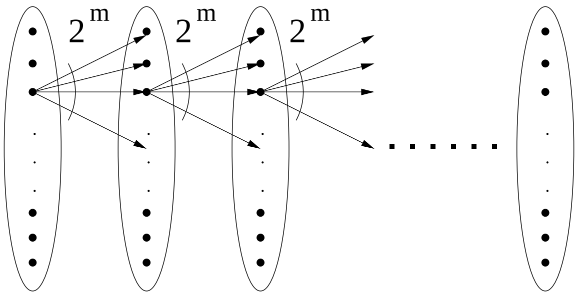
CS395T Psudorandomness and Combinatorial Constructions
Lecture 17, March 20, 2001
Lecturer: Prof. David Zuckerman, Scribe: Li-San Wang[1]

1 Pseudorandom Generators for Space-Bounded Computation

Let n be the length of the input. We define the *space* of a Turing machine to be $S(n) = \log_2(\text{number of configurations of the Turing machine})$. This definition of space is valid up to a constant multiple, when $S(n) \geq \log n$.

We can model a random computation as a layered graph. In the following figure, each layer correspond to the 2^S configurations of the machine; each dot represents a configuration. A transition from one layer to the next, as shown by an arrow, corresponds to a sequence of moves by the Turing machine. Assume the TM uses m random bits at each transition between two adjacent layers, so there are 2^m possible moves.



We want to construct a PRG such that at the end of the computation the distribution on the 2^S configurations using the PRG is close to that by using the true random bits. To say this more precisely, let $D(y)$ denote the distribution on configurations when using a string y of random bits. The goal is to construct $G : \{0,1\}^r \to \{0,1\}^R$ such that $|D(G(U_r)) - D(U_R)| \leq \epsilon$ at any time t of computation (here $R \geq r$).

We first show how well we can achieve by using the Leftover Hash Lemma. Pick $x \in_u \{0,1\}^m$, and pick h from a universal family of hash functions: $h \in_u \mathcal{H} = \{h : \{0,1\}^m \to \{0,1\}^l\}$. Recall the Leftover Hash Lemma: let $U = \{0,1\}^m$, $A \subset U$, $|A| \geq 2^k$. Given a universal family of hash functions $\mathcal{H}$, where each function maps $\{0,1\}^k$ to $\{0,1\}^{k-2l}$. Pick $h \in_u \mathcal{H}$ and $x \in_u A$. Then $h \circ h(x)$ is 2^{-l} uniform.

We have the following naive bound on the probability of the TM reaching the configurations at time t with very small probability:

$$\Pr[\text{Reaching a configuration at time } t \text{ having probability } < 2^{-2S}] \leq 2^S \times 2^{-2S} = 2^{-S}$$

Conditional on the m-bit string x leading to a particular configuration with probability $\geq 2^{-2S}$, x is uniform over a set of size $\geq \frac{2^m}{2^{2S}} = 2^{m-2S}$. Now hash x to $m - 2S - 2l$ bits.

[1] Thanks to Vladimir Trifonov for helping me prepare this note.

We then add fresh random bits and repeat. In symbols, recalling that the circle means concatenation:

$$\begin{aligned} x_1 &\leftarrow h(x) \circ s_1 \\ x_2 &\leftarrow h(x_1) \circ s_2 \\ &\vdots \end{aligned}$$

where s_i has length $2S + 2l$, $i = 1, 2, \ldots$. We use $\{x_i\}$ as the transition bits for the Turing Machine.

If $m = S^2$, then $|h| = S^2$, $|s_i| = O(S)$. So we add $O(S)$ random bits to obtain S^2 pseudorandom bits at each step. This gives a generator mapping $O(S^2)$ bits to S^3 bits.

2 Nisan's Generator

Nisan's PRG maps $O(S \log R)$ bits to R bits and fools any TM with space S, under the assumption $R \leq 2^S$. In the previous example $R = 2^S$, Nisan's PRG improves the result to $O(S^2) \to 2^S$.

The following are some example applications of Nisan's PRG:

- Random walks: maps $O(\log^2 n)$ bits to $n^{O(1)}$ bits, which is good enough for a random walk.
- Choosing random primes: $O(n)$ space to choose a random n bit prime; even with randomized primality test, we only need $O(n \log n)$ bits when using Nisan's PRG.
- Amplification of BPP: assume r random bits are used in the BPP algorithm, we need $O(r \log t)$ bits to get error 2^{-t}.

A simple example to illustrate this PRG is as follows. We consider the following simple layered graph where the space required is 1: we compute the parity of any fixed subset of the bits in the input string, so there are 2 configurations. This is an ϵ-biased test. Since Nisan's PRG fools space 1, the output of the PRG is ϵ-biased.

The idea of Nisan's generator is the recursive use of hash functions:

$$\begin{aligned} x &\in_u \{0,1\}^n \\ h_i &\in_u \mathcal{H} = \{h : \{0,1\}^n \to \{0,1\}^n\} \\ &\quad\text{(assuming } \mathcal{H} \text{ is a pairwise independent family)} \\ G_0(x) &= x \\ G_k(x, h_1, \ldots, h_k) &= G_{k-1}(x, h_1, \ldots, h_{k-1}) \circ G_{k-1}(h_k(x), h_1, \ldots, h_{k-1}) \end{aligned}$$

(Although the Leftover Hash Lemma only requires the hash family to have small collision probability, Nisan's generator requires pairwise independence.)

To motivate the lemma used by Nisan, consider the $m \times m$ symmetric Hadamard matrix H where each entry is 1 or -1. Therefore $HH^T = H^2 = mI$. Pick any minor K from H, and sum up all entries in K. One can show the sum is much smaller than the area of K; this implies the number of 1's and -1's are very close. Let λ be any eigenvalue of H, and v be the corresponding eigenvector. So we have $Hv = \lambda v \Rightarrow mv = H^2 v = \lambda^2 v \Rightarrow |\lambda| = \sqrt{m}$. Let A and B be the set of row and column indices of minor K, and let $a = |A|$, $b = |B|$. The sum of all elements in K is thus $1_A H 1_B$ ($(1_A)_j = 1$ if $j \in A$, 0 otherwise.) Therefore (using the L_2 norm)

$$|1_A^T H 1_B| \leq \ ||1_A||\ ||H||\ ||1_B||\ = \sqrt{mab}$$

When $ab > m$, $\sqrt{mab} < ab$, this bound is better than ab.

Now fix a hash function h. Treat one dimension of H as indices to the input x to the hash function, the other as the value $h(x)$. Then we can write $A, B \subseteq \{0,1\}^n$. Let $\rho(A) = \frac{|A|}{2^n}, \rho(B) = \frac{|B|}{2^n}$.

Definition 1 *A hash function h is $(\epsilon, A, B)-$*good *if*

$$\left| \Pr_x[x \in A \wedge h(x) \in B] - \rho(A)\rho(B) \right| \leq \epsilon$$

We first prove the following lemma:

Lemma 1

$$\Pr_{h \in \mathcal{H}}[h \textit{ is not } (\epsilon, A, B) - \textit{good}] \leq \frac{\rho(A)\rho(B)(1-\rho(B))}{2^n \epsilon^2}$$

Proof:

Let $M(x,h) = \begin{cases} 1 & \text{if } h(x) \in B \\ 0 & \text{otherwise.} \end{cases}$

For all x, $E_{h\in\mathcal{H}} M(x,h) = \Pr[h(x) \in B] = \rho(B)$. By the pairwise independence, for all x_1, x_2, $x_1 \neq x_2$, $E_{h\in\mathcal{H}} M(x_1,h) M(x_2,h) = \rho(B)^2$. We have

$$\begin{aligned}
& E_{h\in\mathcal{H}}[(\Pr{}_{x\in A}[h(x) \in B] - \rho(B))^2] \\
= \;& E_{h\in\mathcal{H}}[E_{x_1\in A, x_2 \in A}[M(x_1,h)M(x_2,h)] - 2\rho(B) E_{x\in A} M(x,h) + \rho(B)^2] \\
= \;& E_{h\in\mathcal{H}} E_{x_1\in A, x_2 \in A}[M(x_1,h)M(x_2,h)] - 2\rho(B) E_{x\in A} E_{h\in\mathcal{H}} M(x,h) + E_{h\in\mathcal{H}} \rho(B)^2 \\
& \text{(The second term follows because } h \text{ and } x \text{ are independent)} \\
= \;& E_{h\in\mathcal{H}} E_{x_1\in A, x_2 \in A}[M(x_1,h)M(x_2,h)] - \rho(B)^2 \\
= \;& \Pr[x_1 = x_2]\rho(B) + \Pr[x_1 \neq x_2]\rho(B)^2 - \rho(B)^2 \\
= \;& \frac{\rho(B)}{|A|} + (1 - \frac{1}{|A|})\rho(B)^2 - \rho(B)^2 \\
= \;& \frac{\rho(B)(1-\rho(B))}{|A|}
\end{aligned}$$

Now use Chebychev's inequality:

$$\Pr_{h \in \mathcal{H}}(|E_{x \in A} M(x,h) - \rho(B)| \geq \delta) \leq \frac{\rho(B)(1-\rho(B))}{\delta^2 |A|}$$

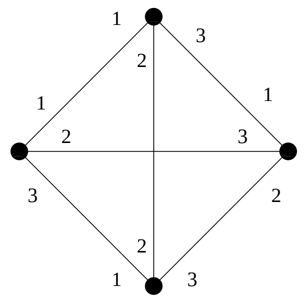
Take $\delta = \frac{\epsilon}{\rho(A)}$, we have the lemma.

$\square$

The proof of Nisan's PRG will continue in the next lecture.

3 Universal Traversal Sequences

A sequence $w \in [d]^r$ is a *universal traversal sequence* for d-regular n node connected undirected graphs, if for any d-regular graph G, any labeling of edges, any start vertex, following w will visit all vertices. In the following example, a sequence 12332, when one starts from any vertex, will visit all vertices in the end.



Using the probabilistic method one can show the existence of such sequences of polynomial length. Nisan's generator allows one to construct a universal traversal sequence of length $n^{O(\log n)}$. This is the best known result.