FGC Lecture 10

1 Hardness from Derandomization

1. Kannan's lower bound

$$\Sigma_3^{\mathsf{E}} \not\subseteq \mathsf{Size}(2^{o(n)}))$$

By re-scaling to $T(n)$ s.t. $n^{\omega(1)} \leq T(n) \leq 2^n$, we get

$$\Sigma_3^{T(n)} \not\subseteq \mathsf{Size}(T(n)^{o(1)})$$

2. Last lecture we showed

Assume $\mathsf{EXP} \subseteq \mathsf{P/poly}$ and **CAPP** $\in \mathsf{TIME}[2^{n^{o(1)}}]$ (therefore in $\mathsf{NTIME}[2^{n^{o(1)}}]$),

a $\mathsf{EXP} \subseteq \Sigma_2^{\mathsf{P}} \subseteq \Sigma_3^{\mathsf{P}} \subseteq \mathsf{P}^{\mathbf{Perm}} \subseteq \mathsf{EXP}$ (Meyer's Theorem)

b If **Perm** $\in \mathsf{P/poly}$ then **Perm** $\in MA$.
Guess circuits. "Purify" it in probabilistic polynomial time. Use as oracle.

c $\mathsf{MA} \subseteq \mathsf{NTIME}[2^{n^{o(1)}}]$. Replace probabilistic checking with **CAPP**.

d $\mathsf{EXP} = \mathsf{P}^{\mathbf{Perm}} = \mathsf{MA} = \Sigma_3^{\mathsf{P}} = \mathsf{NTIME}[2^{n^{o(1)}}]$.

e Rescale: Choose a $T(n) = n^{\omega(1)}$ s.t. $\Sigma_3^{\mathsf{P}} \subseteq \mathsf{NEXP}$.

f So $\mathsf{NEXP} \not\subseteq \mathsf{P/poly}$.

❧ ❧ ❧

Theorem 1.1. If **PIT** $\in \mathsf{NTIME}[2^{n^{o(1)}}]$, then either **Perm** $\notin \mathsf{AlgP/poly}$, or $\mathsf{NEXP} \not\subseteq \mathsf{P/poly}$.

Proof. Assume **PIT** $\in \mathsf{NTIME}[2^{n^{o(1)}}]$, and $\mathsf{NEXP} \subseteq \mathsf{P/poly}$, **Perm** $\in \mathsf{AlgP/poly}$.

Only steps (2.b) and (2.c) in the above proof need to change.

(2.b)': If **Perm** $\in \mathsf{AlgP/poly}$, then $P^{\mathbf{Perm}} \subseteq \mathsf{NP}^{\mathbf{PIT}}$. (This is stronger than (2.b). Now we only allow to do PIT.)

(2.c)': Assume also **PIT** $\in \mathsf{NTIME}[2^{n^{o(1)}}]$. Then $\mathsf{P}^{\mathbf{Perm}} \subseteq \mathsf{NTIME}[2^{n^{o(1)}}]$.

Using downward self-reducibility, aka. expansion by minors, we check

$$\mathbf{Perm}(M_{n\times n}) = \sum_{i=1}^{n} m_{1,i}\mathbf{Perm}([M_{n\times n}]_{-1,i})$$

and

$$\mathbf{Perm}(M_{1\times 1}) = m_{1,1}$$

Given C_n, an algebraic circuit that computes **Perm** on $n \times n$ matrices. We define $C_1, \dots, C_{n-1}$ s.t.

$$C_i = \begin{pmatrix} 1 & & & \\ & \ddots & & \mathbf{0} \\ & & 1 & \\ & \mathbf{0} & & M_{i\times i} \end{pmatrix}$$

Verify the identities

$$C_i(M) = \sum_{j=1}^{i} m_{1,j} D_{i-1}(M_{-1,j})$$

using PIT.

Because (2.c)' helps us get rid of the **PIT** oracle, we have (2.d). □

2 Natural Proofs

How to prove circuit lower bounds?

(A) Characterize what circuits can compute.

(B) Show some particular function doesn't meet (A).

Definition 2.1. f is a boolean function, given by its truth table. A property $N(f)$ is either True or False. (We assume $N(f)$ is True when f is hard)

N is a *Natural Property* if

1. N is *constructive*: $N \in \mathsf{P} = \mathsf{TIME}[2^{O(n)}]$.
2. N is *useful*: If $N(f)$ then f doesn't have small circuits (say $n^{\omega(1)}$).
3. N is *large*: $\mathrm{Prob}_f[N(f)] \geq \Omega(1)$.

The proofs of Razborov-Smolensky Theorem and Switching Lemma are natural. (?)

❧ ❧ ❧

Let PRG $G : \{0,1\}^s \to \{0,1\}^{2s}$ has exponential security (2^{s^ϵ}).

We can construct $\hat{G} : \{0,1\}^s \to \{0,1\}^{2^{s^\delta}}$, that is "random access", given z, and compute $\hat{G}(z)$ in poly-time.

[Figure: PRG tree]

The depth of the tree is s^δ. We pick $\delta < \epsilon$.

Say we want the i-th element. We don't need to compute the whole tree.

Define $f_z(i) = \hat{G}(z)_i$ (the i-th bit of $\hat{G}(z)$). $\forall z, f_z \in \mathsf{Size}(s^{o(1)}) = \mathsf{Size}(|i|^{o(1)})$.

Let N be defined so that $\forall z, N(f(z))$ =False. But for random function f_R, $N(f_R)$ =True with reasonable probability.

So if we have strong cryptographic PRG, then we don't have natural proofs for circuit lower bound.

3 Easy Witness Lemma

In the definition of natrual property, we replace the "largeness" with "non-emptyness": $\text{Prob}_f[N(f)] \neq 0$. We call it "*barely natural*" property.

Theorem 3.1 (Paraphrase of Easy Witness Lemma from IKW (IKW used "sometimes non-empty"))**.** If there exists a barely natural property, then $\mathsf{NEXP} \not\subseteq \mathsf{P/poly}$.

Lemma 3.2. If there exists a barely natural property, then **CAPP** $\in \mathsf{NTIME}[2^{n^{o(1)}}]$.

Proof.

1. Given an instance of **CAPP**, set $m = n^\delta$ (the inverse of the "usefulness" function).
2. Guess a function F_n so that $N(F_n)$ holds.
   (F_n is an n-bit boolean function, given by truth table.)
   The time to verify is $2^{n^\delta} = 2^{o(n)}$.
3. Size(F_n) $\geq n^{O(1)}$ (largeness)
4. Use BFNW to construct a $G : \{0,1\}^{\mathsf{poly}(n)} \to \{0,1\}^n$, a PRG that is hard for size n.
5. Try all seeds to estimate circuit probability.

The total time is subexponential. □

❧ ❧ ❧

A *easy witness* (succinct witness) is a circuit $C(i) = y_i$ that computes the i-th bit of the witness.

Theorem 3.3 (Easy Witness Lemma)**.** All positive instances of all NEXP relations have easy witnesses iff $\mathsf{NEXP} \subseteq \mathsf{P/poly}$.

Proof.

- If x has easy witness: Search for all circuits. Then $\mathsf{NEXP} = \mathsf{EXP}$. And $\mathsf{EXP} \subseteq \mathsf{P/poly}$. (use the witness circuits to construct the poly-size circuit)
- If x doesn't have easy witness: Let $N_x(y) = R(x, y)$. N_x is a natural property. By 3.1, $\mathsf{NEXP} \nsubseteq \mathsf{P/poly}$.

□