CS 6815: Pseudorandomness and Combinatorial Constructions **Fall 2022**

Lecture 6: Sep 08, 2022

Lecturer: Eshan Chattopadhyay *Scribe: Jenny Chen*

6.1 Overview

Today, we want to define "Dream Pseudorandom Generator (PRG)" and see how such a "Dream PRG" would immediately imply BPP = P.

6.2 Boolean Circuits

First, we define the Boolean circuits (analogous to algebraic circuits defined in a previous lecture). A Boolean circuit is a DAG such that its leaves represent input nodes, and contains an output node with 0 out-degree. For internal nodes (or the operators), we have $\wedge$, $\vee$, and $\neg$. Similar to algebraic circuit, the size of the circuit is the number of edges/wires, and the depth is the longest path from root to leaf. Fan-in and fan-out correspond to the in-degrees and out-degrees of nodes respectively.

Illustration of a Boolean circuit.

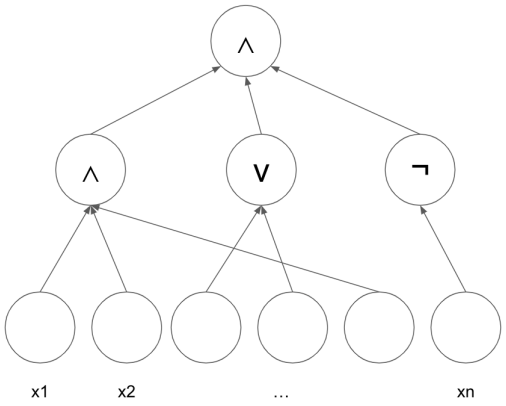

Figure 1: An example of a boolean circuit.

It does not make a lot of sense if we just say the circuit C computes $L \subseteq \{0,1\}^*$ since one circuit can only take in inputs of a certain length. Therefore, we introduce a modified definition.

Definition 6.1. *Circuit **family** $\{C_n\}_{n\geq 0}$ computes $L \subseteq \{0,1\}^*$ if $\forall n \geq 0, \forall x \in \{0,1\}^n, C_n(x) = L(x)$ (where $L(x)$ is 1 iff $x \in L$).*

Definition 6.2. *Define SIZE($S(n)$) to be the complexity class that contains all language L such that there exists some circuit families $\{C_n\}$ computing L, and $\forall n \geq 0$, size($C(n)$) $\leq S(n)$.*

Definition 6.3. *P/poly* $= \bigcup_{c\geq 0} SIZE(n^c)$

Thus, it's the complexity class that contains any language L that can be computed by a polynomial size circuit family. It is known that $P \subsetneq P/\text{poly}$.

Definition 6.4. *DTIME($T(n)$) is the complexity class that contains all language L such that there exist some Turing Machine that can decide L in $T(n)$ time.*

We need the following result of translating Turing machines to Boolean circuits.

Theorem 6.5. *Suppose* $L \in DTIME(T(n)) \Rightarrow L \in SIZE(T(n)log(T(n))$

6.3 Dream PRG and BPP = P

Definition 6.6. *Dream PRG: let* $\{G_n : \{0,1\}^{S(n,m)} \to \{0,1\}^n\}_{n\geq 0}$ *be an* ϵ*-PRG against SIZE(*$m(n)$*) with* $S(n,m) = O(log(m/\epsilon))$*, for all* $m(n)$ *= poly(n).*

Theorem 6.7. *If we have a dream PRG* $\Rightarrow BPP = P$.

This theorem is equivalent to, having a dream PRG that has a seed length of $O(logn)$ against polynomial size circuits, we can derandomize all BPP algorithms.

Proof. Let $L \in BPP$,
$\Rightarrow \exists M$, such that $\forall n \geq 0, \forall x \in \{0,1\}^n, \Pr[M(x,r) = L(x)] \geq \frac{2}{3}$ (from definition of BPP)
We construct an algorithm A as the following: Given input $x \in \{0,1\}^n$

- Let $G_n : \{0,1\}^{s(n,m)} \to \{0,1\}^n$ be a 0.1-PRG against $m = n^{2c}$ circuit.
- Run M on $(x, r_i), \forall i \in \{1, 2, ..., 2^{s(n,m)}\}$ (cycle over all seeds to G_n)
- Output majority

We know that $s(n,m) = c'log(m)$. So there're a total of $2^{s(n,m)} = O(n^{c'})$ seeds. The runtime for each input is polynomial time, say $O(n^c)$. Therefore, the total runtime for this algorithm is $O(n^{c+c'})$ where c comes from the runtime for each input and c' comes from the seeds (since we loop over all possible seeds). By using the seeds generated from the PRG, we reduce the randomness from $O(2^{n^c})$ to $O(2^{s(n,m)})$, which is from exponential to polynomial.

Therefore, we created a polynomial time algorithm A that desides L (but here we assume the calculation of G_n is polynomial time as well).

Then we prove the correctness of A: We want to show that for a fixed input $x \in \{0,1\}^n$, $A(x) = L(x)$.

Since the input x is fixed and in each repetition the only difference is the seed r we provide, we can define $M_x(r) = M(x,r)$ where M_x is a Turing Machine that only takes in r and give the same result as $M(x,r)$. Since we know the runtime of M_x is $T(n) = n^c$, from theorem (6.5), we know that there exists some $\{C_n\}$ with SIZE$(C_n) \leq c'n^c(logn)$ such that $C_n(r) = M_x(r)$.
$\Rightarrow \Pr_{r\sim\{0,1\}^{n^c}}[M_x(r) = L(x)] \geq \frac{2}{3} \Rightarrow \Pr_{r\sim\{0,1\}^{n^c}}[C_n(r) = L(x)] \geq \frac{2}{3}$ ①
And we know $\Pr_{r\sim\{0,1\}^{n^c}}[C_n(r) = 1] - \Pr_{t\sim\{0,1\}^{s(m,n)}}[C_n(G_n(t)) = 1] \leq 0.1$ ② from definition of 0.1-PRG
① + ② $\Rightarrow \Pr_{t\sim\{0,1\}^{s(m,n)}}[C_n(G_n(t)) = L(x)] \geq \frac{2}{3} - 0.1 > \frac{1}{2}$
$\Rightarrow \Pr_{t\sim\{0,1\}^{s(m,n)}}[M(x, G_n(t)) = L(x)] > \frac{1}{2}$
Since in the algorithm we take the majority, we will get the correct answer for sure. □

6.4 Existence of Dream PRGs (probabilistic/non-constructive)

Although we cannot give a specific dream PRG, we can still show that the probability that there exists some dream PRGs is very high.

Theorem 6.8. $\exists G = \{0,1\}^{s(m,\epsilon)} \to \{0,1\}^n$ *that* ϵ*-fools SIZE(m) circuits, with* $s(m,\epsilon) = O(log(m/\epsilon))$.

Note since we won't use any special properties of n, this can generalize to any n.

Proof. Pick a random function G. Let's say G takes in an input of length $s(m, \epsilon)$ and picks a random output of length n uniformly. Let $C \in \text{SIZE}(m)$

The idea of the proof is to show that $\Pr[G \text{ does not } \epsilon\text{-fool } C \text{ (bad events)}] \leq \delta$. Then from union bond $\Pr[G \text{ is an } \epsilon\text{-PRG for size}(m)] \geq 1 - \delta * |\text{size(m)}|$ ①, and we want to pick a δ that can make this probability greater than 0 ($\text{size}(m)$ is at most $2^{\text{poly}(m)}$ here).

The bad events can be represented as $|\mathbb{E}_{t\sim\{0,1\}^{s(m,\epsilon)}}[C(G(t))] - \mathbb{E}_{r\sim\{0,1\}^n}[C(r)]| > \epsilon$, where the function G we picked cannot fool the circuits. Since G is also a random variable, we want to show that $\Pr_G(|\mathbb{E}_{t\sim\{0,1\}^{s(m,\epsilon)}}[C(G(t))] - \mathbb{E}_{r\sim\{0,1\}^n}[C(r)]| > \epsilon) < \delta$ ②.

$\mathbb{E}_{t\sim\{0,1\}^{s(m,\epsilon)}}[C(G(t))] = \frac{1}{2^{s(m,\epsilon)}} \sum_{i=1}^{2^{s(m,\epsilon)}} C(Y_i), Y_i = G(t_i)$. Y_i's are the output of the generators from different seeds t_i. Since Y_i's are all sampled uniformly by G, they are all iid. Therefore, $\forall i, \mathbb{E}[C(Y_i)] = \mathbb{E}_{r\sim\{0,1\}^n}[C(r)]$. So ② can be interpreted as: given a true mean $\mathbb{E}_{r\sim\{0,1\}^n}[C(r)]$ and we sample $2^{s(m,\epsilon)}$ times independently, the probability that we're too far off. Using the Chernoff Bounds here, $\delta = 2^{-c\epsilon^2 2^{s(m,\epsilon)}}$. And we want $\delta = 2^{-c\epsilon^2 2^{s(m,\epsilon)}} < 2^{-m^{c'}}$ to make ① > 0.
$\Rightarrow 2^s(m, \epsilon) = c'' m^{c'} * \frac{1}{\epsilon^2}$
$\Rightarrow s(m, \epsilon) = O(log(m/\epsilon))$ □

With some careful chosen constants (allowing large enough seed lengths), we can show that most random functions are dream PRGs (but of course, explicitly constructing such a PRG remains a daunting task!)