CS 880: Pseudorandomness and Derandomization	4/3/2013
Lecture 19: Pseudorandomness for Regular Branching Programs	
Instructors: Holger Dell and Dieter van Melkebeek	Scribe: Alexi Brooks

DRAFT

1 Condenser construction

In the previous lecture, we began to present how to construct a condenser. In principle, a condenser is a function which takes as input an element from a weakly random source distribution and presents as output an element of another distribution with the same min entropy but much shorter length.

Consider a bipartite graph G_c consisting of two independent vertex sets V_1 and V_2. The vertices in V_1 are members of $\mathbb{F}_q^a$, vectors of length a over $\mathbb{F}_q$. We will interpret them as polynomials, so that we can describe the condenser as a function $C : \mathbb{F}_q[y] \to F$. The variables y correspond to the edges from V_1 to V_2, and F is a function on y. The vertices in V_2 are then defined as

$$c(F, y) = (y, F_0(y), \dots, F_{b-1}(y))$$

where $F_i(y) = F(y)^{h^i} \mod E(y)$. $E(y)$ is an arbitrary irreducible polynomial with degree a. This graph is used to generate a condenser with parameters a, b, h, and q.

G_c is a (K, A)-vertex expander with $K = h^b$ and $A = q - ahb$. □

Every set in V_1 with at most K elements has at least AK neighbors in V_2. We will look at this from the V_2 side:

Proof. Let $T \subseteq V_2$ with $|V_2| < AK$, and let $S = \{F \in V_1 | \Gamma(F) \subseteq T\}$. We need to show $|S| < K$. If we can do that, we will know that G_c is a vertex expander and therefore a condenser. We construct a polynomial of degree at most K where every element in S is a root.

Let $Q(y, Z_0, Z_1, \dots, Z_{b-1}$ be a polynomial over $\mathbb{F}_q$ with $Q(t) = 0 \forall t \in T$, and assume Q is not identically zero. If the degree of Q is large enough, some polynomial meeting these characteristics must exist. All $t \in T$ are roots of this polynomial. We satisfy the following two linear constraints.

- $\deg_y(Q) \leq A - 1$
- $\deg_z(Q) \leq h - 1$

The solution has at least Ah^b monomials, implying $AK > |T|$.

Recall that we want Q to vanish on all S. Let $F \in S$. We know that $\forall y \in \mathbb{F}_q$, $c(F, y) \in T$. Because of how we constructed Q, this implies that $\forall y \in \mathbb{F}_q$, $Q(c(F, y)) = 0$. If we vary the parameter y (calling this variable Y), then we have $Q(c(F, Y)) \equiv 0$. We achieve this by setting the condenser parameter A such that the q possible choices for a value of Y, are more than the degree of Q. $q > \deg(Q) = A - 1 + (h - 1)ba$.

We now have the equation $Q(y, F^{h^0}(y), \dots, F^{h^{b-1}(y)} \equiv 0$, but the degree of this polynomial is too high. We can reduce it by taking the mod of a polynomial $E(y)$. As noted before, the only characteristic of this polynomial that concerns us is the fact that it is irreducible in y.

We then select a polynomial $Q'(z)$ which vanishes on S, is univariate, and whose coefficients are polynomial in y. We define this polynomial as

$$Q'(z) = Q(y, z^{h^0}, z^{h^1}, \dots, z^{h^{b-1}}) \mod E(y)$$

Note that $Q'(z) \in (\mathbb{F}_q[y]/E(y))[z]$. In other words, Q' is a polynomial ring. $\forall F \in S$, we have $Q'(F(y)) = 0$, but $Q'(z) \not\equiv 0$.

Consider the degree of Q':

$$\deg_z(Q') \leq h - 1 + h(h-1) + h^2(h-1) + \ldots + h^{b-1}(h-1) = h^b - 1 = K - 1$$

Each term $h^i(h-1)$ comes from the z_i term in Q'. We get $h^b - 1$ from algebra, and $K - 1$ from our choice of K when we originally made the claim. Since Q' is not identically zero, yet $\forall F \in S$, $Q'(F(y)) = 0$, we know that $|S| \leq \deg(Q')$. Thus, $|S| < K$. □

2 Extractor-based PRGs

In this second part of the lecture, we consider the INW generator in its extractor form. Let

$$G_i : \{0,1\}^{is} * \{0,1\}^s \to \{0,1\}^{2^i}$$

and recall that we previously used this generator to prove that Undirected Connectivity is in logspace. The formula above presents the structure of this extractor with its input length is, seed length s, and output length 2^i. For variable x and some y such that $|y| << |x|$, we define $G_i(x)$:

- $G_0(x) = x_1$
- $G_i(x, y) = G_{i-1}(x)G_{i-1}(\text{Extr}(x, y))$

Extr is a family of extractors which are derived separately and which are known to function well as extractors for certain sources that are already mostly clean (containing a large number of bits of randomness compared to their length).

This function forms a recurrence relation matching the probability of acceptance for a branching program. More precisely, G_i is ϵ-pseudorandom for branching programs $B : V * \{0,1\} \to V$ if $|\Pr[B(s, U) \in \text{Acc}] - \Pr[B(s, G_i(x, y))]| \leq \epsilon$, where $is + s \approx i^2 = (\log n)^2$.

Starting from the final "layer", note that

$$\Pr[\text{Acc}, v] = \begin{cases} 1 & \text{if } v \text{ is in the accepting set} \\ 0 & \text{otherwise} \end{cases}$$

In general, a node v in a binary branching program has probability of acceptance $a(v) = \frac{a(v_0)+a(v_1)}{2}$, where v_0 and v_1 are its child nodes depending on which branch we choose from this program state.

We define the value function $v(x = \{0,1\}^r)$ as the probability of acceptance assuming we have walked according to x from the start vertex. The following two statements are equivalent to this definition: $v(x) = a(B(s, x))$. $\text{E}[v_b(x, U)] = v_B(x)$.

We will be interested in finding a bound on the difference $|v(x) - v(y)|$ for x, y nodes in the branching program.

If x, y are distributions on $\{0,1\}^r$, then $|\mathrm{E}[v(x)] - \mathrm{E}[v(y)]| \leq d_{\mathrm{STAT}}(x,y) * \mathrm{weight}(B)$. The weight of a branching program is difference in value that can be obtained over a single branching step. $\mathrm{weight}(B) = \sum_{x \in \{0,1\}^r, b \in \{0,1\}} |v(x) - v(xb)|$. □

Proof. Consider x of length $|x| = r$. Define vmax and vmin as the maximum and minimum values for $v(x)$ with x of this length. There must be some path x leading from the start state to a vertex with $v(x)$ =vmax. Similarly, there must be some x leading from the start state to a vertex with $v(x)$ =vmin. Then we have $|vmax - vmin| \leq \mathrm{weight}(B)$. Let $p_a = \Pr[X = a]$ and $q_a = \Pr[Y = a]$, and we write:

$$\begin{aligned}
|\mathrm{E}[v(x)] - \mathrm{E}[v(y)]| &= \textstyle\sum_{a \in \{0,1\}^r} p_a v(a) - q_a v(a) \\
&\leq \textstyle\sum_{a, p_a > q_a} (p_a - q_a)\mathrm{vmax} + \sum_{a, p_a \leq q_a} (p_a - q_a)\mathrm{vmin} \\
&= (\mathrm{vmax} - \mathrm{vmin}) d_{\mathrm{STAT}}(x,y) \\
&\leq \mathrm{weight}(B) d_{\mathrm{STAT}}(x,y)
\end{aligned}$$

□

The next step to this extractor construction is to use this lemma to place a bound on the required seed length. We will need a tighter bound on the weight of a branching program, which we will cover in the next lecture.

References