

Lecture 9 : Derandomization and Circuit Lower Bounds

Instructor: Russell Impagliazzo

Scribe: Jiawei Gao

September 24, 2015

1 Complexity Classes

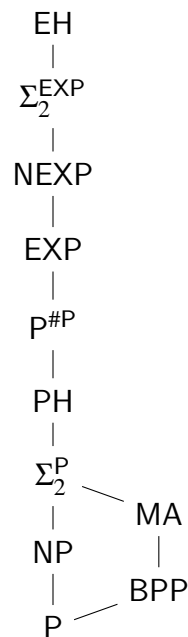


Figure 1: The relation of complexity classes mentioned in this lecture

1. Toda's Theorem

Theorem 1.1 (Toda). $PH \subseteq P^{\#P}$.

The proof of Toda's Theorem includes two steps.

Step 1: $PH \subseteq BPP^{\#P}$. (By Razborov-Smolensky, consider \exists as big \vee and \forall as big \wedge .)

Step 2: $BPP^{\#P} \subseteq P^{\#P}$.

2. Permanent of matrices

The *permanent* of an $n \times n$ matrix is defined as

$$\mathbf{Perm}(M) = \sum_{\sigma} \prod_{i=1}^n x_{i, \sigma(i)}$$

The sum is over all permutations of n elements.

Perm has *expansion by minors* property: $\mathbf{Perm}(M) = \sum_{i=1}^n x_{1i} \cdot \mathbf{Perm}(M_{1i})$.

Perm is equivalent with counting the number of perfect matchings in a bipartite graph.

Theorem 1.2 (Valiant). **Perm** is #P-complete.

3. Exponential Hierarchy

In polynomial hierarchy, a language in $\text{NP}(=\Sigma_1^P)$ iff

$$x \in L \Leftrightarrow \exists y R(x, y), |y| \leq \text{poly}(|x|), R \in P$$

. And Σ_k^P is defined by

$$x \in L \Leftrightarrow \exists y_1 \forall y_2 \dots \exists y_k R(x, y_1, \dots, y_k), |y_i| \leq \text{poly}(|x|), R \in P$$

.

Similarly, $\Sigma_1^{\text{EXP}}(=\text{NEXP})$ is defined by

$$x \in L \Leftrightarrow \exists y R(x, y) \quad |y| \leq \exp(|x|), R \in P$$

. And Σ_k^{EXP} is defined by

$$x \in L \Leftrightarrow \exists y_1 \forall y_2 \dots \exists y_k R(x, y_1, \dots, y_k) \quad |y_i| \leq \exp(|x|), R \in P$$

.

Here $R \in P$ is because the query length can be exponential in $|x|$.

Like Σ_k^P , we have inductive definition for Σ_k^{EXP} via oracle machines:

$$\Sigma_k^{\text{EXP}} = \text{NEXP}^{\Sigma_{k-1}^P}$$

By *padding argument* (padding the inputs to exponential length), we get the following theorem:

Theorem 1.3. If $\Sigma_k^P = \Sigma_{k+1}^P$, then $\Sigma_k^{\text{EXP}} = \Sigma_{k+1}^{\text{EXP}}$.

Fact: Collapses between classes translate upwards. Separations between classes translate downwards.

2 Derandomization and Circuit Lower Bounds

In this lecture we want to show:

Theorem 2.1 (IKW). If $\text{CAPP} \in \text{TIME}[2^{n^{o_c(1)}}]$, then $\text{NEXP} \not\subseteq \text{P/poly}$.

$o_c(1)$ denotes “computably little o of 1”. If $\varepsilon(n) = O_c(1)$, it means given n , we can compute $\varepsilon(n)$ efficiently.

Theorem 2.2 (KI). If $\text{PIT} \in \text{TIME}[2^{n^{o_c(1)}}]$, then $\text{NEXP} \not\subseteq \text{AlgP/poly}$.

3 Kannan’s Theorem

Theorem 3.1 (Kannan). There is a function $f \in \Sigma_3^{\text{EXP}}$ so that $\text{Size}(f) \geq \Omega(2^n/n)$.

“The hardest function there is”

Lemma 3.2. There exists a boolean function f of n variables such that $\text{Size}(f) \geq \Omega(2^n/n)$.

Proof. The i -th gate of a circuit can be represented as $g_i = \text{op}_i(g_{j_i}, g_{k_i})$, where op_i is the operation, and g_{j_i}, g_{k_i} are the two input gates. Let c be the total number of operations.

Since for each gate, there are c choices for op_i , and at most s choices for g_j and g_k , the number of different circuits with s gates is at most $(c \cdot s^2)^s$. However, the number of different boolean functions of n variables is 2^{2^n} , which equals the number of all possible truth tables, i.e. the number of 2^n binary strings. Therefore $(c \cdot s^2)^s \geq 2^{2^n}$, so that $s \geq \Omega(2^n/n)$. \square

“The first hardest function”

Lemma 3.3. There exists a boolean function f of n variables such that for all circuit C of size $O(2^n/n)$, C does not compute f , and for all function g prior to f in lexicographic order, there is a $O(2^n/n)$ size circuit C' that computes g .

Proof. We can find f using the following steps

1. Nondeterministically guess f ; ($O(2^n)$ time)
2. co-nondeterministically guess C and g ; ($O(2^n)$ time)
3. Nondeterministically guess C' . ($O(2^n/n)$ time)
4. Compute C on all inputs x and check if C does not compute f . Compute C' on all inputs x and check if C' computes g . ($O(2^n)$ time)

The algorithm runs in Σ_3^{EXP} . \square

4 Meyer's Theorem

Definition 4.1 (locally uniform circuits). A circuit of size s is *locally uniform* if given input x and index $i \in \{1, \dots, s\}$, we can compute (op_i, j_i, k_i) in polynomial time, where $g_i = (op_i, g_{j_i}, g_{k_i})$ is the i -th gate.

Exercise: Show that if TM M runs in time $T(|x|)$, then there is a locally uniform circuit C of size $O(T(|x|)^2)$ so that $\forall x, C_{|x|}(x) = M(x)$.

Fischer and Pippenger improved the size of locally uniform circuit from $O(T(|x|)^2)$ to $O(T(|x|) \log T(|x|))$.

Theorem 4.1 (Fischer-Pippenger). If TM M runs in time $T(|x|)$, then there is a locally uniform circuit C of size $O(T(|x|) \log T(|x|))$ so that $\forall x, C_{|x|}(x) = M(x)$.

Theorem 4.2 (Meyer). If $\text{EXP} \subseteq \text{P/poly}$, then $\text{EXP} \subseteq \Sigma_2^{\text{P}}$.

Proof. Suppose $L \in \text{EXP}$. Let C_n be a locally uniform circuit computing L .

The problem of computing the value of gate i of $C_n(x)$ on given (x, i) is in EXP. We call this problem GateL.

Assume $\text{EXP} \subseteq \text{P/poly}$. So GateL $\in \text{P/poly}$. Let D be the polynomial-size circuit computing GateL. Thus,

$$x \in L \Leftrightarrow \exists D, \forall i \in \{1, \dots, s\} [D(x, i) = op_i(D(x, j_i), D(x, k_i)) \text{ and } D(x, s)],$$

where s is the index of the final gate.

We can nondeterministically guess circuit D , and co-nondeterministically guess the first gate that messes up. Thus $L \in \Sigma_2^{\text{P}}$. \square

5 “The Collapsed Bookshelf” Argument

Suppose there is a heavy thing on the top level of a bookshelf. We want to argue that there is something heavy on the middle level of the bookshelf.

- If there is something heavy on the middle level, then we’ve done.
- If there isn’t such a heavy thing on the middle level, then the top level collapses, and the heavy thing on the top level falls to the middle level. Therefore there is something heavy on the middle level.

We use this argument to prove the following theorem.

Theorem 5.1. $\Sigma_2^{\text{EXP}} \not\subseteq \text{P/poly}$.

Proof. Either $\text{EXP} \subseteq \text{P/poly}$ or $\text{EXP} \not\subseteq \text{P/poly}$.

- If $\text{EXP} \not\subseteq \text{P/poly}$, then $\Sigma_2^{\text{EXP}} \not\subseteq \text{P/poly}$, because $\text{EXP} \subseteq \Sigma_2^{\text{EXP}}$.
- If $\text{EXP} \subseteq \text{P/poly}$, then by Meyer's Theorem, $\text{EXP} = \Sigma_2^{\text{P}} = \Sigma_3^{\text{P}}$. Then $\Sigma_2^{\text{EXP}} = \Sigma_3^{\text{EXP}}$. By Kannan's Theorem, it contains problems that require $\Omega(2^n/n)$ size circuits. (The bookshelf collapses.)

□

5.1 Proof of Theorem 2.1

We introduce two lemmas in proving this theorem.

Lemma 5.2. If $\text{Perm} \in \text{P/poly}$, then $\text{P}^{\text{Perm}} \subseteq \text{MA}$.

Lemma 5.3. If $\text{CAPP} \in \text{TIME}[2^{n^{o_c(1)}}]$, then $\text{MA} \subseteq \text{NTIME}[2^{n^{o_c(1)}}]$.

Assume $\text{CAPP} \subseteq \text{TIME}[2^{n^{o_c(1)}}]$. Either $\text{EXP} \subseteq \text{P/poly}$ or $\text{EXP} \not\subseteq \text{P/poly}$.

- If $\text{EXP} \not\subseteq \text{P/poly}$, then $\text{NEXP} \not\subseteq \text{P/poly}$, because $\text{EXP} \subseteq \text{NEXP}$.
- If $\text{EXP} \subseteq \text{P/poly}$, then $\text{EXP} = \Sigma_2^{\text{P}} = \text{P}^{\text{Perm}}$. The first equation is by Meyer's Theorem, and the second equation is by the fact $\Sigma_2^{\text{P}} \subseteq \text{PH} \subseteq \text{P}^{\text{Perm}} \subseteq \text{EXP}$, and all these classes collapse to Σ_2^{P} .

Thus if $\text{CAPP} \in \text{TIME}[2^{n^{o_c(1)}}]$, then By Lemmas 5.2 and 5.3, $\Sigma_3^{\text{P}} = \text{EXP} \subseteq \text{NTIME}[2^{n^{o_c(1)}}]$. So $\text{NEXP} \not\subseteq \text{P/poly}$.

5.2 Proof of Lemma 5.3

Use **CAPP** to deterministically verify witness.