Topics in Theoretical Computer Science	March 21, 2016

Lecture 5 (Notes)

Lecturer: Ola Svensson *Scribes: Ola Svensson*

Disclaimer: These notes were written for the lecturer only and may contain inconsistent notation, typos, and they do not cite relevant works. They also contain extracts from the two main inspirations of this course:

1. The book *Computational Complexity: A Modern Approach* by Sanjeev Arora and Boaz Barak;
2. The course *http://theory.stanford.edu/ trevisan/cs254-14/index.html* by Luca Trevisan.

1 Introduction

Recall last lecture:

- Proof of Cook-Levin Theorem through Circuit SAT.
- Randomized computing (algorithm that is able to toss coins).
- Different randomized complexity classes:
  - **BPP** — polytime two-sided error
  - **RP** — polytime one-sided error (error on inputs in language)
  - **coRP** — polytime one-sided error (error on inputs not in language)
  - **ZPP** — expected polytime zero-sided error
- The error probability does not really matter as long as it is smaller than 1/2 because we can do error reduction through repetition. Recall Chernoff bounds as they are very useful!
- We believe that $\mathbf{BPP} = \mathbf{P}$. We know that $\mathbf{BPP} \subseteq \mathbf{EXP}$ but are unable to prove $\mathbf{BPP} \subsetneq \mathbf{NEXP}$.
- We showed that $\mathbf{BPP} \subseteq \mathbf{P}_{/\mathbf{poly}}$, i.e., languages in **BPP** have polysize circuits.

Today:

- Polynomial hierarchy

  > Synthesizing circuits is exceedingly difficult. It is even more difficult to show that a circuit found in this way is the *most* economical one to realize a function. The difficulty springs from the large number of essentially different networks available.
  >
  > — Claude Shannon, 1949

- Why we should expect that $\mathbf{NP} \subsetneq \mathbf{P}_{/\mathbf{poly}}$ (Karp-Lipton Theorem).
- If time, we also show that **BPP** is contained in the second level of the hierarchy (Sipser-Gács Theorem).

2 Definition of Polynomial Hierarchy

As a warm up, consider the familiar **NP**-complete language:

$$\text{INDSET} = \{\langle G, k\rangle : \text{ graph } G \text{ has an independent set of size} \geq k\}.$$

Recall the verifier definition of **NP**: A language L is in **NP** if there exists a polynomial p and a polynomial time TM M (the verifier) such that

$$x \in L \Leftrightarrow \exists u \in \{0,1\}^{p(|x|)} M(x,u) = 1 \qquad \forall x \in \{0,1\}^*.$$

Here u is called the certificate (or proof).

Based on this definition, it is clear that INDSET is in **NP**: given $\langle G, k\rangle$ and a subset S of vertices, we can check if $|S| \geq k$ and whether S is an independent set in polynomial time.

Let us now consider a slight modification to this problem:

$$\text{EXACT-INDSET} = \{\langle G, k\rangle : \text{ the largest independent set in } G \text{ has size exactly } k\}.$$

Here it does not look like we have short certificate as we do not only need to check if a given subset of vertices is independent, but we also need to verify that there is no larger independent set which seems hard.

We can also consider the problem of finding the smallest formula equivalent to a given formula (as in Shannon's quote):

$$\text{MIN-EQ-DNF} = \{\langle \varphi, k\rangle : \exists \text{DNF formula } \psi \text{ of size} \leq k \text{ that is equivalent to the DNF formula } \varphi\}$$

and the complement

$$\overline{\text{MIN-EQ-DNF}} = \{\langle \varphi, k\rangle : \forall \text{DNF formula } \psi \text{ of size} \leq k, \exists \text{ assignment } u \text{ s.t. } \varphi(u) \neq \psi(u)\}.$$

Again there is no obvious notation of certificate of membership for the above problems; it is the combination of quantifiers that make it hard! This motivates the following definition:

Definition 1 *The class Σ_i^p is the set of all languages L for which there exists a polynomial-time TM M and a polynomial q such that*

$$x \in L \Leftrightarrow \exists u_1 \in \{0,1\}^{q(|x|)} \forall u_2 \in \{0,1\}^{q(|x|)} \cdots Q_i u_i \in \{0,1\}^{q(|x|)} M(x, u_1, \ldots, u_i) = 1 \qquad \forall x \in \{0,1\}^*,$$

where Q_i denotes $\forall$ or $\exists$ depending on whether i is even or odd, respectively.

We can also start with $\forall$ instead of $\exists$:

Definition 2 *The class Π_i^p is the set of all languages L for which there exists a polynomial-time TM M and a polynomial q such that*

$$x \in L \Leftrightarrow \forall u_1 \in \{0,1\}^{q(|x|)} \exists u_2 \in \{0,1\}^{q(|x|)} \cdots Q_i u_i \in \{0,1\}^{q(|x|)} M(x, u_1, \ldots, u_i) = 1 \qquad \forall x \in \{0,1\}^*,$$

where Q_i denotes $\exists$ or $\forall$ depending on whether i is even or odd, respectively.

- The polynomial hierarchy is the set $\mathbf{PH} = \cup_i \Sigma_i^p = \cup_i \Pi_i^p$.
- It is also easy to see that $co\Sigma_i^p = \Pi_i^p$ (exercise).

Example 1 *We have* EXACT-INDSET *and* MIN-EQ-DNF *are in Σ_2^p. We have* $\overline{\text{MIN-EQ-DNF}} \in \Pi_2^p$. *Finally, we can see that* $\mathbf{NP} = \Sigma_1^p$ *and* $\mathbf{coNP} = \{\bar{L} : L \in \mathbf{NP}\} = \Pi_1^p$.

3 Properties of the Polynomial Hierarchy

We believe that $\mathbf{P} \neq \mathbf{NP}$ and $\mathbf{NP} \neq \mathbf{coNP}$ appealing to generalizations of these conjectures is that Σ_i^p is strictly contained in Σ_{i+1}^p. This conjecture is ofter referred to as "*the polynomial hierarchy does not collapse*".

We now show that if $\Sigma_i^p = \Sigma_{i+1}^p$ then $\mathbf{PH} = \Sigma_i^p$, i.e., the hierarchy collapses to its i:th level. We actually prove the following stronger statement:

Theorem 3 *For every $i \geq 1$, if $\Sigma_i^p = \Pi_i^p$, then $\mathbf{PH} = \Sigma_i^p$.*

Proof The statement follows from proving that $\Pi_i^p = \Sigma_i^p \Rightarrow \Pi_{k+1}^p = \Sigma_{k+1}^p = \Sigma_k^p$.

- For any language $L \in \Sigma_{i+1}^p$, there exists a polynomial q and a TM M such that

  $$x \in L \Leftrightarrow \exists u_1 \in \{0,1\}^{q(|x|)} \forall u_2 \in \{0,1\}^{q(|x|)} \cdots Q_{i+1} u_i \in \{0,1\}^{q(|x|)} M(x, u_1, \ldots, u_i) = 1 \qquad \forall x \in \{0,1\}^*,$$

- Notice that the expression following $\exists u_1 \in \{0,1\}^{q(|x|)}$ is a Π_k^p statement.
- Thus there is a language $L' \in \Pi_k^p$ such that

  $$x \in L \Leftrightarrow \exists u_1 \in \{0,1\}^{q(|x|)} (x, u_1) \in L'.$$

- Under the assumption $\Pi_k^p = \Sigma_k^p$, we have $L' \in \Sigma_k^p$, which means that

  $$(x, u_1) \in L' \Leftrightarrow \exists z_1 \forall z_2 \cdots Q_i z_i : M'((x, u_1), z_1, \ldots, z_i) = 1.$$

  where we omitted that $z_j \in \{0,1\}^{q(|x|)}$.

- It follows that

  $$\begin{aligned} x \in L &\Leftrightarrow \exists u_1 (x, u_1) :\in L' \\ &\Leftrightarrow \exists u_1 (\exists z_1 \forall z_2 \cdots Q_i z_i : M'((x, u_1), z_1, \ldots, z_i) = 1) \\ &\Leftrightarrow \exists (u_1, z_1) \forall z_2 \cdots Q_i z_i : M'((x, u_1), z_1, \ldots, z_i) = 1 \end{aligned}$$

  and so $L \in \Sigma_i^p$.

- The statement now follows from that for two complexity classes $\mathcal{C}_1$ and $\mathcal{C}_2$ we have that $\mathcal{C}_1 = \mathcal{C}_2$ implies $co\mathcal{C}_1 = co\mathcal{C}_2$. Therefore

  $$\Pi_{i+1}^p = co\Sigma_{i+1}^p = co\Sigma_i^p = \Pi_i^p = \Sigma_i^p$$

  so we have $\Pi_{i+1}^p = \Sigma_{i+1}^p = \Sigma_i^p$.

■

4 Exercises

Exercise 1 *Show that $\Pi_i^p = co\Sigma_i^p := \{\bar{L} : L \in \Sigma_i^p\}$. Also show that $\Pi_i^p \subseteq \Sigma_i^p$ implies $\Pi_i^p = \Sigma_i^p$.*

Exercise 2 (Each level has a complete problem) *Define $\Sigma_i SAT$ to be the language consisting of Boolean formulas φ such that $\exists u_1 \forall u_2 \ldots Q_i u_i \varphi(u_1, u_2, \ldots, u_i) = 1$. Show that $\Sigma_i SAT$ is a complete problem for Σ_i^p.*

(One can similarly define $\Pi_i SAT$ that is complete for Π_i^p.)

Exercise 3 (PH is unlikely to have complete problems) *Show that if there exists a language L that is* **PH** *complete, then there exists an i such that $\mathbf{PH} = \Sigma_i^p$, i.e., the hierarchy collapses to its i:th level.*

5 Karp-Lipton Theorem

Karp and Lipton showed that if SAT has small circuits the polynomial hierarchy collapses to its second level.

Theorem 4 *If* $\mathbf{NP} \subseteq \mathbf{P}_{/\mathbf{poly}}$ *then* $\mathbf{PH} = \Sigma_2^p$.

Proof By Theorem 3 (and first exercise), to show $\mathbf{PH} = \Sigma_2^p$, it suffices to show that $\Pi_2^p \subseteq \Sigma_2^p$. In particular, it suffices to show that Σ_2^p contains the Π_2^p complete language $\Pi_2 SAT$ consisting of all true formulas of the form

$$\forall u \in \{0,1\}^n \exists v \in \{0,1\}^n \varphi(u,v) = 1. \tag{1}$$

- If $\mathbf{NP} \subseteq \mathbf{P}_{/\mathbf{poly}}$ then there exists a polynomial p and a-$p(n)$ sized circuit family $\{C_n\}_{n\in\mathbb{N}}$ such that for every Boolean formula φ and for every $u \in \{0,1\}^n$

$$C_n(\varphi, u) = 1 \Leftrightarrow \text{there exists } v \in \{0,1\}^n \text{ such that } \varphi(u,v) = 1.$$

- Thus the circuit solves the decision problem problem. However, in the exercise below you are asked to convert it into a $q(n)$-sized circuit family $\{C'_n\}_{n\in\mathbb{N}}$, where q is a polynomial, satisfying

  for every formula φ and $u \in \{0,1\}^n$, if there is a string $v \in \{0,1\}^n$ such that $\varphi(u,v) = 1$, then $C'_n(\varphi, u)$ outputs such a string v.

- Of course, assuming $\mathbf{NP} \subseteq \mathbf{P}_{/\mathbf{poly}}$ only implies the existence of such a circuit family.
- The main idea of Karp-Lipton is that this circuit can be guessed using the $\exists$ quantifier.
- Formally, C'_n can be described using $q'(n) \approx q(n)^2$ many bits, we have the following quantified formula:

$$\exists w \in \{0,1\}^{q'(n)} \forall u \in \{0,1\}^n \text{ s.t. } w \text{ describes a circuit } C' \text{ and } \varphi(u, C'(\varphi, u)) = 1. \tag{2}$$

- On the one hand, if (1) is false, then for some u, no v exists such that $\varphi(u,v) = 1$ and hence (2) is false as well.
- On the other hand, if (1) is true, then (2) also is true (just guess w to describe the circuit C'_n that for each u outputs the correct v).
- As both evaluating the circuit C' can be done in polynomial time and also to check whether the resulting formula φ is satisfiable can be done in polynomial time, we have that $\Pi_2 SAT$ is in Σ_2^p if we assume $\mathbf{NP} \subseteq \mathbf{P}_{/\mathbf{poly}}$.

■

Exercise 4 *Given a polysize circuit C that given a formula φ outputs 1 iff the formula is satisfiable, design a polysize circuit C' that outputs a satisfying formula.*

6 BPP and Σ_2^p

This was first proved by Sipser and Gács. We give the much simpler proof by Lautemann.

Theorem 5 **BPP** $\subseteq \Sigma_2^p$.

Proof

- Let $L \in$ **BPP** then there exists a polynomial randomized algorithm M such that

$$\Pr_r[M(x,r) = L(x)] \geq 1 - \frac{1}{3m},$$

  where $m = |r|$ equals the number of random bits used by the algorithm. (This follows from the error reduction technique we saw last lecture.)

- That L is also in Σ_2^p follows from showing that

$$x \in L \Leftrightarrow \exists y_1, \ldots, y_m \in \{0,1\}^m \forall z \in \{0,1\}^m \bigvee_{i=1}^{m} M(x, y_i \oplus z) = 1.$$

- Indeed suppose that $x \in L$. Then

$$\begin{aligned}
&\Pr_{y_1,\ldots,y_m}[\exists z : M(x, y_1 \oplus z) = \cdots = M(x, y_m \oplus z) = 0] \\
\leq &\sum_{z \in \{0,1\}^m} \Pr_{y_1,\ldots,y_m}[M(x, y_1 \oplus z) = \cdots = M(x, y_m \oplus z) = 0] \\
\leq &\frac{2^m}{(3m)^m} < 1.
\end{aligned}$$

  So there exists $y_1, \ldots, y_m$ so that for all z we have $\bigvee_{i=1}^m M(x, y_i \oplus z) = 1$ as required.

- Conversely suppose $x \notin L$. Then fix any sequence $y_1, \ldots, y_m$. We have

$$\begin{aligned}
\Pr_z\left[\bigvee_{i=1}^{m} M(x, y_i \oplus z) = 1\right] &\leq \sum_i \Pr_z[M(x, y_i \oplus z) = 1] \\
&\leq m \cdot \frac{1}{3m} = 1/3.
\end{aligned}$$

  Hence, for any sequence $y_1, \ldots, y_m$ there is a z such that $\bigvee_{i=1}^m M(x, y_i \oplus z) = 1$ is false, as required.

■

Remark As **BPP** is closed under complement, the above proof shows that **BPP** $\subseteq \Sigma_2^p \cap \Pi_2^p$.