6.842 Randomness and Computation MONTH DAY, 2012

Lecture 24

Lecturer: Ronitt Rubinfeld *Scribe: Santiago Cuellar*

1 Pseudorandom Generators and Next bit predictability

We saw the definition of a pseudorandom generator before

Definition 1 *A function $G : \{0,1\}^{\ell(n)} \rightarrow \{0,1\}^n$ is a* pseudorandom generator *if*

- $\ell(n) < n$
- *G is computationally indistinguishable from the uniform random distribution.*

We say a pseudorandom generator is efficient if the function can be computed in polynomial time. This are the kind of generators we are interested.

1.1 Next bit predictability

As we will see, computationally indistinguishability is tightly related with the notion of next bit predictability, which we define bellow

Definition 2 *A sequence $X = x_1 x_2 \dots x_n$ is* next bit predictable *if for all polynomial time algorithm P there is a negligible function $\epsilon(n)$ such that*

$$Pr_{X_i \in [n]} \left[P(x_1, x_2 \dots x_{i-1}) = x_i \right] > \frac{1}{2} + \epsilon$$

Theorem 3 *X is a pseudorandom generator if and only if it is next bit unpredictable.*

Proof *We proved one side of the theorem last class. Then we will prove that if X is next bit unpredictable then it is also a pseudorandom generator. To do so we will show the contrapositive. So, let X be a generator, but not pseudorandom. Then there is a polynomial time algorithm T such that*

$$\left| Pr_X \left[T(X) = 1 \right] - Pr_U \left[T(U) = 1 \right] \right| > \frac{1}{n^k}$$

for some k and infinitely many n's. Notice that there is a polynomial time algorithm $\bar{T}$ that always returns the opposite of T. By considering either of them we can suppose, without loss of generality that

$$Pr_X \left[T(X) = 1 \right] - Pr_U \left[T(U) = 1 \right] \geq \frac{1}{n^k}$$

. In the following we will use a very useful trick know as the hybrid argument*. We define the following hybrid sequences where the u_i's are taken from the uniform random distribution and the x_i's from the generator X.*

$$D_0 = u_1 u_2 \dots u_n \qquad = U \tag{1}$$
$$D_1 = x_1 u_2 \dots u_n \tag{2}$$
$$D_2 = x_1 x_2 \dots u_n \tag{3}$$
$$\vdots \tag{4}$$
$$D_n = x_1 x_2 \dots x_n \qquad = X \tag{5}$$

Now we consider the probabilities of any of the sequences of passing the test T. We use a telescoping sum to get the inequality.

$$\frac{1}{n^k} < Pr_{X\in D_n}\left[T(X)\right] - Pr_{X\in D_0}\left[T(X)\right] \tag{6}$$

$$< \sum_{i=1}^{n} (Pr_{X\in D_i}\left[T(X)\right] - Pr_{X\in D_{i-1}}\left[T(X)\right]) \tag{7}$$

$$\tag{8}$$

Then one of the differences in the sum, has to be larger than the average. That is, there is an i such that

$$\frac{1}{n^{k+1}} < (Pr_{X\in D_i}\left[T(X)\right] - Pr_{X\in D_{i-1}}\left[T(X)\right])$$

With this inequality in mind we define a predictor algorithm P:

- *Chose $u_i, u_{i+1} \dots u_n \in \{0,1\}^{n-1}$*
- $b \leftarrow T(x_1, x_2, \dots x_{i-1}, u_i \dots u_n)$
- *If $b = 1$ output u_i. Otherwise output $\bar{u}_i$.*

Note that $P(x_1, x_2 \dots x_{i-1}) = x_i$ exactly in the two cases

- *$b = 1$ and $u_i = x_i$*
- *$b = 0$ and $u_i \neq x_i$*

Lets reconsider the inequalities. Let Q be the random variable that determines if $P(x_1 \dots x_{i-1}) = x_i$ then

$$Pr\left[Q\right] = Pr\left[Q|u_i = x_i\right] Pr\left[u_i = x_i\right] + Pr\left[Q|u_i \neq x_i\right] Pr\left[u_i \neq x_i\right] \tag{9}$$

$$= \frac{1}{2} Pr\left[Q|u_i = x_i\right] + \frac{1}{2} Pr\left[Q|u_i \neq x_i\right] \tag{10}$$

$$= \frac{1}{2}(Pr\left[b = 1|u_i = x_i\right] + Pr\left[b = 0|u_i \neq x_i\right]) \tag{11}$$

$$= \frac{1}{2}(Pr\left[b = 1|u_i = x_i\right] + 1 - Pr\left[b = 1|u_i \neq x_i\right]) \tag{12}$$

$$= \frac{1}{2} + \frac{1}{2}(Pr\left[b = 1|u_i = x_i\right] - Pr\left[b = 1|u_i \neq x_i\right]) \tag{13}$$

$$= \frac{1}{2}(Pr\left[T(x_1 x_2 \dots x_i u_{i+1} \dots u_n)\right] - Pr\left[T(x_1 x_2 \dots \bar{x}_i u_{i+1} \dots u_n)\right]) \tag{14}$$

Now notice that we have the following relation:

$$Pr\left[T(x_1 \dots x_{i-1} u_i \dots u_n)\right] = \frac{1}{2} Pr\left[T(x_1 \dots x_i u_{i+1} \dots u_n)\right] + \frac{1}{2} Pr\left[T(x_1 \dots \bar{x}_i u_{i+1} \dots u_n)\right]$$

Which is equivalent to

$$Pr\left[T(x_1 \dots \bar{x}_i u_{i+1} \dots u_n)\right] = 2Pr\left[T(x_1 \dots x_{i-1} u_i \dots u_n)\right] - Pr\left[T(x_1 \dots x_i u_{i+1} \dots u_n)\right] + \frac{1}{2}$$

*Replacing this in equation (**??**) we get*

$$Pr\left[Q\right] = \frac{1}{2}(Pr\left[T(x_1 x_2 \dots x_i u_{i+1} \dots u_n)\right] - Pr\left[T(x_1 x_2 \dots \bar{x}_i u_{i+1} \dots u_n)\right]) \tag{15}$$

$$= \frac{1}{2}(2Pr\left[T(x_1 x_2 \dots x_i u_{i+1} \dots u_n)\right] - 2Pr\left[T(x_1 \dots x_{i-1} u_i \dots u_n)\right] \tag{16}$$

$$= (Pr_{X\in D_i}\left[T(X)\right] - Pr_{X\in D_{i-1}}\left[T(X)\right]) \tag{17}$$

$$> \frac{1}{n^{k+1}} \tag{18}$$

That is, our predictor succeeds with non-negligible probability. Thus X is not next bit unpredictable. ■

2 Introduction to One Way functions

Definition 4 *A function f is* one way *if*

- *f is computable in deterministic polynomial time.*
- *For every probabilistic polynomial time algorithm A, there is a negligible function $\epsilon(n)$ such that for large enough n*
$$Pr_X\left[A(f(x)) \in f^{-1}(f(x))\right] \leq \epsilon(n)$$

2.1 One way function cadidates

- *Multiplication: $f(x, y) = xy$. For x, y large prime numbers.*
- *RSA: $f_{m,e}(x) = x^e \bmod p$ for a prime p.*
- *Rabin's function: $f_m(x) = x^2 \bmod p$ for a prime p.*
- *Discrete log: $f_{p,g} = g^x \bmod p$.*

Theorem 5 (Hill) *Pseudorandom generators exist if and only if one way functions exist.*

We will not prove this theorem. But in next class we will show a weaker version.

Theorem 6 *If a permutation one way function exists, then there are efficient pseudorandom generators.*