Lecture 7: Pseudo-Random Generators against BPP, and Interactive Proofs for the Permanent

Topics in Pseudorandomness and Complexity Theory (Spring 2017)
Rutgers University
Swastik Kopparty
Scribe: Amnon Attali

1 Pseudo-Random Generator

Definition 1. *A Pseudo-Random Generator (PRG) against size s circuits is an algorithm to which we feed l bits to produce m bits that can fool a size s circuit. So* $G : \{0,1\}^l \to \{0,1\}^m$ *such that* $\forall$ *Circuits C of size* $\leq s$

$$|\Pr_{x\in\{0,1\}^l}[C(G(x)) = 1] - \Pr_{y\in\{0,1\}^m}[C(y) = 1]| < \epsilon$$

Remark. *Why is such an algorithm G useful?*
Because if we have a uniform algorithm PRG $G : \{0,1\}^{\mathcal{O}(\log m)} \to \{0,1\}^{\mathcal{O}(m)}$ *against circuits of size* $\leq \mathcal{O}(m)$*, with error* $\epsilon < 0.1$ *then P=BPP.*
To see why: say we are given an algorithm A(x,r) that decides some fixed language $L \in BPP$*, input* $|x| = n$*, randomness* $|r| \leq n^c$ *that runs in* $\mathcal{O}(n^c)$*. Now consider a circuit* $C : \{0,1\}^{|r|} \to \{0,1\}$ *on some fixed* $z \in L$*. So C(r) = A(z,r), and C has size* $\leq \mathcal{O}(n^c)$*. Now using G with* $m = n^c, \mathcal{O}(\log m) = \mathcal{O}(\log n)$ *we know that*
$|\Pr_{x\in\{0,1\}^{\log m}}[C(G(x)) = 1] - \Pr_{y\in\{0,1\}^m}[C(y) = 1]| < 0.1$
But since $L \in BPP$ *the* $\Pr[C(y) = 1] > \frac{2}{3}$ *if* $z \in L$ *or* $\Pr[C(y) = 1] < \frac{1}{3}$ *if* $z \notin L$
So finally $\Pr[C(G(x)) = 1] > 0.56$ *and we can decide whether* $z \in L$ *(by taking majority as usual) in time* $\mathcal{O}(m)$*.*
In other words, the PRG allows us to ensure that the majority output of C(r) for a fixed z is correct.

Now we wish to find such a PRG.
Recall that last time we showed how to produce $l + 1$ random bits from a seed of length l. More precisely, given $f : \{0,1\}^l \to \{0,1\}$ we define $G : \{0,1\}^l \to \{0,1\}^{l+1}$ as $G(x) = (x, f(x))$. We showed that if f is $\frac{1}{2} - \epsilon$ hard for circuits of size $\leq$ s, then G is an ϵ-PRG against circuits of size $\leq as$ (for some small constant a).
Notice that this method can be used to make even more random bits: the first $l + 1$ bits are as before, the $l + 2^{\text{nd}}$ is $f(x_2, x_3, ..., f(x))$, and so on recursively. BUT the size of the circuit against which this PRG works decays exponentially...

Theorem 2 (Nisam-Widgerson PRG)**.** *Given* $s_1, s_2, ..., s_m \subseteq [l]$*, each* $|s_i| = a$*, and* $f : \{0,1\}^a \to \{0,1\}$ *that is* $\frac{1}{2} - \epsilon$ *hard for circuits of size* $\leq s$*, then for* $x \in \{0,1\}^l$*,*
$G(x) = (f(x|_{s_1}), f(x|_{s_2}), ..., f(x|_{s_m}))$ *is a* $\mathcal{O}(m\epsilon)$*-PRG against circuits of size* $\tilde{s} = \frac{1}{10}(s - 2^b m)$ *for* $b = \frac{2a^2}{l}$*.*

Proof. We will prove the theorem by contrapositive.
Suppose $\exists\, \tilde{c}$ of size $\tilde{s}$ that distinguishes the output of G, ie.

$$|\Pr_{x\in\{0,1\}^l}[\tilde{C}(G(x))=1]-\Pr_{y\in\{0,1\}^m}[\tilde{C}(y)=1]|>m\epsilon$$

Now consider the hybrid distribution z.
$z_i = G(x)_1, G(x)_2, ..., G(x)_i, y_{m-i}, ..., y_m$ Note that $z_m = G(x)$ and $z_0 = y$.
So

$$|\Pr[\tilde{C}(G(x))=1]-\Pr[\tilde{C}(y)=1]|>m\epsilon \implies |\Pr[\tilde{C}(z_m)=1]-\Pr[\tilde{C}(z_0)=1]|>m\epsilon$$

$$\implies |\sum_{i=0}^{m-1}(\Pr[\tilde{C}(z_{i+1})=1]-\Pr[\tilde{C}(z_i)=1])|>m\epsilon\ (telescoping\,sum)$$

$$\implies \exists\, i\, such\, that\, |\Pr[\tilde{C}(z_{i+1})=1]-\Pr[\tilde{C}(z_i)=1]|>\epsilon\ (by\, pigeonhole)$$

$$\implies |\Pr_{x,y}[\tilde{C}(f(x|_{s_1}),...,f(x|_{s_{i+1}}),y_{i+2},...,y_m)=1]-$$

$$\Pr_{x,y}[\tilde{C}(f(x|_{s_1}),...,f(x|_{s_i}),y_{i+1},...,y_m)=1]|>\epsilon$$

$$\implies |\mathop{\mathbf{E}}_{x,y_{i+1},...,y_m}[\tilde{C}(...f(x|_{s_{i+1}}),y_{i+2}...)-\tilde{C}(...f(x|_{s_i}),y_{i+1}...)]|>\epsilon$$

So there is a fixed $y* = y_{i+2}, ..., y_m$ such that

$$|\mathop{\mathbf{E}}_{x,y_{i+1}}[\tilde{C}(...f(x|_{s_{i+1}}),y*)-\tilde{C}(...f(x|_{s_i}),y_{i+1},y*)]|>\epsilon$$

So $\tilde{C}$ is distinguishing the $i+1^{\text{th}}$ bit.
Our plan now is that given $z\in\{0,1\}^a$ to find $f(z)$ with probability $\geq \frac{1}{2}+\epsilon$ showing that f is not $\frac{1}{2}-\epsilon$ hard.
To do this we will create an $x\in\{0,1\}^l$ such that $x|_{s_{i+1}} = z$
Let $u = x|_{s_{i+1}}\, and\, v = x|_{s_{i+1}{}^c}$ so that $x = (u, v)$ and $u_j = u|_{s_j} = x|_{s_j\cap s_{i+1}}\, and\, v_j = v|_{s_j} = x|_{s_j\cap s_{i+1}{}^c}$.
Note that $f(u_k, v_k) = f(x|s_k)$, and notably $f(u_{i+1}, v_{i+1}) = f(u) = f(z)$ So:

$$|\mathop{\mathbf{E}}_{u,v,y_{i+1}}[\tilde{C}(f(u_1,v_1),...,f(u_{i+1},v_{i+1}),y*)-\tilde{C}(f(u_1,v_1),...,f(u_i,v_i),y_{i+1},y*)]|>\epsilon$$

So there is a fixed v* such that this holds. Let C' be the circuit such that
$C'(u,w) = \tilde{C}(f(u_1,v*_1),...,f(u_i,v*_i),w,y*)\, for\, u\in\{0,1\}^a,\ w\in\{0,1\}$

$$\implies |\mathop{\mathbf{E}}_{u,y_{i+1}}[C'(u,f(u))-C'(u,y_{i+1})]|>\epsilon$$

Size of C' $\leq \mathcal{O}(size(\tilde{C}+m2^b)) = \mathcal{O}(\tilde{s}+m2^b)$
This means that we can distinguish (u,f(u)) from (u,random bit) so as we showed last class this means that $\exists$ c of size $\mathcal{O}(\tilde{s}+m2^b)$ such that

$$\Pr_u[C(u)=f(u)]>\frac{1}{2}+\epsilon$$

□

We now move on to our major theorem...

Theorem 3 (Impagliazzo-Wigderson)**.** *If $\exists$ some function $f \in TIME(2^{\mathcal{O}(n)})$ such that circuits of size $2^{\delta n}$ cannot compute f, then P=BPP.*

Proof. 1. Hardness amplification:
$\exists f \in TIME(2^{\mathcal{O}(n)})$ hard for size $2^{\delta n} \implies \exists f \in TIME(2^{\mathcal{O}(n)})$ that is $(\frac{1}{2} - 2^{\epsilon n})$ hard for size $2^{\epsilon n}$ circuits.
2. Use this f in Nisan-Wigderson with $l = \frac{1}{\epsilon^2}n$, $a = n$, $b = \frac{\epsilon}{4}n$, $m = 2^{\frac{\epsilon}{4}n}$
So we get a $2^{\mathcal{O}(\epsilon n)}$ - PRG against size $2^{\mathcal{O}(\epsilon n)}$ circuits. So our PRG G takes in $\mathcal{O}(\log(m)) = \mathcal{O}(n)$ bits to m bits computable in poly(n) time. □

2 Interactive Proofs

Definition 4 (Permanent)**.**

$$Perm : \mathbb{F}^{n\times n} \to \mathbb{F}, Perm(m) = \sum_{\substack{\sigma_i[n]\to[n] \\ permutations}} \prod_{i=1}^{n} m_{i\sigma(i)}$$

(don't have efficient computation of $Perm(m)$)

Theorem 5 (Valiant)**.** *The Permanent is #P-hard*

Definition 6 (NP proof system for L)**.** *is an algorithm $V(X, \Pi), |\Pi| \leq |X|^C and$:*
If $X \in L, \exists\, \Pi$ such that $V(X, \Pi) \to ACCEPTS$
If $X \notin L, \forall\, \Pi$ such that $V(X, \Pi) \to REJECTS$

Definition 7 (Interactive proof for L)**.** *is a randomized algorithm V(X), which runs in time $|X|^C$ such that:*

$$\text{if } X \in L, \exists \text{ prover } P, \Pr_{rand.\ of\ V}(V(X) \text{ interacting with } PACCEPTS) \geq 0.99$$

$$\text{if } X \notin L, \forall \text{ provers } P, \Pr_{rand.\ of\ V}(V(X) \text{ interacting with } PACCEPTS) \leq 0.01$$

Theorem 8. *There $\exists$ an interactive proof for coSAT (set of unsatisfiable boolean formulae).*

Theorem 9. *There $\exists$ an interactive proof for the value of the permanent, over $\mathbf{F}_q$.*

Theorem 9 implies theorem 8 by Valiant (there is a matrix whose Perm = # of satisfying assignments = 0)

Proof. Let $N_{ij} = \text{minor}_{ij}(m) \to \text{Perm}(m) = \sum_{i=1}^{n} m_{ij} \times \text{Perm}(N_{ij})$
So $\text{Perm}(m)$ can be computed from $\text{Perm}(N_{ij})_{i=1}^{n}$
We have matrices $N_{11}, \ldots, N_{1n} \in \mathbf{F}_q^{(n-1)\times(n-1)}$
For $q \gg n^{10}$, let $m = n - 1$

Let $q(T) \in (\mathbf{F}_q(T))^{m \times m} \to$ this represents a curve.
Pick $q(T)$ unique of degree $n-1$ subject to $q(1) = N_1, q(2) = N_2, \cdots, q(T)$ is a matrix of polynomials.
Verifier sends $q(T)$ to prover.
Prover gives $\text{Perm}(q(T)) \in \mathbf{F}_q(T)$ of degree $\leq n \times m$
Verifier checks that $\sum_{j=1}^{m} m_{ij} N(j) = \text{Perm}(m) = h(T)$
Verifier picks $t \in \mathbf{F}_q$ uniformly at random and asks prover to prove that $h(t) = \text{Perm}(q(t))$
Take an arbitrary prover.
Case 1: $h(t)$ is correct, everything is good...
Case 2: $h(t)$ is wrong, then $h(t) \neq \text{Perm}(q(T)) \implies$ they are distinct polynomials. Then $\Pr_t(h(t) = \text{Perm}(q(t))) \leq \frac{mn}{q} \leq \frac{n^2}{q} \leq \frac{1}{n^8}$

So with probability $\geq 1 - \frac{1}{n^8}$ we have to prove a wrong statement in the next round. So in n rounds (we are reducing n to (n-1) to (n-2) minors etc...), probability of cheating prover gets away $\leq n \frac{1}{n^8} = \frac{1}{n^7}$

□