Lecture 10 - Derandomization, Pseudorandom Generators (PRGs)

Rafael Oliveira
rafael.oliveira.teaching@gmail.com
University of Waterloo

CS 860 - Graduate Complexity Theory
Fall 2022

Overview

- Pseudorandom Generators (PRGs)
- Unpredictability vs Randomness & PRGs from Hard Functions

Derandomization

- Derandomization is the process of “removing randomness” from PTMs
  - Sometimes term is used to simply refer to a deterministic algorithm for the same problem
  - In this case, just says that language $L \in \mathsf{P}$

Derandomization

- Derandomization is the process of “removing randomness” from PTMs
  - Sometimes term is used to simply refer to a deterministic algorithm for the same problem
  - In this case, just says that language $L \in \mathsf{P}$
- Is there a general way to (non-trivially) remove randomness from BPP machines?

$$\mathsf{BPP} \subseteq^? \mathsf{SUBEXP} := \bigcap_{\varepsilon > 0} \mathsf{DTIME}(2^{n^\varepsilon})$$

Derandomization

- Derandomization is the process of "removing randomness" from PTMs
  - Sometimes term is used to simply refer to a deterministic algorithm for the same problem
  - In this case, just says that language $L \in \mathsf{P}$
- Is there a general way to (non-trivially) remove randomness from BPP machines?

$$\mathsf{BPP} \subseteq^{?} \mathsf{SUBEXP} := \bigcap_{\varepsilon > 0} \mathsf{DTIME}(2^{n^{\varepsilon}})$$

- To do the above, cannot use PTMs as a black-box. That is, general derandomization cannot relativize

See literature in **[Pap 1994]**

Derandomization

- Also know that lower bounds cannot relativize.

  Could we use (strong enough) lower bounds to derandomize BPP?

Derandomization

- Also know that lower bounds cannot relativize.

  Could we use (strong enough) lower bounds to derandomize BPP?

- In a sense reduce use of non-relativization to proving lower bounds.[1]

[1] Though admittedly it could be that on the way to prove lower bounds, our non-reltivizing technique also works against BPP, in which case all of the below will be sort of redundant.

Derandomization

- Also know that lower bounds cannot relativize.

  Could we use (strong enough) lower bounds to derandomize BPP?

- In a sense reduce use of non-relativization to proving lower bounds.
- Still interesting that hardness can imply randomness, as we are now using reductions to prove:

  some impossible result $\Rightarrow$ possible result!

  Usually a reduction $A \leq B$ is used to show that B tractable then A tractable or conversely A intractable then B intractable

Pseudorandom Generators

Definition 1 (Pseudorandom Distributions)

A distribution R over $\{0,1\}^m$ is (s,ε)-pseudorandom if for every circuit C such that $S(C) \leq s$

$$|\Pr[C(R)=1] - \Pr[C(U_m)=1]| < \varepsilon$$

where U_m is the uniform distribution over $\{0,1\}^m$.

Pseudorandom Generators

Definition 1 (Pseudorandom Distributions)

A distribution R over $\{0,1\}^m$ is (s,ε)-pseudorandom if for every circuit C such that $S(C) \leq s$

$$|\Pr[C(R)=1] - \Pr[C(U_m)=1]| < \varepsilon$$

where U_m is the uniform distribution over $\{0,1\}^m$.

- We say that $G : \{0,1\}^\ell \to \{0,1\}^m$ is (s,ε)-pseudorandom if the distribution $G(U_\ell)$ is (s,ε)-pseudorandom.

Pseudorandom Generators

Definition 1 (Pseudorandom Distributions)

A distribution R over $\{0,1\}^m$ is (s, ε)-pseudorandom if for every circuit C such that $S(C) \leq s$

$$|\Pr[C(R) = 1] - \Pr[C(U_m) = 1]| < \varepsilon$$

where U_m is the uniform distribution over $\{0,1\}^m$.

Definition 2 (Pseudorandom Generators)

Let $s : \mathbb{N} \to \mathbb{N}$ be a time-constructible and non-decreasing function. A 2^n-time constructible string function $G : \{0,1\}^* \to \{0,1\}^*$ is an $s(\ell)$-pseudorandom generator if

- $|G(z)| = s(|z|)$ for all $z \in \{0,1\}^*$
- for every $\ell \in \mathbb{N}$, $G(U_\ell)$ is $(s(\ell)^3, 1/10)$ pseudorandom.[1]

[1] Constants 3 and $1/10$ chosen for convenience

PRGs and Derandomization

Proposition 3

$s(\ell)$-*PRG* $\Rightarrow$ *BPTIME*$(s(t(n))) \subseteq$ *DTIME*$(2^{ct(n)}s(t(n)))$ *for some constant* $c > 0$*, where* $t(n)$ *is a poly-time computable function.*

PRGs and Derandomization

Proposition 3

$s(\ell)$-PRG $\Rightarrow$ BPTIME$(s(t(n))) \subseteq$ DTIME$(2^{ct(n)}s(t(n)))$ for some constant $c > 0$, where $t(n)$ is a poly-time computable function.

1. $s(\ell) = 2^{\gamma\ell} \Rightarrow \mathsf{BPP} = \mathsf{P}$
2. $s(\ell) = 2^{\ell^{\gamma}}$ where $\gamma \in (0, 1)$ then $\mathsf{BPP} \subseteq \mathsf{DTIME}(2^{\mathsf{poly}\log n})$
3. if $s(\ell) = \ell^c$ then $\mathsf{BPP} \subseteq \mathsf{DTIME}(2^{n^{1/c}})$

PRGs and Derandomization

Proposition 3

$s(\ell)$-PRG $\Rightarrow$ BPTIME$(s(t(n))) \subseteq$ DTIME$(2^{ct(n)} s(t(n)))$ for some constant $c > 0$, where $t(n)$ is a poly-time computable function.

- Say $s(\ell) = 2^{\gamma\ell}$ and let $M \in \text{BPTIME}(n^c)$,
  $G_m : \{0,1\}^{\gamma^{-1}\log m} \to \{0,1\}^m$

PRGs and Derandomization

Proposition 3

$s(\ell)$-PRG $\Rightarrow$ BPTIME$(s(t(n))) \subseteq$ DTIME$(2^{ct(n)}s(t(n)))$ for some constant $c > 0$, where $t(n)$ is a poly-time computable function.

- Say $s(\ell) = 2^{\gamma\ell}$ and let $M \in \text{BPTIME}(n^c)$, $G_m : \{0,1\}^{\gamma^{-1}\log m} \to \{0,1\}^m$
- If M uses $m := m(n)$ random bits over $\{0,1\}^n$, then

$$\Pr_{r \in U_m}[M(x,r) = L(x)] \geq 2/3$$

PRGs and Derandomization

Proposition 3

$s(\ell)$-PRG $\Rightarrow$ BPTIME$(s(t(n))) \subseteq$ DTIME$(2^{ct(n)}s(t(n)))$ for some constant $c > 0$, where $t(n)$ is a poly-time computable function.

- Say $s(\ell) = 2^{\gamma\ell}$ and let $M \in \text{BPTIME}(n^c)$, $G_m : \{0,1\}^{\gamma^{-1}\log m} \to \{0,1\}^m$
- If M uses $m := m(n)$ random bits over $\{0,1\}^n$, then

$$\Pr_{r \in U_m}[M(x,r) = L(x)] \geq 2/3$$

- Given x, r, note that $M(x,r)$ is deterministic TM, hence (Proposition 2, Lecture 6), $M(x,r) \in \text{SIZE}(t^2(n))$, thus

$$|\Pr_{r \in G_m(U_\ell)}[M(x,r)] - \Pr_{r \in U_m}[M(x,r)]| < 1/10$$

- Then, if $\ell := \gamma^{-1}\log m$,

$$\Pr_{r \in G_m(U_\ell)}[M(x,r) = L(x)] > 2/3 - 1/10 > 5/9$$

PRGs and Derandomization

Proposition 3

$s(\ell)$-PRG $\Rightarrow$ BPTIME$(s(t(n))) \subseteq$ DTIME$(2^{ct(n)} s(t(n)))$ for some constant $c > 0$, where $t(n)$ is a poly-time computable function.

- The above shows why it's ok to let the PRG run in 2^{ℓ} time for inputs of length ℓ - for derandomization we will have to go over all seeds!

- Pseudorandom Generators (PRGs)

- Unpredictability vs Randomness & PRGs from Hard Functions

Constructing PRGs

- It seems to be very hard to construct PRGs unconditionally

Constructing PRGs

- It seems to be very hard to construct PRGs unconditionally
- As we will see soon, it turns out that one can use hard boolean functions to construct PRGs
- Idea:
  1. unpredictability equivalent to pseudorandomness (**[Yao 1982]**)
  2. a hard function should be hard to predict

Unpredictability vs Pseudorandomness

Lemma 4

If $f : \{0,1\}^\ell \to \{0,1\}$ *and there is a circuit* D *with* $S(D) \le s$ *s.t.*

$$|\Pr_x[D(x \circ f(x)) = 1] - \Pr_{x,b}[D(x \circ b) = 1]| > \varepsilon$$

then there is a circuit A *with* $S(A) \le s + 3$ *s.t.*

$$\Pr_x[A(x) = f(x)] > 1/2 + \varepsilon.$$

Unpredictability vs Pseudorandomness

Lemma 4

If $f : \{0,1\}^\ell \to \{0,1\}$ *and there is a circuit* D *with* $S(D) \le s$ *s.t.*

$$|\Pr_x[D(x \circ f(x)) = 1] - \Pr_{x,b}[D(x \circ b) = 1]| > \varepsilon$$

then there is a circuit A *with* $S(A) \le s + 3$ *s.t.*

$$\Pr_x[A(x) = f(x)] > 1/2 + \varepsilon.$$

- Above lemma shows that hard functions (on average), should "look random" to "efficient computation"

Unpredictability vs Pseudorandomness

Lemma 4

If $f : \{0,1\}^\ell \to \{0,1\}$ *and there is a circuit* D *with* $S(D) \leq s$ *s.t.*

$$|\Pr_x[D(x \circ f(x)) = 1] - \Pr_{x,b}[D(x \circ b) = 1]| > \varepsilon$$

then there is a circuit A *with* $S(A) \leq s + 3$ *s.t.*

$$\Pr_x[A(x) = f(x)] > 1/2 + \varepsilon.$$

- Above lemma shows that hard functions (on average), should "look random" to "efficient computation"
- Can assume there is circuit D' of size $\leq s + 1$ s.t.

$$\Pr_x[D'(x \circ f(x)) = 1] - \Pr_{x,b}[D'(x \circ b) = 1] > \varepsilon$$

  Since either D or $\neg D$ will do.

Proof of Lemma 4

Let's use D' as our circuit D

Main idea: guess random bit b and compute $D(x, b)$ to check whether b is a good guess for $f(x)$.

Let A_b be the procedure:

- Sample $b \sim \{0, 1\}$
- If $D(x, b) = 1$ then output b
- Else, output $1 - b$

Proof of Lemma 4

We will show from our assumption that

$$\Pr_{x,b}[A_b(x) = f(x)] > 1/2 + \varepsilon$$

$$\begin{aligned}\Pr_{x,b}[A_b(x) = f(x)] = &\Pr_{x,b}[A_b(x) = f(x) \mid b = f(x)] \cdot \Pr_{x,b}[b = f(x)] \\ &+ \Pr_{x,b}[A_b(x) = f(x) \mid b \neq f(x)] \cdot \Pr_{x,b}[b \neq f(x)]\end{aligned}$$

Proof of Lemma 4

We will show from our assumption that

$$\Pr_{x,b}[A_b(x) = f(x)] > 1/2 + \varepsilon$$

$$\begin{aligned}
\Pr_{x,b}[A_b(x) = f(x)] &= \Pr_{x,b}[A_b(x) = f(x) \mid b = f(x)] \cdot \Pr_{x,b}[b = f(x)] \\
&+ \Pr_{x,b}[A_b(x) = f(x) \mid b \neq f(x)] \cdot \Pr_{x,b}[b \neq f(x)] \\
&= \frac{1}{2} \cdot \Pr_{x,b}[A_b(x) = f(x) \mid b = f(x)] \\
&+ \frac{1}{2} \cdot \Pr_{x,b}[A_b(x) = f(x) \mid b \neq f(x)]
\end{aligned}$$

Proof of Lemma 4

We will show from our assumption that

$$\Pr_{x,b}[A_b(x) = f(x)] > 1/2 + \varepsilon$$

$$\begin{aligned}
\Pr_{x,b}[A_b(x) = f(x)] &= \Pr_{x,b}[A_b(x) = f(x) \;\mid\; b = f(x)] \cdot \Pr_{x,b}[b = f(x)] \\
&+ \Pr_{x,b}[A_b(x) = f(x) \;\mid\; b \neq f(x)] \cdot \Pr_{x,b}[b \neq f(x)] \\
&= \frac{1}{2} \cdot \Pr_{x,b}[A_b(x) = f(x) \;\mid\; b = f(x)] \\
&+ \frac{1}{2} \cdot \Pr_{x,b}[A_b(x) = f(x) \;\mid\; b \neq f(x)] \\
&= \frac{1}{2} \cdot \Pr_{x,b}[D(x \circ b) = 1 \;\mid\; b = f(x)] \\
&+ \frac{1}{2} \cdot \Pr_{x,b}[D(x \circ b) = 0 \;\mid\; b \neq f(x)]
\end{aligned}$$

Proof of Lemma 4

We will show from our assumption that

$$\Pr_{x,b}[A_b(x) = f(x)] > 1/2 + \varepsilon$$

$$\begin{aligned}
\Pr_{x,b}[A_b(x) = f(x)] &= \frac{1}{2} \cdot \Pr_{x,b}[D(x \circ b) = 1 \ | \ b = f(x)] \\
&+ \frac{1}{2} \cdot \Pr_{x,b}[D(x \circ b) = 0 \ | \ b \neq f(x)] \\
&= \frac{1}{2} + \frac{1}{2} \cdot \Pr_{x,b}[D(x \circ b) = 1 \ | \ b = f(x)] \\
&- \frac{1}{2} \cdot \Pr_{x,b}[D(x \circ b) = 1 \ | \ b \neq f(x)]
\end{aligned}$$

Proof of Lemma 4

We will show from our assumption that

$$\Pr_{x,b}[A_b(x) = f(x)] > 1/2 + \varepsilon$$

$$\begin{aligned}
\Pr_{x,b}[A_b(x) = f(x)] &= \frac{1}{2} + \frac{1}{2} \cdot \Pr_{x,b}[D(x \circ b) = 1 \mid b = f(x)] \\
&\quad - \frac{1}{2} \cdot \Pr_{x,b}[D(x \circ b) = 1 \mid b \neq f(x)] \\
&= \frac{1}{2} + \Pr_{x,b}[D(x \circ b) = 1 \mid b = f(x)] \\
&\quad - \frac{1}{2} \cdot \Pr_{x,b}[D(x \circ b) = 1 \mid b = f(x)] \\
&\quad - \frac{1}{2} \cdot \Pr_{x,b}[D(x \circ b) = 1 \mid b \neq f(x)]
\end{aligned}$$

Proof of Lemma 4

We will show from our assumption that

$$\Pr_{x,b}[A_b(x) = f(x)] > 1/2 + \varepsilon$$

$$\begin{aligned}
\Pr_{x,b}[A_b(x) = f(x)] &= \frac{1}{2} + \Pr_{x,b}[D(x \circ b) = 1 \ \mid \ b = f(x)] \\
&\quad - \frac{1}{2} \cdot \Pr_{x,b}[D(x \circ b) = 1 \ \mid \ b = f(x)] \\
&\quad - \frac{1}{2} \cdot \Pr_{x,b}[D(x \circ b) = 1 \ \mid \ b \neq f(x)] \\
&= \frac{1}{2} + \Pr_{x}[D(x \circ f(x)) = 1] - \Pr_{x,b}[D(x \circ b) = 1] \\
&> 1/2 + \varepsilon
\end{aligned}$$

Proof of Lemma 4

We will show from our assumption that

$$\Pr_{x,b}[A_b(x) = f(x)] > 1/2 + \varepsilon$$

- Thus, there is bit b^* such that

$$\Pr_{x}[A_{b^*}(x) = f(x)] > 1/2 + \varepsilon$$

Proof of Lemma 4

We will show from our assumption that

$$\Pr_{x,b}[A_b(x) = f(x)] > 1/2 + \varepsilon$$

- Thus, there is bit b^* such that

$$\Pr_{x}[A_{b^*}(x) = f(x)] > 1/2 + \varepsilon$$

- Circuit for A_{b^*}

$$A_{b^*}(x) = b^* \oplus (\neg D'(x, b^*))$$

Nisan-Wigderson PRG

Definition 5 (Average-Case Hardness)

Given $f : \{0,1\}^n \to \{0,1\}$, its average-case hardness, denoted by $H(f)$, is the smallest $s \in \mathbb{N}$ such that

$$\forall C \text{ circuit } \text{ s.t. } S(C) \leq s \Rightarrow \Pr_x[C(x) = f(x)] \leq 1/2 + 1/s$$

Nisan-Wigderson PRG

Definition 5 (Average-Case Hardness)

Given $f : \{0,1\}^n \to \{0,1\}$, its average-case hardness, denoted by $H(f)$, is the smallest $s \in \mathbb{N}$ such that

$$\forall C \text{ circuit s.t. } S(C) \leq s \Rightarrow \Pr_x[C(x) = f(x)] \leq 1/2 + 1/s$$

Theorem 6 (Special case of **[NW 1994]**)

If there is $L \in$ E and $\delta > 0$ such that for all sufficiently large n, $H(L_n) \geq 2^{\delta n}$, then there is constant $c > 0$ and family of PRGs $G_m : \{0,1\}^{c \log m} \to \{0,1\}^m$ which are computable in poly(m) *time and are $(2m, 1/8)$-pseudorandom.*

Nisan-Wigderson PRG

Definition 5 (Average-Case Hardness)

Given $f : \{0,1\}^n \to \{0,1\}$, its average-case hardness, denoted by $H(f)$, is the smallest $s \in \mathbb{N}$ such that

$$\forall C \text{ circuit } \text{ s.t. } S(C) \leq s \Rightarrow \Pr_x[C(x) = f(x)] \leq 1/2 + 1/s$$

Theorem 6 (Special case of **[NW 1994]**)

If there is $L \in$ E and $\delta > 0$ such that for all sufficiently large n, $H(L_n) \geq 2^{\delta n}$, then there is constant $c > 0$ and family of PRGs $G_m : \{0,1\}^{c \log m} \to \{0,1\}^m$ which are computable in poly(m) *time and are $(2m, 1/8)$-pseudorandom.*

- In particular, the above implies P $=$ BPP.

Constructing PRGs from hardness

One can actually obtain derandomization from worst-case hardness.

Theorem 7 ([IW 1997])

If there is $L \in$ E and $\delta > 0$ such that for all sufficiently large n, $S(L \cap \{0,1\}^n) \geq 2^{\delta n}$, then BPP = P.

Constructing PRGs from hardness

One can actually obtain derandomization from worst-case hardness.

Theorem 7 ([IW 1997])

If there is $L \in$ E and $\delta > 0$ such that for all sufficiently large n, $S(L \cap \{0,1\}^n) \geq 2^{\delta n}$, then BPP = P.

Theorem 8 ([IW 1998])

If BPP $\neq$ EXP, then for every $L \in$ BPP and $\varepsilon > 0$, there is a deterministic algorithm $A \in$ DTIME(2^{n^ε}) and, for infinitely many $n \in \mathbb{N}$ solves $L \cap \{0,1\}^n$ on a $1 - 1/n$ fraction of its inputs

Constructing PRGs from hardness

One can actually obtain derandomization from worst-case hardness.

Theorem 7 ([IW 1997])

If there is $L \in E$ and $\delta > 0$ such that for all sufficiently large n, $S(L \cap \{0,1\}^n) \geq 2^{\delta n}$, then BPP = P.

Theorem 8 ([IW 1998])

If BPP $\neq$ EXP, then for every $L \in$ BPP and $\varepsilon > 0$, there is a deterministic algorithm $A \in$ DTIME(2^{n^ε}) and, for infinitely many $n \in \mathbb{N}$ solves $L \cap \{0,1\}^n$ on a $1 - 1/n$ fraction of its inputs

- Assumptions in Theorem 7 stronger than in Theorem 8
  1. Non-uniform vs uniform
  2. exponential hardness vs super-polynomial hardness
- With stronger assumptions, (should) come stronger consequences
  1. Theorem 7 works over all inputs
  2. running time of simulations

Constructing PRGs from hardness

One can actually obtain derandomization from worst-case hardness.

Theorem 7 ([IW 1997])

If there is $L \in$ E and $\delta > 0$ such that for all sufficiently large n, $S(L \cap \{0,1\}^n) \geq 2^{\delta n}$, then BPP $=$ P.

Theorem 8 ([IW 1998])

If BPP $\neq$ EXP, then for every $L \in$ BPP and $\varepsilon > 0$, there is a deterministic algorithm $A \in$ DTIME(2^{n^ε}) and, for infinitely many $n \in \mathbb{N}$ solves $L \cap \{0,1\}^n$ on a $1 - 1/n$ fraction of its inputs

Turns out worst-case hypothesis $\Rightarrow$ average-case hypothesis

Theorem 9 ([BFNW 1993, I 1995, IW 1997])

If there is $L \in$ E and $\delta > 0$ s.t. for all sufficiently large n, $S(L_n) \geq 2^{\delta n}$, then there is $L' \in$ E and $\delta' > 0$ s.t. for sufiiciently large n, $H(L') \geq 2^{\delta' n}$.

References I

Arora, Sanjeev and Barak, Boaz (2009)
Computational Complexity, A Modern Approach — Chapters 9 & 20
Cambridge University Press

Papadimitriou, C (1994)
Computational Complexity
Addison-Wesley

Trevisan, Luca (2002)
Lecture notes — Lectures 23, 24
See webpage

Goldreich, Oded (2006)
Computational complexity: a conceptual perspective. — Chapter 6
https://www.wisdom.weizmann.ac.il/~oded/cc-drafts.html

References II

Babai, L and Fortnow, L and Nisan, N and Wigderson, A (1993)
BPP has subexponential time simulations unless EXPTIME has publishable proofs
Computational Complexity

Impagliazzo, Russell (1995)
Hard-core distributions for somewhat hard problems
FOCS

Impagliazzo, Russell and Wigderson, Avi (1997)
P = BPP unless E has subexponential circuits
STOC

Impagliazzo, Russell and Wigderson, Avi (1998)
Randomness vs Time: Derandomization under a uniform assumption
FOCS

References III

Nisan, Noam and Wigderson, Avi (1994)
Hardness vs Randomness
Journal of Computer and System Sciences

Yao, Andrew C. (1982)
Theory and applications of trapdoor functions
FOCS