FINE-GRAINED COMPLEXITY AND ALGORITHMS

Lecture 9 : Derandomization and Circuit Lower Bounds

Instructor: Russell Impagliazzo
Scribe: Jiawei Gao

September 24, 2015

1 Complexity Classes

Figure 1 shows the relation of complexity classes mentioned in this lecture.

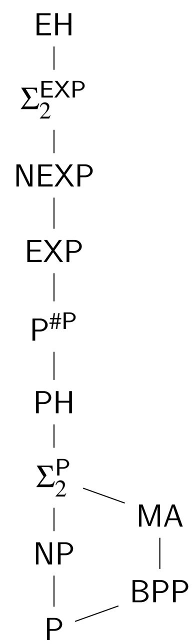

Figure 1: complexity classes

1. Toda's Theorem

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

The proof of Toda's Theorem inlcudes two steps.
Step 1: $\mathsf{PH} \subseteq \mathsf{BPP}^{\#\mathsf{P}}$. (By Razborov-Smolensky, consider $\exists$ as big $\vee$ and $\forall$ as big $\wedge$.)
Step 2: $\mathsf{BPP}^{\#\mathsf{P}} \subseteq \mathsf{P}^{\#\mathsf{P}}$.

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 $\mathsf{NP}(=\Sigma_1^{\mathsf{P}})$ iff

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

And Σ_k^{P} is defined by

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

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

$$x \in L \Leftrightarrow \exists y R(x,y)\ |y| \leq \exp(|x|), R \in \mathsf{P}.$$

And Σ_k^{EXP} is defined by

$$x \in L \Leftrightarrow \exists y_1 \forall y_2 \ldots \exists y_k R(x, y_1, \ldots, y_k)\ |y_i| \leq \exp(|x|), R \in \mathsf{P}.$$

Here $R \in \mathsf{P}$ 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:

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

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

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

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

2 Derandomization and Circuit Lower Bounds

In this lecture we want to prove that derandomization of **CAPP** implies circuit lower bounds.

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

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

Derandomization of **PIT** implies a similar lower bound for algebraic circuits.

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

3 Kannan's Theorem

Kannan's Theorem shows there are hard functions in Σ_3^{EXP}.

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

"The hardest function there is"

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

Proof. The i-th gate of a circuit can be represented as $g_i = op_i(g_{j_i}, g_{k_i})$, where op_i is the operation, and g_{j_i}, g_{k_i} are the two input gates. Let c be the number of operations, and let s be the circuit size. 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 $s \geq \Omega(2^n/n)$. □

"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}.

□

4 Meyer's Theorem

Definition 4.1 (locally uniform circuits)**.** A circuit of size s is *locally uniform* if given input x and index $i \in \{1,\ldots,s\}$, 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 $\mathsf{EXP} \subseteq \mathsf{P/poly}$, then $\mathsf{EXP} \subseteq \Sigma_2^{\mathsf{P}}$.

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

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

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 $L \in \Sigma_2^{\mathsf{P}}$. □

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. $\Sigma_2^{\mathsf{EXP}} \not\subseteq \mathsf{P}/\mathsf{poly}$.

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

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

$\square$

5.1 Proof of Theorem 2.1

We introduce two lemmas in proving this theorem.

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

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

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

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

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

5.2 Proof of Lemma 5.3

Use **CAPP** to deterministically verify witness.

5.3 Proof of Lemma 5.2

We show that If **Perm** $\in \mathsf{P}/\mathsf{poly}$, then $\mathsf{P}^{\textbf{Perm}} \in \mathsf{MA} \cap \mathsf{coMA}$.

Given Matrix M, Merlin gives a value v. We want to verify that $\textbf{Perm}(M) = v$. Suppose C_n is a circuit that supposedly computes **Perm**.

For $i = 1, \ldots, n$, we define circuit C_i to be

$$C_i = C_n \left(\begin{array}{ccc|c} 1 & & & \\ & \ddots & & \mathbf{0} \\ & & 1 & \\ \hline & \mathbf{0} & & M_{i \times i} \end{array} \right)$$

From the C_n given by Merlin, we compute $C_{n-1}, \ldots, C_1$.

We pick a random prime q. Say we have circuit C_i. We can use expansion by minors to compute $\mathbf{Perm}(M_{(i+1)\times(i+1)}) \mod q$. Then, check that $C_i(M_{(i+1)\times(i+1)}) \equiv \mathbf{Perm}(M_{(i+1)\times(i+1)}) \mod q$ by plugging in random values.

If we don't reject, with high probability that for any matrix $M_{i\times i}$, C_i computes its permanent mod q.

If C_n really does compute **Perm**, we will never reject for $i = 1, \ldots, n$.

If we accept the circuit C_n, we can use C_n to simulate the **Perm** oracle.