Instructor: Andrej Bogdanov **Notes by:** Wei Yu

It the last lecture we showed that if one-way permutations exist, then so do pseudorandom generators from n to $n+1$ bits. Now we show that these can be used to go from n bits to $m(n)$ bits. Then we use these generators to construct another objest called a "pseudorandom function". A pseudorandom function from $\{0,1\}^n$ to $\{0,1\}^n$ is a function that to any polynomial-size circuit appears completely random, yet takes only n random bits to specify – as opposed to a random function, which requires many more random bits.

1 Pseudorandom generators with longer output

We now show how, given a pseudorandom generator $G : \{0,1\}^n \to \{0,1\}^{n+1}$, we can get $G' : \{0,1\}^n \to \{0,1\}^{m(n)}$ for arbitrary polynomial $m(n)$. For simplicity, we will only show the argument for $m(n) = n+2$. It will then be clear how this construction can be bootstrapped to obtain larger values for $m(n)$.

Let $G : \{0,1\}^n \to \{0,1\}^{n+1}$ be an arbirtary pseudorandom generator. Write $G(x) = (g(x), b(x))$, where $g(x)$ are the first n output bits of $G(x)$, and $b(x)$ is the last bit. We will show that the function $G' : \{0,1\}^n \to \{0,1\}^{n+2}$ given by

$$G'(x) = (G(g(x)), b(x)) = (g(g(x)), b(g(x)), b(x))$$

is a pseudorandom generator. What happens here is that we apply $G(x)$, save the last bit of the output, then apply G again on the first n bits. We can continue the same procedure to

Theorem 1. *If G is a pseudorandom generator then so is G'.*

Proof. To argue that G' is pseudorandom, let's look at the following picture, which shows how the output of G' is obtained:

$$x \xrightarrow{G} G(x) = (g(x), b(x)) \xrightarrow{G_1} (G(g(x)), b(x))$$

here, $G_1 : \{0,1\}^{n+1} \to \{0,1\}^{n+2}$ is the function that applies G to the first n bits of input and preserves the last bit.

To argue that the output of G' is indistinguishable from a random string of length $n+2$, it will be convenient to think "in parallel" what happens when G_1 is applied to a uniformly random string of length $n+1$:

$$\begin{array}{rcll} y_n & \xrightarrow{G} & G(y_n) = (g(y_n), b(y_n)) & \xrightarrow{G_1} \quad G'(y_n) = (G(g(y_n)), b(y_n)) \\ & & (y_n, y_1) & \xrightarrow{G_1} \qquad\qquad (G(y_n), y_1) \\ & & & \qquad\qquad (y_{n+1}, y_1) \end{array}$$

Here, y_i is a random string of length i. Now suppose that some circuit C can ϵ-distinguish between the distributions $G'(y_n)$ and $y_{n+2} = (y_{n+1}, y_1)$. Then C can either $\epsilon/2$-distinguish between $G'(y_n)$ and $(G(y_n), y_1)$, or it can $\epsilon/2$-distinguish between $(G(y_n), y_1)$ and (y_{n+1}, y_1).

In the first case, the following circuit C' can $\epsilon/2$-distinguish $G(y_n)$ from y_{n+1}: On input z, output $G_1(z)$. In the second case, we can use the following randomized circuit C'': On input z, choose a random $y_1 \sim \{0,1\}$ and output $C(z, y_1)$. Either way, the output of G is $\epsilon/2$-distinguishable from a random output by a circuit whose size is larger than C by at most $\text{poly}(n)$. □

One idea in this proof is that if two distributions X and Z are distinguishable, then for any distribution Y either X and Y are distinguishable or Y and Z are distinguishable. To turn one type of distinguisher into another, it is often helpful to introduce one or several intermediate distributions Y which will make the problem more closely related to what we are aiming at. This is known as "the hybrid argument", as the distribution Y typically represents some sort of hybrid of X and Z. We already saw one application of the hybrid argument in Lecture 9 on the Nisan-Wigderson generator.

For our next application, the construction of pseudorandom functions, it will be convenient to use an alternative definition of pseudorandom generator. In the definition we used, the generator obtains one sample, and based on this one sample it is supposed to tell if the sample looks random or pseudorandom. However, we can also consider a distinguisher that obtains as many samples as it wants to have, and we expect the distinguishing probability to remain small. This is indeed the case.

Theorem 2. *If $G : \{0,1\}^n \to \{0,1\}^{m(n)}$ is a pseudorandom generator, then for any polynomial-size family of oracle circuits C that queries $q(n)$ strings of length $m(n)$ and every polynomial $p(n)$*

$$\left|\Pr_{y_1,\ldots,y_q \sim \{0,1\}^m}[C^{y_1,\ldots,y_q}(1^n) = 1] - \Pr_{x_1,\ldots,x_q \sim \{0,1\}^m}[C^{G(x_1),\ldots,G(x_q)}(1^n) = 1]\right| < 1/p(n).$$

Proof. Let $\epsilon = 1/p(n)$. Suppose there is some C such that

$$\left|\Pr_{y \sim \{0,1\}^{m(n)}}[C^{y_1,\ldots,y_q}(1^n) = 1] - \Pr_{x \sim \{0,1\}^n}[C^{G(x_1),\ldots,G(x_q)}(1^n) = 1]\right| \geq \epsilon.$$

Consider the distribution $H_i = (G(x_1), \ldots, G(x_{i-1}), y_i, \ldots, y_q)$, where the first $i-1$ strings are random, and the last $q-i$ of them are pseudorandom, all of them independent. Then the above inequality says that

$$\left|\Pr_{h \sim H_0}[C^h(1^n) = 1] - \Pr_{h \sim H_q}[C^h(1^n) = 1]\right| \geq \epsilon.$$

There must exist a value $0 \leq i < q$ such that

$$\left|\Pr_{h \sim H_i}[C^h(1^n) = 1] - \Pr_{h \sim H_{i+1}}[C^h(1^n) = 1]\right| \geq \epsilon/q.$$

Fix choices of $x_1, \ldots, x_{i-1}, y_{i+1}, \ldots, y_q$ that maximize this probability. Then

$$\left|\Pr_{y_i}[C^{G(x_1),\ldots,y_i,\ldots,y_q}(1^n) = 1] - \Pr_{x_i}[C^{G(x_1),\ldots,G(x_i),\ldots,y_q}(1^n) = 1]\right| \geq \epsilon/q.$$

Then the circuit $C'(z) = C^{G(x_1),\ldots,G(x_{i-1}),z,y_{i+1},\ldots,y_q}(1^n)$ will ϵ/q-distinguish y_i and $G(x_i)$, which is a contradiction. □

2 Pseudorandom functions

We now define pseudorandom functions. For this, we want to think of a function $F : \{0,1\}^k \to \{0,1\}^n$ as a table of values $F(x) \in \{0,1\}^n$, one for every $x \in \{0,1\}^k$. A random function R has the property that all the values $R(x)$ are uniformly distributed in $\{0,1\}^k$ and mutually independent. To specify such a function, we need to choose 2^k random values $R(x)$ uniformly and independently from $\{0,1\}^n$, so the construction requires $n \cdot 2^k$ random bits. A pseudorandom function is one that looks like a random function to any polynomial-size circuit family, yet can be specified using only n random bits.

Definition 3 (Pseudorandom Function)**.** *A pseudorandom function is a family of functions $F_z : \{0,1\}^{k(n)} \to \{0,1\}^n$ ($k(n) = \text{poly}(n)$), where $z \in \{0,1\}^n$ such that on input (z,x), the value $F_z(x)$ is polynomial-time computable, and for every polynomial-size family of oracle circuits C and every polynomial p:*

$$\left|\Pr_{z\sim\{0,1\}^n}\left[C^{F_z}(1^n)=1\right] - \Pr_{R\sim\{0,1\}^k}\left[C^{R}(1^n)=1\right]\right| < \frac{1}{p(n)}$$

where R is a random function.

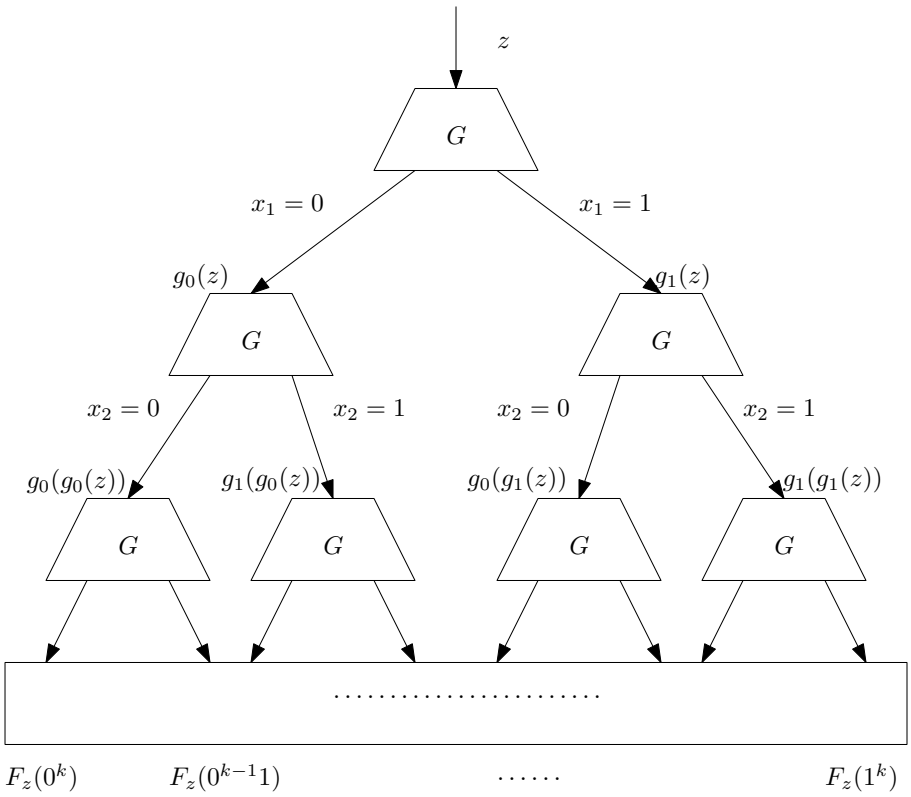

Figure 1: Construction of a pseudorandom function

Theorem 4. *If a pseudorandom generator $G : \{0,1\}^n \to \{0,1\}^{2n}$ exists, then a pseudorandom function exists for any $k(n) = \text{poly}(n)$.*

Proof. Let $G(z) = (g_0(z), g_1(z))$, where $g_0(z)$ and $g_1(z)$ represent the first and last k bits of $G(z)$, respectively. We will show that the family of functions

$$F_z(x) = g_{x_k}(\cdots g_{x_2}(g_{x_1}(z))\cdots)$$

is pseudorandom. The construction is showed in Figure 1.

Suppose that for some C and ϵ,

$$\left|\Pr_{z\sim\{0,1\}^n}\left[C^{F_z}(1^n)=1\right] - \Pr_{R\sim\{0,1\}^k}\left[C^{R}(1^n)=1\right]\right| \geq \epsilon.$$

We will use a hybrid argument to design a distinguisher that violates Theorem 2. To do this, we introduce the following family of hybrid semi-random functions $H_0, \ldots, H_k$:

H_i: Choose a random function $R : \{0,1\}^i \to \{0,1\}^n$
On input x, output the value $g_{x_k}(\cdots g_{x_{i+1}}(R(x)))$.

Notice that H_0 is exactly the distribution F_z (in this case R is a constant string of length n), while H_k is a random function. For instance, for $k = 3$, the hybrids H_0, H_2, and H_3 are illustrated in Figure 2.

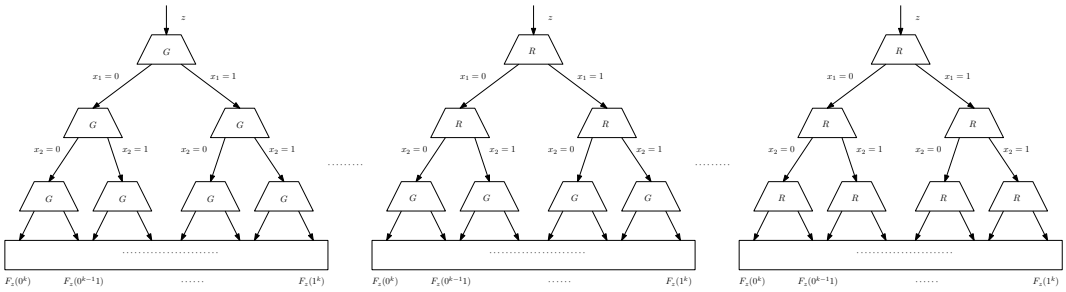

Figure 2: Hybrid functions. Here R represents a random function.

By the hybrid argument, there must exist an index i such that

$$\left|\Pr[C^{H_{i-1}}(1^n)=1] - \Pr[C^{H_i}(1^n)=1]\right| \geq \epsilon/k.$$

Recall that our aim is to construct an oracle circuit C' such that

$$\left|\Pr[C'^{y_1,\ldots,y_q}=1] - \Pr[C'^{G(x_1),\ldots,G(x_q)}]=1]\right| \geq \epsilon'$$

for some $\epsilon' = \text{poly}(\epsilon, 1/n)$. To do this, notice that the functions H_{i-1} and H_i differ only in what happens at "level" i. In H_{i-1}, the inputs chosen at this level look like the outputs of G, while in H_i they look random. So if we can distinguish between H_{i-1} and H_i, we should be able to distinguish random and pseudorandom strings of length $2n$ (see figure 3).

The distinguisher C' will do the following:

$C'(1^n)$: Let $z_j = (z_{j0}, z_{j1})$ denote the jth query of C ($z_{j0}, z_{j1} \in \{0,1\}^n$).
Simulate $C(1^n)$. When $C(1^n)$ makes its jth query x_j,
If no query x_l such that $\overline{x_{l1} \dots x_{l(i-1)}} = \overline{x_{j1} \dots x_{j(i-1)}}$ was made before ($l < j$)
Pretend that the answer to query x_i is $g_{x_k}(\cdots g_{x_{i+1}}(z_{jx_i}) \cdots)$.
Otherwise,
Pretend that the answer to query x_i is $g_{x_k}(\cdots g_{x_{i+1}}(z_{lx_i}) \cdots)$,
where l is the smallest index of a query with the above property.
Return the output of $C(1^n)$.

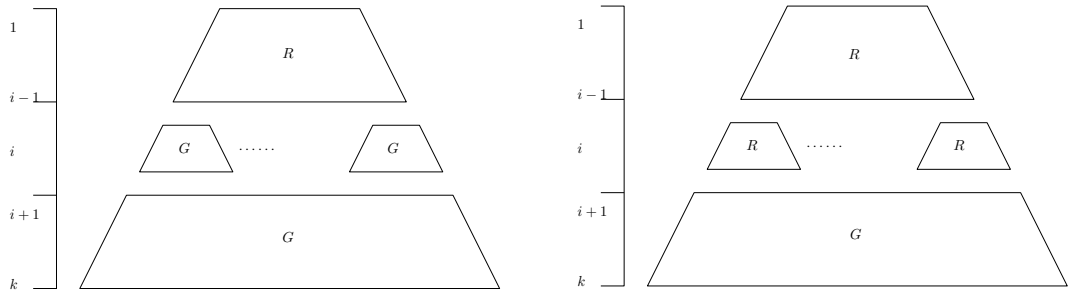

Figure 3: Hybrid argument

To see what is happening, consider the partial function $P : \{0,1\}^i \to \{0,1\}^n$ defined as follows: If x_j is the jth query of $C(1^n)$, then $P(\overline{x_{j1} \cdots x_{ji}}) = z_{lx_{ji}}$, where l is the smallest index of a query such that $\overline{x_{l1} \dots x_{l(i-1)}} = \overline{x_{j1} \dots x_{j(i-1)}}$. When z_j are uniformly random strings of length $2n$, then $P(\overline{x_{j1} \cdots x_{ji}})$ behaves like random function from $\{0,1\}^i$ to $\{0,1\}^n$. When $z_j = G(u_j), u_i \sim \{0,1\}^n$, then $P(\overline{x_{j1} \cdots x_{ji}}) = g_{x_i}(u_l)$, so P behaves like g_{x_i} applied to the output of a random function. In the first case, the queries made by C follow the distribution H_i, and in the second case they follow the distribution H_{i+1}. It follows that

$$\Pr[C'^{y_1,\dots,y_q}(1^n) = 1] - \Pr[C'^{G(x_1),\dots,G(x_q)}(1^n) = 1]$$
$$= \Pr[C^{H_{i-1}}(1^n) = 1] - \Pr[C^{H_i}(1^n) = 1] \geq \epsilon/q. \qquad \square$$