INFR11102: Computational Complexity 22/10/2019

Lecture 11: Polynomial hierarchy

Lecturer: Heng Guo

1 Polynomial hierarchy

We could easily extend the definition of coNP further, leading towards the polynomial hierarchy introduced by Meyer and Stockmeyer [MS72, Sto76].

Definition 1. *The class Σ_k^p consists of all languages L such that there exists a polynomial-time TM M and polynomials $q_1(\cdot), \cdots, q_k(\cdot)$ satisfying*

$$x \in L \Leftrightarrow \exists y_1 \forall y_2 \cdots (\exists/\forall) y_k,\ |y_i| \leq q_i(|x|) \text{ and } M(x, y_1, \cdots, y_k) = 1.$$

Similarly, The class Π_k^p consists of all languages L such that there exists a polynomial-time TM M and polynomials $q_1(\cdot), \cdots, q_k(\cdot)$ satisfying

$$x \in L \Leftrightarrow \forall y_1 \exists y_2 \cdots (\exists/\forall) y_k,\ |y_i| \leq q_i(|x|) \text{ and } M(x, y_1, \cdots, y_k) = 1.$$

For any $k \geq 1$, let $\Delta_k^p := \Sigma_k^p \cap \Pi_k^p$.
The polynomial hierarchy *is defined as* $\mathtt{PH} := \cup_{k \in \mathbb{N}} \Sigma_k^p$.

Here we list a few basic properties of these classes.
It is easy to see that $\Sigma_1^p = \mathtt{NP}$ and $\Pi_1^p = \mathtt{coNP}$. More generally, $\Pi_k^p = \{L \mid \overline{L} \in \Sigma_k^p\}$ for all k.

It is commonly believed that PH has infinite levels, namely that it does not collapse to some fixed level. The next theorem is a sufficient condition for it to collapse.

Theorem 1. *If $\Sigma_k^p = \Pi_k^p$ for some k, then $\mathtt{PH} = \Sigma_k^p = \Pi_k^p$.*

To show Theorem 1, we need a simple lemma.

Lemma 2. *For any integer $k \geq 0$,*

$$(\Sigma_k^p \cup \Pi_k^p) \subseteq (\Sigma_{k+1}^p \cap \Pi_{k+1}^p).$$

The proof of Lemma 2 is straightforward from Definition 1.

Proof of Theorem 1. First notice that if $\Sigma_k^p \subseteq \Pi_k^p$ or $\Pi_k^p \subseteq \Sigma_k^p$, then they must be equal. If, say $\Sigma_k^p \subseteq \Pi_k^p$ and $L \in \Pi_k^p$, then its complement $\overline{L}$ is in Σ_k^p. It impies that $\overline{L} \in \Pi_k^p$. Hence $L = \overline{\overline{L}} \in \Sigma_k^p$.

Given this, we only need to show that $\Sigma^p_{k+1} \subseteq \Sigma^p_k$, since by Lemma 2, it implies that $\Sigma^p_{k+1} \subseteq \Pi^p_{k+1}$ and therefore $\Pi^p_{k+1} = \Sigma^p_{k+1} = \Sigma^p_k = \Pi^p_k$. The theorem holds by induction (whose validity will become clear later) from there .

We will show that if $\Sigma^p_1 = \Pi^p_1$, then $\Sigma^p_2 \subseteq \Sigma^p_1$. The proof easily generalizes to other k (and hence induction works). For $L \in \Sigma^p_2$, there exist polynomials q_1 and q_2 and a poly-time TM M such that

$$\begin{aligned} x \in L &\Leftrightarrow \exists y_1 \forall y_2 \text{ s.t. } |y_i| \leq q_i(|x|) \text{ and } M(x, y_1, y_2) = 1 \\ &\Leftrightarrow \exists y_1 \text{ s.t. } \langle x, y_1 \rangle \in L', \end{aligned} \tag{1}$$

where L' is defined as follows

$$\langle x, y_1 \rangle \in L' \Leftrightarrow \forall y_2 \text{ s.t. } |y_2| \leq q_2(|x|) \text{ and } M(x, y_1, y_2) = 1.$$

It is clear that $L' \in \Pi^p_1 = \Sigma^p_1$. Hence, there is a polynomial q_2' and a TM M' such that

$$\langle x, y_1 \rangle \in L' \Leftrightarrow \exists y_2 \text{ s.t. } |y_2| \leq q_2'(|x|) \text{ and } M'(\langle x, y_1 \rangle, y_2) = 1.$$

Now, going back to (1), we can rewrite L in Σ^p_1 form:

$$x \in L \Leftrightarrow \exists y_1 \exists y_2 \text{ s.t. } |y_1| \leq q_1(|x|),\ \ |y_2| \leq q_2'(|x|) \text{ and } M''(x, y_1, y_2) = 1,$$

where the machine M'' mimics M' except that it decouples $\langle x, y_1 \rangle$. $\square$

Last time we talked about the graph isomorphism problem GI. In fact, we will show (toward the end of the course) that if GI is NP-complete, then $\Sigma^p_2 = \Pi^p_2$ and the hierarchy collapses. This is an evidence against GI being NP-complete.

Complete languages for Σ^p_k and Π^p_k are similar to SAT except that we need to change the quantifier accordingly. We define the following problem of the validity of quantified Boolean formulae (QBF).

Name: QBF_k

Input: A Boolean formula $\exists X_1, \forall X_2, \cdots, (\exists/\forall) X_k\ \varphi(X_1, \cdots, X_k)$ where φ is quantifier-free.

Output: Is the formula valid?

The following is a straightforward generalization of Cook-Levin theorem.

Theorem 3. *QBF_k is Σ^p_k-complete (under Karp's reduction).*

Remark (Bibliographic). The name of polynomial hierarchy comes from its similarity of the arithmetical hierarchy in mathematical logic. Relevant chapters are [AB09, Chapter 5] and [Pap94, Chapter 17].

2 TQBF and PSpace

Along the same vein of QBF_k, we define the following problem of the validity of totally quantified Boolean formulae (TQBF).

Name: TQBF

Input: An integer k and a Boolean formula $\exists X_1, \forall X_2, \cdots, (\exists/\forall) X_k \; \varphi(X_1, \cdots, X_k)$ where φ is quantifier-free.

Output: Is the formula valid?

The difference between TQBF and QBF_k are that there is no *fixed* level of quantifier alternations in TQBF. The integer k is an input in TQBF.

Theorem 4. *TQBF is* `PSpace`*-complete.*

Proof. One direction is easy, namely TQBF $\in$ `PSpace`. Once again, to achieve a space-efficient algorithm, we use recursion. If the leading quantifier is $\exists x$, then we recursively check the two cases of setting x to 0 and 1, and return true if one of them is true. Similarly, if the leading quantifier is $\forall x$, then we recursively check the two cases of setting x to 0 and 1, and return true if both of them are true. At any point of the recursion, we will only need polynomial space. The recursion depth is at most n, and therefore this is a polynomial space algorithm.

For the other direction, let M be a TM with space bound $s(n)$ and x be an input. Recall that M accepts x if and only if there is an accepting path from q_0 to q_{acc} in the configuration graph $G_{M,x}$, whose number of vertices is $2^{cs(n)}$ for some constant c. Next we express this property by a TQBF φ.

The basic idea is the same as Savitch's theorem. To encode that q_1 can reach q_2 in 2^ℓ steps, denoted $q_1 \to_{2^\ell} q_2$, we go through all possible middle points q'. Namely we ask whether $\exists q'(q_1 \to_{2^{\ell-1}} q') \wedge (q' \to_{2^{\ell-1}} q_2)$. Now, notice that if we recursively expand the $\to$ inside, we would end up with an exponential size formula. The trick, is to rewrite $(q_1 \to_{2^{\ell-1}} q') \wedge (q' \to_{2^{\ell-1}} q_2)$ as

$$\forall x, y \quad ((x = q_1 \text{ and } y = q') \vee (x = q' \text{ and } y = q_2)) \Rightarrow (x \to_{2^{\ell-1}} y).$$

Basically, we trade one $\to$ with a $\forall$ quantifier and a couple of new variables. Now, we may recursively expand $\to$ inside the expression.

We apply this construction to $q_0 \to_{2^{cs(n)}} q_{acc}$. The depth of this procedure is $\log 2^{cs(n)} = cs(n)$. Thus we end up with a TQBF whose length is $O(s(n))$. This TQBF is valid if and only if there is an accepting path in $G_{M,x}$, and the final formula has polynomial size and is computed in polynomial time. $\square$

Clearly QBF_k is a special case of TQBF for any k. Hence, `PH` $\subseteq$ `PSpace` by Theorem 3, Lemma 2, and Theorem 4.

TQBF captures many problems in game theory. Think of odd quantifiers (all are $\exists$) as the strategy of player one, and even quantifiers (all are $\forall$) as the counter-strategy of player two, and the Boolean formula encodes the claim that "player one wins". Then the validity of such a formula asks the existence of a winning strategy of player one. Asymptotic versions of many natural games, like Chess and Go, are indeed `PSpace`-complete.

Remark (Bibliographic). The name of polynomial hierarchy comes from its similarity of the arithmetical hierarchy in mathematical logic. Relevant chapters are [AB09, Chapter 4.2] and [Pap94, Chapter 19].

References

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009.

[MS72] Albert R. Meyer and Larry J. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *SWAT (now known as FOCS)*, pages 125–129. IEEE Computer Society, 1972.

[Pap94] Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[Sto76] Larry J. Stockmeyer. The polynomial-time hierarchy. *Theor. Comput. Sci.*, 3(1):1–22, 1976.