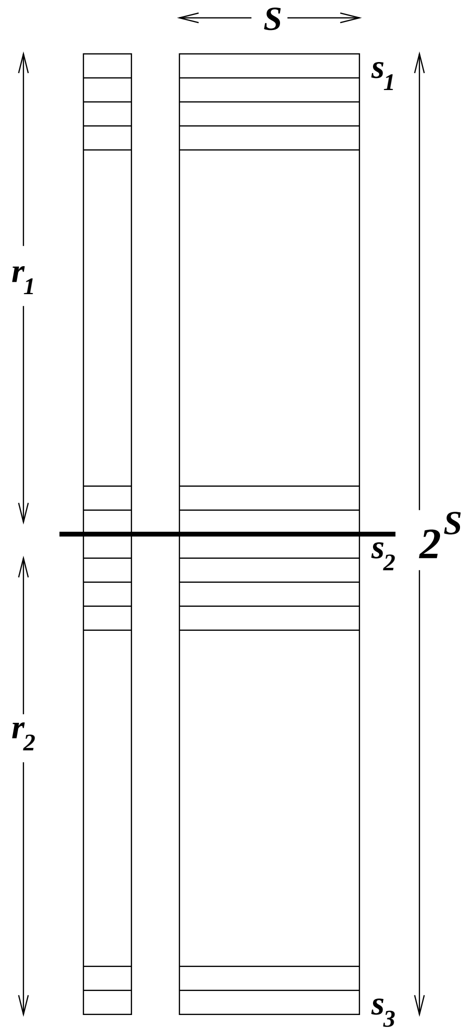

Figure 1: The Computation Tableau of a space S TM

In the original definition of the randomised TM, the computation requires atmost $R(= 2^S)$ random bits. Consider the tableau being divided into 2 halves with each half requiring random strings r_1 and r_2 respectively each of length $r(= R/2)$. If r_1 and r_2 are chosen independently, then by definition the randomised TM works to give the right results. We would like to choose r_1 and r_2 in such a manner that the probability of going from state s_1 to state s_2 (ie., Prob $[s_1 \to s_2]$) and Prob $[s_2 \to s_3]$ are not significantly different from the case when r_1 and r_2 are chosen independently. We would now use the fact that the computation tableau is very thin (width S compared to length 2^S) to let us choose r_1 and r_2 in a manner better than independently.

To put things more formally, we have 2 algorithms A_1 and A_2 such that

$$\begin{tikzcd} A_1 & x_1 \arrow[l] \\ A_2 \arrow[from=u, "b_1"] & x_2 \arrow[l] \\ b_2 \arrow[from=u] \end{tikzcd}$$

Figure 2: A_1, A_2 with random inputs

- A_1 takes an input x_1 of length r and outputs a string b_1 of length c.
- A_2 takes as input the output b_1 of A_1 and another string x_2 of length r and outputs a string b_2 of length c.

We can consider A_1 and A_2 to be the upper and lower halves respectively of the computation of the randomised space S TM and x_1, x_2 to be the random strings r_1, r_2. What we are in search of is a generator that supplies strings x_1 and x_2 in a fashion better than choosing them independently. For notational brevity, given a function g, define functions g^l and g^r such that $g^l(z)$ and $g^r(z)$ denote the left half and the right half of the string $g(z)$ (ie., $g(z) = g^l(z) \circ g^r(z)$ and $|g^l(z)| = |g^r(z)|$)[2]

Definition 3 *A function g ($g : \{0,1\}^t \to \{0,1\}^r \times \{0,1\}^r$) is defined to be a $\epsilon-$ generator for communication c if for all functions A_1 and A_2 such that $A_1 : \{0,1\}^r \to \{0,1\}^c$ and $A_2 : \{0,1\}^c \times \{0,1\}^r \to \{0,1\}^c$, we have that*

$$\forall b \qquad \left| \underset{x_1,x_2\in\{0,1\}^r}{\text{Prob}} [A_2(A_1(x_1), x_2) = b] - \underset{z\in\{0,1\}^t}{\text{Prob}} \left[A_2(A_1(g^l(z)), g^r(z)) = b\right] \right| < \epsilon$$

$$\begin{tikzcd} A_1 & g^l(z) \arrow[l] \\ A_2 \arrow[from=u, "b_1"] & g^r(z) \arrow[l] \\ b_2 \arrow[from=u] \end{tikzcd}$$

Figure 3: A_1, A_2 with inputs from generator

For notational convenience, we shall call a $2^{-c}-$generator for communication c a $c-$generator.

4 Proof of Theorem 2

For the present we shall assume the following lemma and give a sketch of its proof in the end of the lecture.

Lemma 4 *There exists a constant $k > 0$ such that for all r, c, there exists a polynomial time computable $c-$generator g where g is such that $g : \{0,1\}^{r+kc} \to \{0,1\}^r \times \{0,1\}^r$.*

[2] $\circ$ denotes the concatenation operator

Let $g_i : \{0,1\}^{(i+1)kS} \rightarrow \{0,1\}^{ikS} \times \{0,1\}^{ikS}$ be a $2S-$generator for $i = 1, 2, \ldots, S$ as guaranteed by lemma 4. Define functions $G_i : \{0,1\}^{(i+1)kS} \rightarrow \{0,1\}^{2^i kS}$ for $i = 0, 1, \ldots S$ inductively as follows

$$\begin{aligned} G_0(z) &= z \\ G_i(z) &= G_{i-1}(g_i^l(z)) \circ G_{i-1}(g_i^r(z)) \end{aligned}$$

We shall show that the existence of G_S implies Theorem 2. Clearly, by definition of G_S, G_S is a SPACE(n) TM (ie., it runs in space $O(kS^2)$). We only have to show that G_S fools all randomised space S TMs, which is implied by the following theorem.

Theorem 5 *For all* TMs *A that run in space S and in time $2^i kS$, G_i is an $(2^{i+1}-1)2^{-2S}-$generator for A*

Proof We shall prove by induction. Assume the statement of the theorem to be true for all $i < j$. Consider any TM A that runs in space S and time $2^j kS$ and divide its computation into 2 halves A_1 and A_2. By induction hypothesis we have that G_{j-1} is a $(2^j - 1)2^{-2S}-$generator for both A_1 and A_2 (since both A_1 and A_2 run in space S and time $2^{j-1}kS$). Hence we have that for all b_1, b_2:

$$\left| \operatorname*{Prob}_{x_1}[A_1(x_1) = b_1] - \operatorname*{Prob}_{y_1}[A_1(G_{j-1}(y_1)) = b_1] \right| < (2^j - 1)2^{-2S} \tag{1}$$

$$\left| \operatorname*{Prob}_{x_2}[A_2(b_1, x_2) = b_2] - \operatorname*{Prob}_{y_2}[A_2(b_1, G_{j-1}(y_2)) = b_2] \right| < (2^j - 1)2^{-2S} \tag{2}$$

where $x_1, x_2 \in \{0,1\}^{2^{j-1}kS}$ and $y_1, y_2 \in \{0,1\}^{jkS}$. Combining (1) and (2) we have,

$$\left| \operatorname*{Prob}_{x_1,x_2}[A_2(A_1(x_1), x_2) = b_2] - \operatorname*{Prob}_{y_1,y_2}[A_2(A_1(G_{j-1}(y_1)), G_{j-1}(y_2)) = b_2] \right| < (2^{j+1} - 2)2^{-2S} \tag{3}$$

As g_j is a $2S-$generator, we have

$$\left| \operatorname*{Prob}_{y_1,y_2}[A_2(A_1(G_{j-1}(y_1)), G_{j-1}(y_2)) = b_2] - \operatorname*{Prob}_{z}\left[A_2(A_1(G_{j-1}(g_j^l(z))), G_{j-1}(g_j^r(z))) = b_2\right] \right| < 2^{-2S} \tag{4}$$

Combining (3) and (4) we have,

$$\left| \operatorname*{Prob}_{x_1,x_2}[A_2(A_1(x_1), x_2) = b_2] - \operatorname*{Prob}_{z}\left[A_2(A_1(G_{j-1}(g_j^l(z))), G_{j-1}(g_j^r(z))) = b_2\right] \right| < (2^{j+1} - 1)2^{-2S}$$

which proves that G_j is a $(2^{j+1}-1)2^{-2S}-$generator for A. To complete the proof, we have to anchor the induction. As $G_0(z) = z$, the statement of the theorem is trivially true for $i = 0$. ■

5 Sketch of Proof of Lemma 4

Lemma 4 basically tells us that Figure 2 can be replaced by Figure 3. In other words, the $\epsilon-$generator for communication $r + kc$, g, should construct strings $g^l(z)$ and $g^r(z)$ such that functions A_1 and A_2 are fooled into believing that these strings were random ones. For a given b_1, b_2, the "good" x_1, x_2 are given by

$$R_{b_1,b_2} = \{(x_1, x_2) | A_1(x_1) = b_1, A_2(b_1, x_2) = b_2; x_1, x_2 \in \{0,1\}^r\}$$

The generator g should satisfy the following requirement in terms of R_{b_1,b_2}

$$\forall b_1, b_2 \qquad \left| \operatorname*{Prob}_{z \in \{0,1\}^{r+kc}}\left[(g^l(z), g^r(z)) \in R_{b_1,b_2}\right] - \operatorname*{Prob}_{x_1,x_2 \in \{0,1\}^r}\left[x_1, x_2 \in R_{b_1,b_2}\right] \right| < \epsilon$$

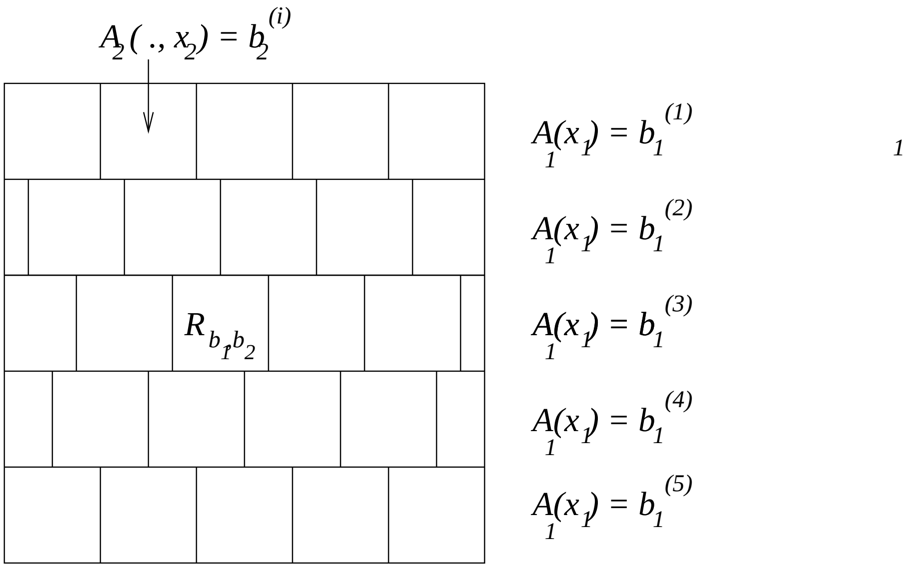

Figure 4: Structure of R_{b_1,b_2}

Let us look at the structure of R_{b_1,b_2}. let us consider a graph on vertices $V \times V$ where $V = \{0,1\}^r$ (ie., the vertices correspond to all possible pairs (x_1, x_2). Also for the present we shall assume there are no edges in the graph.) The graph on $V \times V$ can be divided into rows as shown in Figure 4 where each row corresponds to an element $b_1^{(i)}$ and contains all x such that $A_1(x) = b_1^{(i)}$. Now, each of these rows is further divided into blocks corresponding to $b_2^{(j)}$ with each block containing all elements x such that $A_2(b_1^{(i)}, x) = b_2^{(j)}$. Viewing $V \times V$ in this manner, R_{b_1,b_2} looks like a combinatorial rectangle of the form $S_1 \times S_2; S_1, S_2 \subseteq V$, where $S_1 = \{x_1 | A_1(x_1) = b_1\}, S_2 = \{x_2 | A_2(b_1, x_2) = b_2\}$ (refer figure). Thus the probability that a random (x_1, x_2) is in R_{b_1,b_2} is equal to the probability that a random pair of vertices in $V \times V$ is from $S_1 \times S_2$. (ie., $\text{Prob}\left[(x_1, x_2) \in R_{b_1,b_2}\right] = \frac{|S_1||S_2|}{|V|^2}$) Now if we imagine an expander G on the set of vertices $V \times V$, then probability that a random edge in $E(G)$ lies in $E(S_1, S_2)$[3] is very close to the probability that a pair of random vertices lies in $S_1 \times S_2$ as guaranteed by the following lemma. And that does the job for us as then the generator could choose a random edge in G and output the 2 endpoints of that edge.

Lemma 6 *If G be an expander graph of degree d on vertices V and λ is an upper bound on all eigen values but for the largest, then*

$$\forall S_1, S_2 \subseteq V \qquad \left| \frac{E(S_1, S_2)}{|E|} - \frac{|S_1||S_2|}{|V|^2} \right| < \frac{\lambda}{d}$$

[3] $E(V_1, V_2) = \{e | e$ has one endpoint in V_1 and another in V_2