Derandomization

Based on Arora and B. Barak (2006)

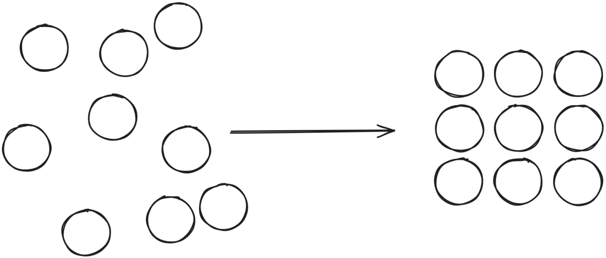
Benjamín Benčík

Seminar: Advanced Complexity Theory

March 20, 2025

Table of contents

What is derandomization?



What is derandomization?

Definition (Definition BPP Class)

Let $L \subseteq \{0,1\}^*$ be a language. We say that L is in **BPP** if there is a poly-time **PTM** M such that for all $x \in \{0,1\}^*$

$$x \in L \implies \Pr(M(x) = 1) \geq \frac{2}{3} \text{ and } x \notin L \implies P(M(x) = 0) \geq \frac{2}{3}$$

- Derandomization is process **BPP** $\rightarrow$ **P**
- It is not known **BPP** $\stackrel{?}{=}$ **P**
- At least **P** $\subset$ **BBP** holds trivially

Why do we care?

Derandomization gives insight into questions:

- What is the difference between problems that are randomized and deterministic?
- How can we use an imperfect source of randomness to achieve an almost perfect source?
- How much true randomness algorithm is neededs?

How to approach derandomization?

Common techniques of derandomization:

1. **Maximization of considional expectation** replace random choices with deterministic ones by iteratively fixing decisions to maximize expected value
2. **Use pseudorandom generators** replace perfect randomness with pseudorandom-randomness generated using a small seed
3. **Limited Independence** Instead of fully independent random variables use k-wise independent

In this lecture we will focus on pseudorandom generators.

Pseudo-random generators

Pseudorandom Distribution

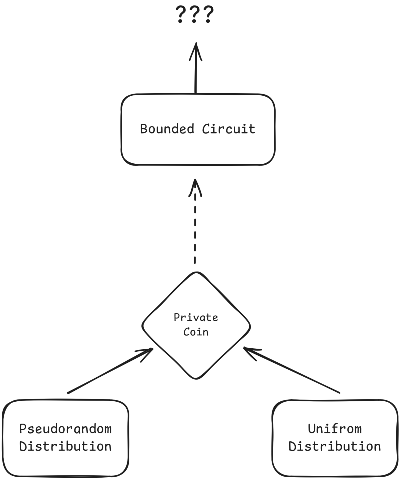
Definition (Pseudorandom Distribution (PD))

Let R be a distribution over $\{0,1\}^m, S \in \mathbb{N}, \epsilon > 0$. We say that R is (S, ϵ)-pseudorandom distribution if for every circuit C of size at most S:

$$|\Pr_{x \sim R}(C(x) = 1) - \Pr_{x \sim U_m}(C(x) = 1)| < \epsilon$$

Defined in terms of maximum advantage ϵ a circuit of size S can get when distinguishing R from uniform distribution.

Pseudorandom Distribution



Pseudorandom Generator

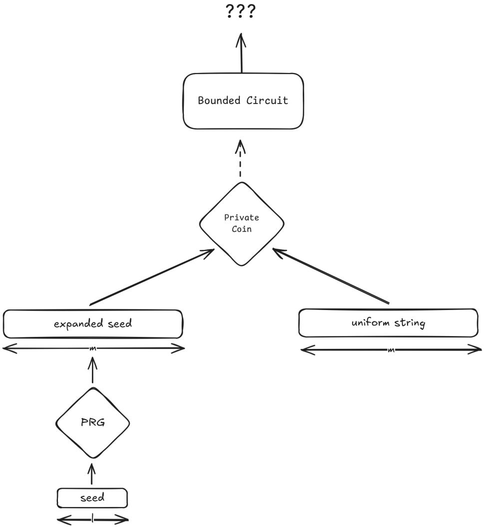
Definition (Pseudorandom generator PG)

If $S : \mathbb{N} \to \mathbb{N}$ is a poly-time computable monotone function then a function $G : \{0,1\}^* \to \{0,1\}^*$ with input z (seed) is called $S(\ell)$-pseudorandom generator if $\forall z \in \{0,1\}^\ell$

- $|G(z)| = S(l)$
- $G(z)$ can be computed in time $2^{c\ell}$ for a constant c
- $\forall \ell \in \mathbb{N} : G(U_\ell)$ is an $\left(S(l)^3, \frac{1}{10}\right)$-pseudorandom distribution

Maps a seed $z \in \{0,1\}^\ell$ to a longer output $G(z) \in \{0,1\}^m$ that is indistinguishable from uniform distribution by and small circuit C

Pseudorandom Generator



Pseudorandom Generator

- The definition allows the pseudorandom generator to run in exponential time with respect to the seed size.
- **Question:** *What can possible PRG distinguisher use to its advantage?*

Pseudorandom Generator

- The definition allows the pseudorandom generator to run in exponential time with respect to the seed size.
- **Question:** *What can possible PRG distinguisher use to its advantage?*
  - Are any bits biased towards certain values?
  - Is any pair of indices correlated?
  - How frequent are characters in a string?

Pseudorandom Generator

- The definition allows the pseudorandom generator to run in exponential time with respect to the seed size.
- **Question:** *What can possible PRG distinguisher use to its advantage?*
  - Are any bits biased towards certain values?
  - Is any pair of indices correlated?
  - How frequent are characters in a string?

Homework: *Consider* $h(z) = \left(\sum_{i=0}^{l} z_i\right) \mod 2$ *is* $G(z) = z \circ h(z)$ *valid* $(l+1)$*-pseudorandom generator? Give a proof for your answer.*

BPP vs P

Lemma (BPP vs P)

If there is a $2^{\epsilon l}$-pseudorandom generator for $\epsilon > 0$ then $\boldsymbol{BPP = P}$.

If $L \in$ **BPP** then by definition there exists an algorithm PTM $M(x, r)$ that uses random bits $r = \{0, 1\}^{poly(|x|)}$ and has correctness and soundness $2/3$.

Idea: instead of using truly random bits, use PRG over all possible seeds.

BPP vs P

Proof sketch:

- Let $G : \{0,1\}^\ell \to \{0,1\}^{S(\ell)}$ where $S(\ell) = 2^{\epsilon\ell}$ for $\epsilon > 0$
- PTM M can use at most $poly(|x|)$ randomness
- Use G to simulate randomness of PTM M on $l = \log(|x|)$
  1. For each seed $z \in \{0,1\}^\ell$ compute $G(s) \in \{0,1\}^{2^{\epsilon\ell}}$
  2. Run $M(x, G(z))$ and record the output
  3. Output the majority result over all runs
- The number of possible seeds is $2^l = 2^{\log(|x|)} = |x|$
- Argue that correctness and soundness is maintained

Computational Hardness

Hardness

Definition (Average-case Hardness)

The *average-case hardness* of f denoted $\mathrm{H}_{\mathrm{avg}}(f) : \mathbb{N} \to \mathbb{N}$ maps every $n \in \mathbb{N}$ to largest number S such that for every boolean circuit C on n inputs and size $\leq S$ holds

$$\Pr_{x \sim \{0,1\}^n}(C(x) = f(x)) \leq \frac{1}{2} + \frac{1}{|S|}$$

A function g is hard to compute if no "small" circuit can do much better at computing the function than guessing.

Claim: There exists a function that has an exponential average-case hardness.

Hardness

Definition (Worst-case Hardness)

Let $f : \{0,1\}^* \to \{0,1\}$, the *worst-case hardness* of f denoted $\mathrm{H}_{\mathrm{wrs}}(f) : \mathbb{N} \to \mathbb{N}$ maps $n \in \mathbb{N}$ to the largest number S such that every boolean circuit of size $\leq S$ fails to compute f on input $\{0,1\}^n$.

Worst-case hardness is weaker than average-case hardness.

Hardness

Definition (Worst-case Hardness)

Let $f : \{0,1\}^* \to \{0,1\}$, the *worst-case hardness* of f denoted $\mathrm{H}_{\mathsf{wrs}}(f) : \mathbb{N} \to \mathbb{N}$ maps $n \in \mathbb{N}$ to the largest number S such that every boolean circuit of size $\leq S$ fails to compute f on input $\{0,1\}^n$.

Worst-case hardness is weaker than average-case hardness.

Homework: *Show the construction of a function that has exponential worst-case hardness.*

Pseudorandomness $\to$ Hardness

Lemma (Pseudorandomness implies harndess)

Let $G : \{0,1\}^\ell \to \{0,1\}^{\ell+1}$ be a $S(\ell)$-pseudorandom generator. Let $T = \{G(z) : z \in \{0,1\}^\ell)$ and define $f : \{0,1\}^{n+1} \to \{0,1\}$ as $f(x) = 1$ if $x \in T$ and 0 otherwise. The function f is $S(\ell)$-hard

To show validity of the construction assume that f is not hard and arrive to contradiction that $S(l)$ is not pseudorandom.

Proof:

- If f is not hard there exists C computing it
- Number of possible seeds is at most $2^\ell \implies |T| \leq 2^n$
- C can accept in at most $1/2$ cases

$$\Pr_{x \sim U_{n+1}}(C(x) = 1) \leq \frac{2^n}{2^{n+1}} = \frac{1}{2}$$

Pseudorandomness $\to$ Hardness

- Trivially $\Pr_{x \sim U_{n+1}}(C(G(x)) = 1) = 1$
- The distinguishing advantage is too high

$$\left| \Pr_{x \sim Un}[C(G(x)) = 1] - \Pr_{x \sim U_{n+1}}[C(x) = 1] \right| \geq 1 - \frac{1}{2} = \frac{1}{2}$$

- There is a small C that can distinguish distribution generated by G from U_{n+1}, therefore G cannot be a PRG

$\square$

Hardness $\rightarrow$ pseudorandomness - Yao's Theorem

Yao's Theorem (16.14, AB)

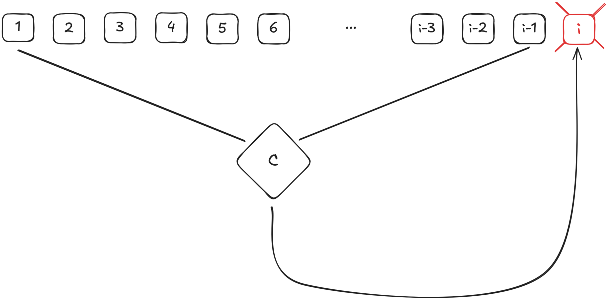
Let Y be a dirtribution over $\{0,1\}^m$. Supose there exists $S \geq 10n, \epsilon > 0$ such that for every circuit C of size $\leq 2S$ and $i \in [m]$

$$\Pr_{z \sim Y}(C(z_1, z_2, \ldots, z_{i-1}) = z_i) \leq \frac{1}{2} + \frac{\epsilon}{m}$$

The Y is (S, ϵ)-pseudorandom distribution.

If no small circuit can predict the next output of distribution Y, then the distribution is pseudorandom.

Hardness → pseudorandomness - Yao's Theorem



Hardness $\rightarrow$ pseudorandomness - Yao's Theorem

Proof of Yao's theorem:
Suppose the Y is is not (S, ϵ)-pseudorandom, then there exists D

$$|D| < S \;\wedge\; |\Pr_{r\sim Y}(D(r) = 1) - \Pr_{r\sim U}(D(r) = 1)| > \epsilon \tag{1}$$

Construct m distributions $H_1, H_2, \ldots H_m$:

$$H_i = \begin{cases} r \sim U_m & i = 1 \\ r \sim Y & i = m \\ (r1, \ldots, r_i) \sim Y \circ (r_{i+1}, \ldots, r_m) \sim U_{m-i} & 0 < i < m \end{cases}$$

Hardness $\rightarrow$ pseudorandomness - Yao's Theorem

Split the total distinguishing advantage into sum of distinguishing advantages for each position:

$$|\Pr_{r\sim Y}(D(r){=}1){-}\Pr_{r\sim U}(D(r){=}1)|{=}\sum_{i=1}^{m}|\Pr_{r\sim Y}(D(H_i){=}1){-}\Pr_{r\sim U}(D(H_{i-1}){=}1)|$$

Since the overall difference is at least ϵ, there must be i such that

$$|\Pr_{r\sim Y}(D(H_i)=1)-\Pr_{r\sim U}(D(H_{i-1})=1)|\geq\frac{\epsilon}{m}$$

Distinguisher C can be constructed by combining $D(H_i), D(H_{i-1})$ and resulting circuit has size $\leq 2S$ which implies contradiction

$$\Pr_{z\sim Y}(C(z_1,z_2,\ldots,z_{i-1})=z_i)\geq\frac{1}{2}+\frac{\epsilon}{m}$$

$\square$

Hardness $\rightarrow$ pseudorandomness

Lemma (Pseudorandomness from hardness (16.13, AB))

Suppose there is $f \in \mathbf{E}$ with $\mathrm{H}_{\mathrm{avg}}(f) \geq n^4$. Then there is a $S(l)$-pseudorandom generator for which $S(l) = l + 1$

The construction of generator is $\forall z \in \{0,1\}^l : G(z) = z \circ f(z)$

We just need to show this it is a valid $((l+1)^3, 1/10)$ pseudorandom generator by using Yao's theorem.

Hardness $\to$ pseudorandomness

Proof:
By Yao's theorem It is enough to show that there is no circuit C of size $\leq 2(l+1)^3$ and $i \in [l+1]$ that could predict:

$$\Pr_{r \sim G(U_l)}(C(r_1, r_2, \ldots, r_{i-1}) = r_i) > \frac{1}{2} + \frac{\epsilon}{2(l+1)}$$

- For $i < l$, the i-th bit of $G(z)$ is completely random by construction thus, no circuit, regardless of size, can predict it
- For $i = l + 1$, boils down to computing f but the function is n^4 hard so it cannot be computed by circuit of size $2(l+1)^3$

□

Nisan-Wigderson Construction

Approaches to extending

- We showed expansion one bit
- **Question:** *How about larger expansion?*

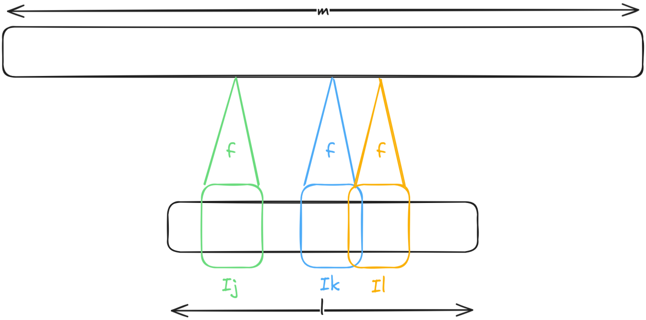
Approaches to extending

- We showed expansion one bit
- **Question:** *How about larger expansion?*

$$G_2(z) = z_1 \dots z_{l/2} \circ f(z_1 \dots z_{l/2}) \circ z_{l/2+1} \dots z_l \circ f(z_{l/2+1} \dots z_l)$$
$$G_3(z) = z_1 \dots z_{l/3} \circ f(z_1 \dots z_{l/3}) \circ \dots \circ f(\circ z_{2l/3+1} \dots z_l)$$

- By above, we do not get past linear expansion
- ...yes, we can do better, that is why we are here

Larger expansions



Pick random subsets of indices, pass them through the hard function f, and concatenate the results.

Construction

Definition (Nisan-Widgerson Generator)

If $\mathcal{I} = \{I_1, \ldots, I_m\}$ is a family of subsets of $[\ell]$ where each $|I_j| = l$ and $f : \{0,1\}^n \to \{0,1\}$ is any function then the $(\mathcal{I}, f)$-NW generator is the function $\mathrm{NW}^f_{\mathcal{I}} : \{0,1\}^l \to \{0,1\}^m$:

$$\forall Z \in \{0,1\}^l : \mathrm{NW}^f_{\mathcal{I}}(Z) = f(Z_{I_1}) \circ f(Z_{I_2}) \circ \ldots \circ f(Z_{I_m})$$

where $Z_{I_j} = \{z_k : k \in I_j\}$

Vaguely we require that family of subsets is constructed in a *"reasonable way"* and f is *"sufficiently hard"*.

How to construct subsets?

Definition (Combinatorial design)

If $d, n, \ell \in \mathbb{N}$ are numbers with $\ell > n > d$ then a family $\mathcal{I} = \{I_1, \ldots, I_m\}$ of subsets of $[\ell]$ is an (ℓ, n, d)-design if $\forall j \in [m] : |I_j| = n$ and $|I_j \cap I_k| \leq d$ for every $j \neq k$.

The above definition guarantees that the inputs to f

- have a constant size
- are pairwise dependent on at most d bits

Algorithm 1 Subset Construction

Input: ℓ: seed size, d: max intersection, n: subset size
Output: (ℓ, n, d)-design with $2^{d/10}$ sets

$\mathcal{I} \leftarrow \emptyset$
for each n-sized set $I \in [\ell]$ **do**
    if $|\mathcal{I}| = 2^{d/10}$ **then**
        return $\mathcal{I}$
    end if
    for each $j \in [m]$ **do**
        if $|I \cap I_j| \leq d$ **then**
            $\mathcal{I} \leftarrow \mathcal{I} \cup \{I_j\}$
            break
        end if
    end for
end for

Construction of combinatorial design

Lemma (Construction of combinatorial design (16.18 AB))

On input ℓ, d, n with $\ell > 10n^2/d$ algorithm for Algorithm 1 will construct a (ℓ, d, n)-design with $2^{d/10}$ sets.

- Running time: $\text{poly}(m)2^{\ell} \to 2^{\mathcal{O}(\ell)}$
- Need to prove the greedy algorithm does not *"get stuck"*
- More formally for $m < 2^{d/10}$ there always exists n-sized $I \subseteq [\ell]$ which could be added into $\mathcal{I}$
- We show that by picking elements from $[\ell]$ into $\mathcal{I}$ with uniform probability above condition is always satisfied

Construction of combinatorial design - Sufficient size of I

Proof:

- Add $x \in [l]$ to I with probability $2n/\ell$
- Now we calculate $\Pr(|I| \geq n)$
- Model $|I| \sim Bin(l, 2n/\ell) \implies \mathbb{E}[|I|] = 2n$
- By Chernoff for $\delta = 1/2$

$$\begin{aligned}
\Pr(|I| > n) &= 1 - \Pr(|I| \leq n) \\
&= 1 - \Pr(|I| \leq (1 - 1/2)\mathbb{E}[|I|]) \\
&\geq 1 - \exp(-(\delta^2 \mathbb{E}[|I|])/2) \\
&= 1 - \exp(-n/4)
\end{aligned}$$

- When I is larger than n we truncate it without damaging properties of the design

Construction of combinatorial design - Sufficient Independance

- Each I_j picks elements uniformly at random,
- Therefore $\Pr(x \in I_j) = n/\ell$
- Model $\forall j \in m : |I \cap I_j| \sim Bin(\ell, n/\ell) \implies \mathbb{E}[|I \cap I_j|] = n$
- By Chernoff for all $j \in [m]$ and $\delta = d/n - 1$

$$
\begin{aligned}
\Pr(|I \cap I_j| \geq d) &= \Pr(|I \cap I_j| \geq (1-\delta)\mathbb{E}[|I \cap I_j|]) \\
&\leq \exp(-(\delta^2 n)/3) = \exp\left(-\frac{(d/n-1)^2 n}{3}\right) \\
&\leq \exp\left(-\frac{(d/n)^2 n}{3}\right) \\
&= \exp\left(-\frac{d^2}{n3}\right)
\end{aligned}
$$

Construction of combinatorial design - Putting it together

- Proof sets $S(n) < 2^n$ and $d = \log(S(n)/10)$
- From above $d \leq n/10$
- From statement we have $m < 2^{d/10}$
- In Algorithm 1 there is not a suitable set $\mathcal{I}$

$$
\begin{aligned}
&\Pr(\nexists \text{ suitable subset }) \\
&= \Pr(|I| > n) \cdot \Pr(\forall j\ [l] : |I \cap I_j| < d) \\
&= \Pr(|I| > n) \cdot (1 - \Pr(\forall j\ [l] : |I \cap I_j| \geq d))^{|\mathcal{I}|} \\
&\geq (1 - \exp(-n/4)) \cdot (1 - (\exp(-d^2/n^3))^m \\
&\geq (1 - \exp(-n/4)) \cdot (1 - (\exp(-n/100))^{2^{n/100}}
\end{aligned}
$$

Construction of combinatorial design - Putting it together

According to AB *"together these two conditions imply that with probability at least 0.4, the set I will simultaneously conditions"*. Considering this probability is good enough, we conclude proof for the existence of combinatorial design □

Pseudorandomness using the NW generator

We use the combinatorial design to show the central theorem of this lecture

Theorem (Pseudorandomness from NW generator (16.19 AB))

If $\mathcal{I}$ is an (l, n, d)-design with $|\mathcal{I}| = 2^{d/10}$ and $f : \{0,1\}^n \to \{0,1\}$ satisfying $\mathsf{H}_{\mathsf{avg}}(f)(n) > 2^{d^2}$, then the distribution $\mathsf{NW}_{\mathcal{I}}^{f}(U_l)$ is a $(\mathsf{H}_{\mathsf{avg}}(f)(n)/10, 1/10)$-pseudorandom distribution.

Unformally: f is a hard function and $\mathcal{I}$ is a design with sufficiently large parameters, then $\mathsf{NW}_{\mathcal{I}}^{f}(U_l)$ is a pseudorandom distribution.

NW Proof Outline

1. Idea similar as proof for $(l+1)$-pseudorandom generator
2. We want to prove that that for $i \in [2^{10/d}]$ there does not exist $S/2$ sized circuit guessing the next bit (Yao's theorem)
3. Assume there exists such a circuit
4. Manipulate the expression and apply averaging principle
5. Arrive to contradiction about hardness of f

NW is PRG - Using Yao's Theorem

For contradiction suppose there exists a circuit C and $i \in [2^{d/10}]$ deciding random bit R_i from distribution $R_1, \dots R_{i-1}$:

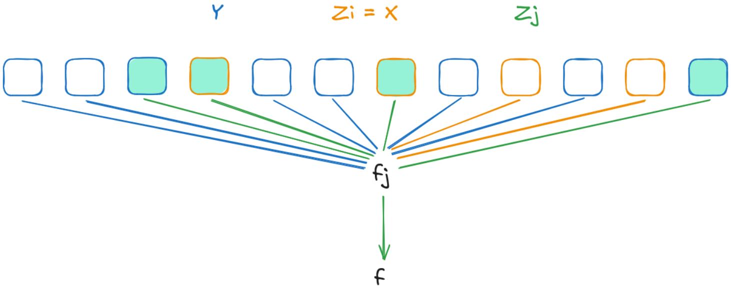
$$\Pr_{\substack{Z \sim U_\ell \\ R = \mathrm{NW}^{\mathrm{f}}_{\mathcal{I}}(Z)}} \left(C(R_1, R_2, \dots, R_{i-1}) = R_i \right) \geq \frac{1}{2} + \frac{1}{10 \cdot 2^{d/10}} \tag{2}$$

Assuming Equation (2) holds we use the definition of $\mathrm{NW}^{\mathrm{f}}_{\mathcal{I}}(Z)$ where Z_j with Z being seeds to hard function f:

$$\Pr_{Z \sim U_\ell} \left(C(f(Z_{I_1}), \dots, f(Z_{I_{i-1}})) = f(Z_{I_i}) \right) \geq \frac{1}{2} + \frac{1}{10 \cdot 2^{d/10}} \tag{3}$$

NW is PRG - Splitting the Seed

- $X = Z_{I_i}$
- $Y = Z_{[\ell]\setminus I_i}$
- Bits of Z are independent X, Y are independent r.v.
- f_j takes the role of combinatorial design



NW is PRG - Splitting seed

Based on the new notation rewrite Equation (3)

$$\Pr_{\substack{X\sim U_\ell \\ Y\sim U_{\ell-n}}} (C(f_1(X,Y),\dots,f_{i-1}(X,Y) = f(X)) \geq \frac{1}{2} + \frac{1}{10\cdot 2^{d/10}} \tag{4}$$

$j < i : f_j(X,Y) = f(Z_{I_j})$ picks parts of X, Y that are relevant to I_j
Observe Y is dependant only on I_i thus can be fixed to some string $y \in \{0,1\}^{\ell-n}$

$$\Pr_{X\sim U_n}(C(f_1(X,y),\dots,f_{i-1}(X,y)) \geq \frac{1}{2} + \frac{1}{10\cdot 2^{d/10}} \tag{5}$$

NW is PRG - Averaging principle

Lemma (Averaging principle)

Have event E depend on two uniform independent random variables $A \in U_e, B \in U_f$:

$$\exists b \in B : \Pr_A(E(A, b)) \geq \Pr_{A,B}(E(A, B))$$

$$\Pr_{A,B}(E(A, B)) = \sum_{b \in B} \Pr_B(B = b) \Pr_A(E(A, b)) \tag{6}$$

$$\Pr_{A,B}(E(A, B)) = \frac{1}{|B|} \sum_{b \in B} \Pr_A(E(A, b)) \text{ by } B \in U_f \tag{7}$$

NW is PRG - Averaging principle

Joint probability is the average of $\Pr_A(E(A,b))$. For contradiction:

$$\forall b \in B : \Pr_A(E(A,b)) < \Pr_{A,B}(E(A,B))$$

$\Pr_{A,B}(E(A,B))$ cannot be average $\square$

Applying the lemma to Equation (3) there exists $y \in \{0,1\}^{n-\ell}$:

$$\Pr_{X \sim U_n}(C(f_1(X,y),\ldots,f_{i-1}(X,y) = f(X)) \geq \frac{1}{2} + \frac{1}{10 \cdot 2^{d/10}} \tag{8}$$

NW is PRG - Constructing a circuit

Since $i \neq j : |I_i \cap I_j| \leq d$ $f_j(X, y)$ depends on at most d coordinates
Construct a circuit D:

1. Take $X \sim U_n$ and hard-wire $y \in \{0,1\}^{n-\ell}$
2. For each $j < i$ compute $f_j(X, y)$ using small circuit of size $d2^d$
3. Feed the results in to C to obtain $f(X) = f(Z_{I_i})$

D has size $i \cdot$ |small circuit| $+ |C|$

- $i \leq 2^{d/10}$
- |small circuit| $\leq d2^d$
- $|C| = S/2$ by Yao

NW is PRG - Constructing a circuit

Given $d = \log(S(n))/10$

$$|D| \leq 2^{d/10} \cdot d2^d + S/2 = d2^{d \cdot \frac{11}{10}} + S/2 \tag{9}$$

$$= \frac{\log(S(n))}{10} S(n)^{\frac{11}{100}} + S/2 \leq S \tag{10}$$

$$\Pr_{X \sim U_n}(D(X) = f(X)) \geq \frac{1}{2} + \frac{1}{10 \cdot 2^{d/10}} \geq \frac{1}{2} + \frac{1}{S} \tag{11}$$

Since there exists a small circuit, this breaks the hardness of f, and we get contradiction. $\square$

Consequences of NW Generator

By setting parameters $l > \frac{100n^2}{\log(S(n))}$, $d = \frac{\log(S(n))}{10}$, we can show:

Theorem (Consequences of NW Generator (16.10 AB))

Given $f \in \boldsymbol{E}$ and every polynomial-time computable monotone $S : \mathbb{N} \to \mathbb{N}$ with $\mathrm{H}_{\mathrm{avg}}(f) \geq S$ we can construct $S(l)$-pseudorandom generator where $S(\epsilon\sqrt{l}\log(S(\epsilon\sqrt{l})))^{\epsilon}$ for $\epsilon > 0$.

Consequences of NW Generator

By setting parameters $l > \frac{100n^2}{\log(S(n))}$, $d = \frac{\log(S(n))}{10}$, we can show:

Theorem (Consequences of NW Generator (16.10 AB))

Given $f \in \boldsymbol{E}$ and every polynomial-time computable monotone $S : \mathbb{N} \to \mathbb{N}$ with $\mathrm{H}_{\mathrm{avg}}(f) \geq S$ we can construct $S(l)$-pseudorandom generator where $S(\epsilon\sqrt{l}\log(S(\epsilon\sqrt{l})))^{\epsilon}$ for $\epsilon > 0$.

Homework: *If there exists $f \in \mathbf{E} = \mathbf{DTIME}(2^{\mathcal{O}(n)})$ and $\epsilon > 0$ such that $\mathrm{H}_{\mathrm{avg}}(f) \geq 2^{\epsilon n}$ then $\mathbf{BPP} = \mathbf{P}$.*

Extractors

Motivation

- For this part suppose that we are happy with probablistic algorithms and feel now need to derandomize them
- Sources of randomness rarely behave as perfectly uncorrelated and unbiased coin tosses
- Application of extractors
  - Running randomized algorithms using weak random sources
  - Recycling random bits

Definition

Definition (Minimum Entropy)

Minimum entropy: of X denoted as $H_\infty(X)$ is

$$\mathrm{argmax}_{k \in \mathbb{R}} \{\Pr(X = x) \leq 2^{-k}\}$$

- $H_\infty(x) \leq n$
- $H_\infty(x) = n$ iff X is U_n
- Our goal will be to execute probabilistic algorithms on sources of randomness with as small $H_\infty(X)$ as possible
- If $H_\infty(x) \geq k$ then it is (n, k)-source

Definition

Definition (Statistical distance)

Fro two variables X and Y in $\{0,1\}^n$ their statistical distance is defined as $\delta(X,Y) = \max_{S \subseteq \{0,1\}^n}\{\Pr(X \in S) - \Pr(Y \in S)\}$

- Statistical distance quantifies the maximum difference in probabilities that X and Y assign to any subset of S
- Small statistical distance implies that the distributions are statistically indistinguishable

Definition

Definition (Extractor)

A function $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ is (k, ϵ) extractor then for all $X \in \{0,1\}^n$ with minimal entropy k

$$\forall S \subseteq \{0,1\}^m : \left| \Pr_{a \in X, z \in \{0,1\}^t}(\mathrm{Ext}(a,z) \in S) - \Pr_{r \in \{0,1\}^m}(r \in S) \right| \leq \epsilon$$

- Extractor is given weak random source X and small seed of size t and outputs string of m bits that is close to uniform
- The extractor "purifies" the weak randomness of X using a small amount of true randomness
- Intuitively, you cannot extract more randomness than what is present in the source.

Deterministic extractors do not work

Lemma

For every $\mathrm{Ext} : \{0,1\}^n \to \{0,1\}^m$ *and every* $k \leq n-1$ *there is a* (n,k)*-source such that for every* $x \in X : \mathrm{Ext}(x)$ *the first bit is constant.*

Proof:

- Fix a deterministic extractor Ext
- Denote $S_0 = \{x \in \{0,1\}^n : \mathrm{Ext}(x) \text{ has the first bit } 0\}$ and analogously S_1
- Observe either $|S_0| \geq 2^{n-1}$ or $|S_1| \geq 2^{n-1}$
- Assume $|S_0| \geq 2^{n-1}$ and construct $X \subseteq S_0$

Deterministic extractors do not work

- Let $X = S_0$
- Evaluate $H_\infty(X)$

$$
\begin{aligned}
\Pr(X = x) &\leq 2^{-k} \\
\frac{1}{|S_0|} &\leq 2^{-k} \\
-\log(|S_0|) &\leq -k \\
k &\leq n - 1
\end{aligned}
$$

$\square$

Corollary: A deterministic extractor is at least $1/2$ statistical distance from U_m; thus, it needs additional randomness.

Wrap-up and Applications

We have seen...

- What is derandomization.
- Formalization of derandomization through PRG.
- Relations between hardness and randomness.
- Combinatorial design for selecting subsets.
- Nisan-Wigderson: a PRG with exponential extension.
- Extractors and relation to derandomization.

Applications

- Relation of classes BPP and P
- Space-bound computation: is randomness necessary for space-efficient computation? Hoza (2022)
- Extractors to "purify" weak sources in cryptography
- Construction of commitment schemes in cryptography using NW Boaz Barak, Ong, and Vadhan (2005)
- New lower bounds in circuit classes

References

Arora, S. and B. Barak (2006). *Computational Complexity: A Modern Approach*. Cambridge University Press. ISBN: 978-0-521-42426-4. URL: https://theory.cs.princeton.edu/complexity/book.pdf.

Barak, Boaz, Shien Jin Ong, and Salil Vadhan (2005). *Derandomization in Cryptography*. Cryptology ePrint Archive, Paper 2005/365. URL: https://eprint.iacr.org/2005/365.

Hoza, William M. (2022). "Recent Progress on Derandomizing Space-Bounded Computation". In: *Bull. EATCS* 138. URL: http://eatcs.org/beatcs/index.php/beatcs/article/view/728.