PRINCETON UNIVERSITY COS 522: COMPUTATIONAL COMPLEXITY

Lecture 19: PRG for randomized log-space computation

Lecturer: *Sanjeev Arora* Scribe: *Elad Hazan*

1 Introduction

This part of the lecture concerns an application of Pseudo-Random Generators due to Nisan. Nisan shows PRG for space bounded computation classes. In particular, we'll see how every polynomial time algorithm in $BP - L$ can be simulated in $SPACE(\log^2 n)$ and polynomial time.
Notice that Savitch's theorem implies that $RL \subseteq SPACE(\log^2 n)$. However, Savitch's construction requires super-polynomial time for simulation.
In almost 15 years since Nisan's result, there has been no major improvement of the construction. This is despite the conjecture $BP - L = L$, which remains open.

2 Extractors

Good extractors play a central role in construction henceforth. We use the following lemma regarding existence of good extractors:

DEFINITION 1 *A function $Ext : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ is a (k,ε)-extractor if for any (n,k) source X, the distribution $Ext(X, U_t)$ is ε close to the uniform distribution on m bits.*

LEMMA 1
Fix constants $\varepsilon > 0, c > 0$. For every n and $k \leq n$ there exists an (k,ε)-extractor $Ext : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^n$ where $t = O(n - k + \log \frac{1}{\varepsilon})$.

PROOF: The extractor we shall build uses random walks on constant degree expander graphs. The proof henceforth assumes some background in random walks and expander graphs. For a more detailed discussion see [**?**].
We build upon known constructions of constant degree expander graph family, who's representation is polynomial (and even linear) in the representation size of a single vertex [**?**].
Let G be any degree d uniform expander, with adjacency matrix A. Let M be the Markov chain defined by G (that is, M is a matrix such that in entry (i,j) there is the transition probability from node i to node j), hence $M = \frac{1}{d}A$.
Clearly M is ergodic, and the stationary distribution is $\frac{1}{n}\vec{1}$.
Let X be some distribution on the vertices of G. Then x can be written as:

$$X = \frac{1}{n}\vec{1} + \sum_{i=2}^{n} \alpha_i e_i$$

Where e_i are the eigenvectors of M (that form an orthogonal basis since M is symmetric). The reason that the coefficient of the first eigenvector is $\frac{1}{n}$ is since:

$$< X, \frac{1}{n}\vec{1} >= \frac{1}{n}\sum X_i = \frac{1}{n}$$

Therefore, after a random walk of length t, the resulting distribution on the vertices is:

$$\begin{aligned} M^t X &= M^{t-1}(MX) = M^{t-1}\left(\frac{1}{n}\vec{1} + \sum_{i=2}^{n} \alpha_i \lambda_i e_i\right) \\ &= \ldots = \frac{1}{n}\vec{1} + \sum_{i=2}^{n} \alpha_i \lambda_i^t e_i \end{aligned}$$

And hence (Cauchy-Schwartz):

$$\left\| M^t X - \frac{1}{n}\vec{1} \right\|_1 \leq \lambda^t \left\| X - \frac{1}{n}\vec{1} \right\|_2 \sqrt{n}$$

Where λ denotes the second largest eigenvalue of the Markov chain matrix. Therefore, given n bits with min entropy k, we perform the following:

- Construct an expander graph on $N = 2^n$ vertices with constant degree d.
- Let the start configuration x be a single vertex defined by taking the vertex denoted by the input n bits.
- Perform a random walk of length $O(n - k + \log \frac{1}{\varepsilon})$

Since the n bits had k entropy, we deduce that:

$$\left\| X - \frac{1}{N}\vec{1} \right\|_2^2 \leq \sum_i (p_i - \frac{1}{N})^2 \leq p_{max} \sum_i |p_i - \frac{1}{N}| \leq 2 \cdot 2^{-k}$$

Therefore, after a random walk of length t the distance to the uniform is:

$$\left\| M^t X - \frac{1}{N}\vec{1} \right\|_1 \leq \lambda^t \left\| X - \frac{1}{N}\vec{1} \right\|_2 \sqrt{N} \leq \lambda^t 2^{(n-k)/2}$$

And hence $t = O(n - k + \log \frac{1}{\varepsilon})$ should suffice for this to be smaller then ε. □

3 PRG for space bounded computation

3.1 Intuition

We expect a space-bounded machine not to be able to distinguish a truly random string from a long pseudo-random string. Specifically, a log-space machine that uses a long random string, because of it's memory limitation, should not be able to distinguish the case that part of the string is reused after some time interval. This idea is used as follows:
Let machine Q use $\frac{1}{2} d \log n$ space. Obviously Q has $W \leq n^d$ configurations and runs in time $L \leq n^d$. Consider the graph of all configurations of Q, such that each configuration is labelled with a timestamp in which it can be reached (w.l.o.g each configuration is valid for a single timestamp, as we can duplicate it to several configurations otherwise).
We now explain why approximately $L/2$ random bits should suffice for the execution of Q (instead of L). Suppose that for a certain run of machine Q, $\frac{L}{2}$ random bits were used to reach configuration $u_{L/2}$, as illustrated below. Also, suppose that the probability, over all random strings of length $\frac{L}{2}$, to reach $u_{L/2}$ is non-negligible, say at least $\frac{1}{W^c}$ for some constant c.

Then we claim that the random string which was used to reach $u_{L/2}$ contains almost enough "randomness" that suffices to continue with the execution of Q. The reason is that the min-entropy of the distribution on all the strings that will cause Q to reach $u_{L/2}$ is at least:

$$\log \frac{1}{W^c} 2^{L/2} = L/2 - c \log W$$

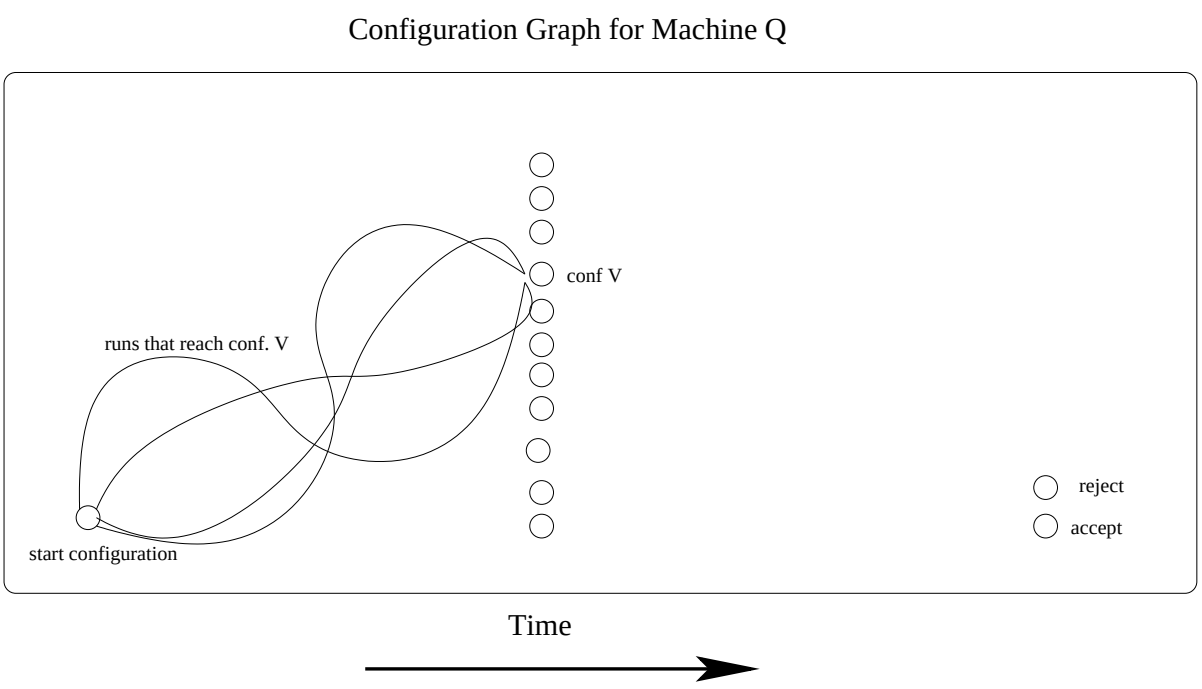

Figure 1: Configuration graph for machine Q

And thus using an extractor, we can extract almost this many truly random bits, add about $c \log W$ more new truly random bits, and get a random string with min-entropy approximately $L/2$. This "almost" random string can be used to continue with the execution of Q with small error compared to a standard execution. This idea can be carried further, such that randomness is recycled every smaller interval recursively. We proceed to describe the implementation of these ideas formally:

3.2 Formalization

The Nisan extractor is defined as:

DEFINITION 2 *For every integer $k \geq 0$ let the Nisan generator:*

$$G_k : \{0,1\}^{k \times r} \to \{0,1\}^{2^k \times r}$$

be defined recursively as (where $|S| = (k-1)r, |z| = r$ and the parameter r we fix later):

$$G_k(S \circ z) = \begin{cases} S & k = 1 \\ \\ G_{k-1}(S) \circ G_{k-1}(Ext(S,z)) & k > 1 \end{cases}$$

Denote by $S_{u,w,(k-1)\times r}$ the set of all strings x of length $(k-1) \times r$, such that the machine Q gets from configuration u to configuration w using $G_{k-1}(x)$ as the random string.
Now fix two possible configurations: v, u, and some time t. Denote by $C_{v,t}$ all the configurations such that the probability to reach them from configuration v in time t is at least $\frac{2}{W^c}$ (for some parameter c which we fix later on). Hence, Starting from v, the total probability to reach one of the configurations in $C_{v,t}$ in time t is at least $1 - \frac{2}{W^{c-1}}$. We make the following observation:

PROPOSITION 2
If $u \in C_{v,t}$ then the min entropy of the distribution $S_{v,u,t}$ (where each string $x \in S_{v,u,t}$ is chosen with equal probability) is at least $t - c \log W$.

PROOF: Since $u \in C_{v,t}$, the number of random strings to bring us to this configuration is at least $\frac{2}{W^c}(1 - \frac{2}{W^{c-1}})$ fraction of all strings.
Therefore, the min entropy on the distribution of these strings is at least:

$$\log \frac{1}{W^c} 2^t = t - c \log W$$

□

Define:

- $p_{u,v,i}$ - the probability to get from configuration u to configuration v using a truly random string of length 2^i.
- $q_{u,v,i}(S)$ - the probability to get from configuration u to configuration v using the string $G_i(S)$, where S is of length $i \cdot r$. If $S = U_{k \times r}$ then we denote this just as $q_{u,v,i}$.

Our goal is to prove the following main lemma regarding the extractor:

LEMMA 3
For every two configurations u, v and parameters $k \leq \log L$, $\varepsilon > \frac{2}{W^{c-1}}$:

$$|p_{u,v,k} - q_{u,v,k}| \leq 4^k \varepsilon$$

PROOF: The proof is by induction. For $k = 0$ this is obviously true. Now assume correctness for $k - 1$ and we prove for k. Note that:

$$p_{u,v,k} = \sum_w p_{u,w,k-1} \cdot p_{w,v,k-1}$$

$$q_{u,v,k} = \sum_w q_{u,w,k-1} \cdot q_{w,v,k-1}(Ext(S_{u,w,(k-1)\times r}, U_r))$$

According to these observations, we attempt to obtain a bound on the quantity in the claim. First, according to the triangle inequality:

$$\begin{aligned} |p_{u,v,k} - q_{u,v,k}| &= \left| \sum_w p_{u,w,k-1} \cdot p_{w,v,k-1} - \sum_w q_{u,w,k-1} \cdot q_{w,v,k-1}(Ext(S_{u,w,(k-1)\times r}, U_r)) \right| \\ &\leq \left| \sum_w p_{u,w,k-1} \cdot p_{w,v,k-1} - \sum_w q_{u,w,k-1} \cdot q_{w,v,k-1} \right| \\ &+ \left| \sum_w q_{u,w,k-1} \cdot q_{w,v,k-1} - \sum_w q_{u,w,k-1} \cdot q_{w,v,k-1}(Ext(S_{u,w,(k-1)\times r}, U_r)) \right| \end{aligned}$$

For the first summand, we can use the induction hypothesis, some algebra, and the fact that $\sum_w p_{u,w,t} = 1 = \sum_w q_{u,w,t}$ (since these are distributions) and derive a bound:

$$\begin{aligned} S_1 &:= \left| \sum_w p_{u,w,k-1} \cdot p_{w,v,k-1} - \sum_w q_{u,w,k-1} \cdot q_{w,v,k-1} \right| \\ &\leq \sum_w p_{u,w,k-1} \left| p_{w,v,k-1} - q_{u,w,k-1} \right| \\ &+ \sum_w q_{u,w,k-1} \left| p_{u,w,k-1} - q_{w,v,k-1} \right| \\ &\leq 4^{k-1}\varepsilon(\sum_w p_{u,w,k-1} + \sum_w q_{u,w,k-1}) \\ &\leq 2 \cdot 4^{k-1}\varepsilon \end{aligned}$$

We now try to bound the second summand. According to proposition 2, for all but a negligible part of the configurations, the min entropy of $S_{u,w,(k-1)\times r}$ is at least $(k-1)\times r - c\log W$. Hence, intuitively, if we add $c\log W$ more randomness then we can extract t more random bits and use them as a "random" string to continue with the machine execution.
Therefore, taking the parameter t from the extractor of lemma 1 to be $r = O((k-1)\times r - ((k-1)\times r - c\log W) + \log\frac{1}{\varepsilon}) = O(\log W + \log\frac{1}{\varepsilon})$, will result in the string $Ext(S_{u,w,(k-1)\times r}, U_r)$ to be of length $(k-1)\times r$ and ε-close to uniform. Hence:

$$\forall w \in C_{u,t} \ . \ |Ext(S_{u,w,(k-1)\times r}, U_r) - U_{(k-1)\times r}| \leq \varepsilon$$

And therefore:

$$\forall w \in C_{u,t} \ . \ |q_{w,v,k-1} - q_{w,v,k-1}(Ext(S_{u,w,(k-1)\times r}, U_r))| \leq \varepsilon \tag{1}$$

Returning to the second summand, we use equation 1, and derive the following bound:

$$\begin{aligned} S_2 &:= \left| \sum_w q_{u,w,k-1} \cdot q_{w,v,k-1} - \sum_w q_{u,w,k-1} \cdot q_{w,v,k-1}(Ext(S_{u,w,(k-1)\times r}, U_r)) \right| \\ &\leq \sum_w q_{w,v,k-1} \cdot |q_{w,v,k-1} - q_{w,v,k-1}(Ext(S_{u,w,(k-1)\times r}, U_r))| \\ &\leq \sum_{v \in C_{w,t}} q_{w,v,k-1} \cdot |q_{w,v,k-1} - q_{w,v,k-1}(Ext(S_{u,w,(k-1)\times r}, U_r))| + \frac{2}{W^{c-1}} \\ &\leq \varepsilon \cdot \sum_{v \in C_{w,t}} q_{w,v,k-1} + \frac{2}{W^{c-1}} \\ &\leq \varepsilon + \frac{2}{W^{c-1}} \leq 2\varepsilon \end{aligned}$$

Where the derivation above uses the fact that $\varepsilon > \frac{2}{W^{c-1}}$. Using the bounds on both summands we have proved the lemma. □

Using this lemma, we can create a PRG for space bounded computation classes. For completeness, we give a definition of PRG:

DEFINITION 3 *A function $G : \{0,1\}^m \to \{0,1\}^n$ is an ε-PRG for $SPACE(w)$ if for every TM in $SPACE(w)$, we have:*

$$|\Pr_y[Q(y) = 1] - \Pr_x[Q(G(x)) = 1]| \leq \varepsilon$$

Consider for example the class $BP - SPACE(\log n)$. Denote the start configuration by s, and the (single w.l.o.g) accepting configuration by a. Then taking $\varepsilon = \frac{1}{W^{c-2}}$:

$$|p_{s,a,\log n} - q_{s,a,\log n}| \leq 4^{\log n}\varepsilon \leq n^2 \frac{1}{n^{c-2}} \leq n^{c-4}$$

And taking $c > 5$ we get that the difference in probability is less then the total number of configurations of the machine. The total number of random bits will be $k \times r$, where $2^k = L = n^d$, and $r = O(\log W + \log \frac{1}{\varepsilon}) = O(\log n)$, for a total of $r \times k = O(\log^2 n)$ random bits.