Lecture 28

We were in the middle of proof for the Nisan-Wigderson construction. We will complete the proof and then begin discussion of Extractors.

1 Nisan-Wigderson Construction

Let f be a boolean function $f : \{0,1\}^l \to \{0,1\}$. Let the hardness of f be $H(f) = O(m^2)$, $m = 2^{rl}$ $r > 0$. We define a family of subsets:

$$\begin{array}{l} S_1, \ldots, S_m \ \subseteq \ \{1, \ldots, t\}, \qquad t = O(l/r) \\ |S_i| = l \\ |S_i \cap S_j| \ \leq \ rl = \log(m) \end{array}$$

Then, the pseudorandom generator $NW_{f(S_1,\ldots,S_m)} : \{0,1\}^t \to \{0,1\}^m$ is given by:

$$NW_{f(S_1,\ldots,S_m)}(z) \ = \ f(z|S_1), \ldots, f(z|S_m)$$

Where $z|S = z_{s_1}, z_{s_2}, \ldots, z_{s_l}$, if $S = \{s_1, \ldots, s_l\} \subseteq \{1, \ldots, t\}$. $z|k$ denotes the kth bit in the sequence z.

Intuitively, since f is a hard function, it is easy to compute one pseudorandom bit from the input seed, since f looks random to any small circuit. The problem is to generate more than one random bit. To do this, we try to compute f on m nearly disjoint subsets of bits.

Lemma 1

If $S_1, \ldots, S_m \subseteq \{1, \ldots, t\}, t = O(l/r)$ are such that $|S_i| = l$ and $|S_i \cap S_j| \leq \log(m)$, for every f with $H(f) = O(m^2)$ and every C such that $|\mathbf{Pr}_z[C(NW_{f(S_1,\ldots,S_m)}(z) = 1] - \mathbf{Pr}_r[C(r) = 1]| > \delta$, there is a circuit A of size $\leq O(m^2) + size(C)$ such that $\mathbf{Pr}_x[A(x) = f(x)] \geq 1/2 + \delta/m$

Proof: We present the algorithm A.

Input: $x \in \{0,1\}^l$
Output: $\in \{0,1\}$

Pick at random $y \in \{0,1\}^{t-l}$; $r_i, r_{i+1}, \ldots, r_m \in \{0,1\}$
Compute $C(f_1(x,y), f_2(x,y), \ldots, f_{i-1}(x,y), r_i, \ldots, r_m)$
//Notation: $f_j(x,y) = f(z|S_j)$ where z equals x on bits in S_j and
// z equals y on bits in $\{1, \ldots, t\} - S_j$.
// Essentially we are seperating z into relevant bits (x) and irrelevant bits (y)
// $f_j(x,y) = f(z|S_j) = f(x)$
if $C(\ldots) = 1$ output r_i
else output $1 - r_i$

Lets compute the probability bounds of A. The proof is using the hybrid argument.

$$\begin{aligned}
H_0 &= r_1, r_2, \ldots, r_m \quad \text{completely random} \\
H_1 &= f(z|S_1), r_2, \ldots, r_m \\
H_{i-1} &= f(z|S_1), \ldots, f(z|S_{i-1}), r_i, \ldots, r_m \\
H_i &= f(z|S_1), \ldots, f(z|S_i), r_{i+1}, \ldots, r_m \\
H_m &= NW(z)
\end{aligned}$$

H_{i-1} and H_i are very close. If $|\mathbf{Pr}[C(H_m) = 1] - \mathbf{Pr}[C(H_0) = 1]| \ > \ \delta$, then we must have that $|\mathbf{Pr}[C(H_i) = 1] - \mathbf{Pr}[C(H_{i-1}) = 1]| \ > \ \delta/m$

So $\mathbf{Pr}_{x,y,r_i,\ldots,m}[A(x) = f(x)] \ > \ 1/2 + \delta/m$

We are not done yet. The probability here is over $x, y, r_i, \ldots, r_m$.

$$\begin{aligned}
\mathbf{Pr}_{x,y,r_i,\ldots,m}[A(x) = f(x)] &= E_{y,r_i,\ldots,r_m}\left(\mathbf{Pr}_x\left[A(x) = f(x)\right]\right) & \geq 1/2 + \delta \\
\Rightarrow \ \exists \ y, r_i, \ldots, r_m : \ &\mathbf{Pr}_x[A_{y,r_i,\ldots,r_m}(x) = f(x)] & \geq 1/2 + \delta \qquad (1)
\end{aligned}$$

So our final algorithm would be a modified version of A. $A_{y,r_i,\ldots,r_m}$ would have the parameters in Eqn. 1 built in. Otherwise it would be the same as A.

We now bound the space required by a circuit for $A_{y,r_i,\ldots,r_m}$. If we fix y only $|x| = \log(m)$ bits are needed in the computation of $f_i(x, y)$. So $f_i(x, y)$ can be computed by a circuit of size $O(2^{\log(m)}) = O(m)$. Since we may at most have to compute m functions, $A_{y,r_i,\ldots,r_m}$ has a circuit of size $O(m^2) + \text{size}(C)$.

□

Using this lemma we can proove the following theorem

Theorem 2 (Nisan-Wigderson)
If there is $L \in \mathbf{DTIME}(2^{O(n)})$ such that for all sufficiently large n if $H(L_n) \geq 2^{\epsilon n}$, $\epsilon > 0$, then for every m, there is a $G_m : \ \{0,1\}^{O(\log(m))} \to \{0,1\}^m$ that is a $(m^2, 1/m)$ pseudorandom generator computable in poly(m) time.

Proof: Given m fix l such that $2^{\epsilon l} \geq O(m^2)$.
$m = 2^{\epsilon l/2 - O(1)}; \ l = \frac{2}{\epsilon}\log(m) + O(1)$
Define $f : \ \{0,1\}^l \to \{0,1\}$ to be L restricted on inputs of length l.
f is computable in time $2^{O(l)} = m^{O(1/\epsilon)}$
Do NW construction with $r = 2/\epsilon$.
NW: $\{0,1\}^t \to \{0,1\}^m, \ t = O(l/r) = O(\frac{1}{\epsilon^2}\log(m))$
The security bounds follow from the previous lemma.

□

Corollary 3
Under the same assumptions as in the Nisan-Wigderson theorem, if $L \in \mathbf{BPP}(t(n))$ then $L \in \mathbf{DTIME}(t(n)^{O(1/\epsilon^2)})$ and so $\mathbf{P} = \mathbf{BPP}$

2 Extractors

Many probabilistic algorithms are designed under the assumption that the input distribution is uniform.

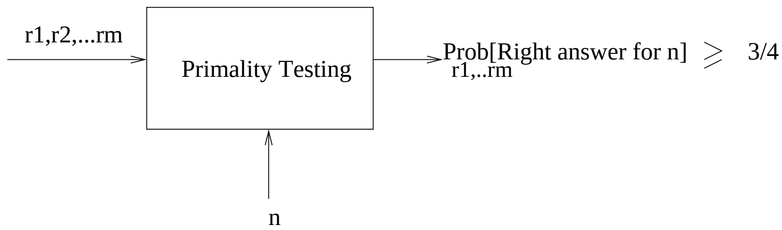

Figure 1: Ideal probabilistic algorithm

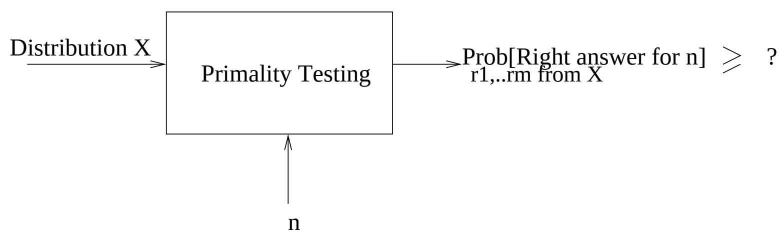

Figure 2: Real World

If the input distribution is not uniform, then no guarentees are given. But this might be the case in the real world. We might not know about the process giving the input to the algorithm. We want to transform a biased distribution to a uniform distribution. Its an open question as to which cases can be done.

A Random extractor attempts to make the input distribution uniform

DEFINITION 1 *A distribution D is ϵ-close to uniform if for every $T : \{0,1\}^m \rightarrow \{0,1\}$,* $|\mathbf{Pr}_{r_1,\ldots,r_m}[T(r_1,\ldots,r_m)] - \mathbf{Pr}[T(z)=1]| \;\leq\; \epsilon$

REMARK 1 If algorithm A succeeds with probability p using a uniform distribution then it succeeds with probability at least $p-\epsilon$ under a distribution that is ϵ-close to uniform.

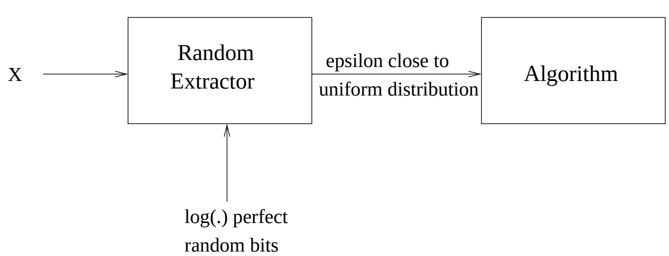

Figure 3: Random Extractors

DEFINITION 2 *A distribution D over $\{0,1\}^n$ has min entropy k if for every $z \in \{0,1\}^n$, $\mathbf{Pr}[D = z] \ \leq \ 1/2^k$.*
Min entropy of D $=$ $min_{z\in\{0,1\}^n} \log(1/\mathbf{Pr}[D = z])$

DEFINITION 3 *$E:\ \{0,1\}^m \times \{0,1\}^t \to \{0,1\}^m$ is a (k, ϵ) extractor if for every ditribution D with min entropy at least k, $E(D, z)$ is ϵ-close to uniform where z is uniform in $\{0,1\}^t$.*

REMARK 2 We need some restrictions on the input distribution. The output must contain approximately m random bits. t units of randomness can come from the second argument. So the input distribution should contain $m - t$ units of randomness. Randomness cannot be created out of nothing

REMARK 3 Why did we choose min-entropy? Consider the shannon entropy defined as $H(D) = \sum_z \mathbf{Pr}[D = z] \log(\frac{1}{\mathbf{Pr}[D=z]})$. Now consider a ditribution X, where $X = \{0, 0, \ldots, 0\}$ with probability $1 - 1/\sqrt{n}$ and is uniform in $\{0,1\}^n$ with probability $1/\sqrt{n}$. The shannon entropy of X is approximately $\sqrt{n}$. However X is not usable to produce a uniform distribution. No extractor can generate a uniform distribution as it would have to be $1/\sqrt{n}$ close to uniform when given all $X = 0$.

So it seems if we want to create something uniform, then we need high entropy most of the times. This is approximately equal to having high entropy all the time, which is the notion of min entropy.