CMPT 407: Computational Complexity **Summer 2017**

Lecture 3: Polynomial Hierarchy (29 May - 2 June)

Lecturer: Valentine Kabanets *Scribe: Lily Li*

3.1 Polynomial Hierarchy

Definition 3.1 *For $i \geq 1$, a language L is in $\sup_2^P$ if there exists a polynomial-time TM M and a polynomial q such that*

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

where Q_i is a $\forall$ or a $\exists$ depending if i is even or odd. ***Polynomial hierarchy (PH)*** *is $\mathsf{PH} = \cup_i \sum_i^P$.*

The polynomial hierarchy does not collapse.

Theorem 3.2 *The following hold:*

1. *For every $i \geq 1$, if $\sum_i^P = \prod_i^P$ then $\mathsf{PH} = \sum_i^P$ i.e. the hierarchy collapses to the ith level.*
2. *If $\mathsf{P} = \mathsf{NP}$ then $\mathsf{PH} = \mathsf{P}$ i.e. the hierarchy collapses to P.*
3. *If $\mathsf{NP} = \mathsf{coNP}$, then $\mathsf{PH} = \mathsf{NP}$ e.e. the hierarchy collapses to NP.*

Proof: (We will be doing the third one.) Proof by induction on i. Consider $L \in \Sigma_2^p$. Interpret every element $x \in L$ as $\exists y \forall z : R(x, y, z)$. Think of a special language $L' = \phi(x, y)$ such that $R'(\phi(x, y), z) = R(x, y, z)$. Then we have $\forall z : R'(\phi(x, y), z)$ which is an instance of Π_1^p. Since we assumed that $\Pi_1^p = \Sigma_1^p$, it follows that $\exists w : R"(\phi(x, y), w) = \forall z : R'(\phi(x, y), z)$. Thus we have collapsed Σ_2^p to Σ_1^p. In general, if you want to collapse Σ_i^p to Σ_1^p you can do it by induction. Item two is a corollary of item three (*what we just showed*). The proof of item one is very similar to the proof of item three. In the base case we have that $\sum_i^P = \prod_i^P$. Suppose for $n > i$, $\sum_i^P = \sum_n^P$. Show that the same holds for $n + 1$. Consider $L \in \sum_{n+1}^P$. Then there exists a polytime polybalanced relation R such that x is in L if and only if $\exists y_1 \forall y_2 \exists y_3 \cdots \exists y_{n+1} R(x, y_1, y_2, ..., y_{n+1}$. Consider the language $L' = \{(x, y) : \forall y_2 \exists y_3 \cdots \exists y_{n+1} R(x, y, y_2, ..., y_{n+1}\}$. The key is to think of (x, y) as a single variable and observe that $L' \in \Pi_n^P$. Since $\Pi_n^P = \Sigma_n^P$, it must also be the case that $L' \in \Sigma_n^P = \Sigma_i^P$. Thus L' consists of all (x, y) such that $\exists y_1 \forall y_2 \exists y_3 \cdots R'((x, y), y_1, y_2, ..., y_i)$. Thus $x \in L$ if and only if $\exists (y, y_1) \forall y_2 \exists y_3 \cdots R'(x, (y, y_1), y_2, ..., y_i)$ and $L \in \Sigma_i^P$. ■

In addition to number 3, we can show the following:

Theorem 3.3 *If $\mathsf{PH} = \Sigma_i^p$ for some $i \geq 1$, then $\Sigma_i^p = \Pi_i^p$.*

Proof: To see this, observe that $\Pi_i^p \subset \Sigma_i^p$. Taking the complement of this inclusion, we get that $\Sigma_i^p \subset \Pi_i^p$. ■

Together we have that for $i \geq 1$, $\mathsf{PH} = \Sigma_i^p$ if and only if $\Sigma_i^p = \Pi_i^p$.

An interesting result: Graph Isomorphism is *not* NP-complete, unless PH collapses $\mathsf{PH} = \Sigma_2^p$. This is proved by interactive proofs.

Now we are ready to prove the Time-Space trade-off for SAT.

Theorem 3.4 *(Fortnow)* $\mathsf{SAT} \notin \mathsf{TiSp}(n^{1.1}, n^{0.1})$. *This means you have $O(n^{1.1})$ time and $O(n^{0.1})$ space, then you definitely cannot solve* SAT.

Proof: We will first prove the following lemma.

Lemma 3.5 $\mathsf{NTime}(n) \not\subseteq \mathsf{TiSp}(n^{1.2}, n^{0.2})$.

Proof: This lemma is quite difficult to prove and it requires a good handle on a large number of moving parts. Pay attention! First we define Σ_2-computation with running time t: these are the languages with formulas

$$\phi(x) = \exists y \in \{0,1\}^{O(t(n))} \forall z \in \{0,1\}^{O(t(n))} \psi(x, y, z)$$

where ψ is a predicate computable in deterministic time. Lets begin. Suppose for a contradiction that $\mathsf{NTime}(n) \subseteq \mathsf{TiSp}(n^{1.2}, n^{0.2})$. By padding the inputs to all languages in $\mathsf{NTime}(n)$ we have that $\mathsf{NTime}(n^{10}) \subseteq \mathsf{TiSp}(n^{12}, n^2)$. We will now show that $\mathsf{TiSp}(n^{12}, n^2) \subset \Sigma_2\mathsf{Time}(n^8)$. That is, by introducing some alternation we can remove the space bound.

Choose any language $L \subset \mathsf{TiSp}(n^{12}, n^2)$ and let TM M decide L. Let x be an input to L of length n. x can be computed in $O(n^{12})$ and $O(n^2)$ time and space simultaneously. Equivalently, $x \in L$ if and only if $\exists c_1, \exists c_2 \cdots \exists c_{n^{12}} R(x, c_1, ..., c_{n^{12}})$. Were each c_i is a configuration on the work tape so $|c_i| \leq O(n^2)$. Divide the configurations into n^6 blocks of n^6. We will only consider the first and last configurations as well as configurations between two blocks. These $n+1$ configurations are: $c_0, c_{n^6}, c_{2n^6}, ..., c_{n^{12}}$. We forms a $\Sigma_2\mathsf{Time}(n^8)$ as follows:

$$\exists (c_0, c_{n^6}, c_{2n^6}, ..., c_{n^{12}}) \forall i \in 1, ..., n^6 : c_{in^6} \text{ can be reached from } c_{(i-1)n^6} \text{ in } O(n^6) \text{ steps.}$$

here $(c_0, c_{n^6}, ..., c_{n^{12}})$ is consider one large input. Checking that c_{in^6} can be reached from $c_{(i-1)n^6}$ can be done in $O(n^8)$ time since you simply need to keep track of the n^2 bit configuration tape over n^6 time steps.

Next we need to see that $\mathsf{NTime}(n) \subseteq \mathsf{Time}(n^{1.2})$ then $\Sigma_2\mathsf{Time}(n^8) \subset \mathsf{NTime}(n^{9.6})$. What we are not going to do is trade alternation for non-determinism. First note that $\Sigma_2\mathsf{Time}(n^8)$ is of the form $\exists y \in \{0,1\}^{O(|x|^8)} \forall z \in \{0,1\}^{O(|x|^8)} : \psi(x, y, z)$. Just like in the collapsing Polynomial Hierarchy proof we can rewrite this as: all inputs (x, y), $\forall z, \psi(x, y, z)$. If you squint a little this should look a coNP instance. If $\mathsf{NTime}(n) \subset \mathsf{Time}(n^{1.2})$ then $\mathsf{coNTime}(n) \subseteq \mathsf{Time}(n^{1.2})$ as well (why?). Look carefully and you will notice that $1.2 \times 8 = 9.6$. This is not a coincidence. With padding, we have that $\mathsf{coNTime}(n^8) \subseteq \mathsf{Time}(n^{9.6})$.

Since we have $\mathsf{NTime}(n) \subset \mathsf{TiSp}(n^{1.2}, n^{0.2})$ by assumption, $\mathsf{NTime}(n) \subset \mathsf{Time}(n^{1.2})$ (just ignore the space constraint), so indeed we can make the above conversion. Through this chain of inclusions we have reached $\mathsf{NTime}(n^{10}) \subset \mathsf{NTime}(n^{9.6})$, but this contradicts non-deterministic time hierarchy so our assumption $\mathsf{NTime}(n) \subset \mathsf{TiSp}(n^{1.2}, n^{0.2})$ is false. ■

Why is this sufficient? Well, for any language in time $\mathsf{NTime}(t(n))$ can be reduced to a SAT-instance of size $O(t \log t)$. Where the reduction itself takes $\mathsf{poly}(\log n)$ space and time (how?). Thus if $\mathsf{SAT} \in \mathsf{TiSp}(n^{1.1}, n^{0.1})$ then $\mathsf{NTime}(n) \subseteq \mathsf{TiSp}(n^{1.1}\mathsf{poly}(\log n), n^{0.1}\mathsf{poly}(\log n))$. ■

Using the above techniques we can improve the above time and space bounds but we cannot get to quadratic space unfortunately. There are also a lot of other weird statements in complexity of the form

$$\text{unlikely statement} \implies \text{superunlikely statement}$$

here are a sampling:

Proposition 3.6 *(Karp-Lipton) If* $\mathsf{NP} \subseteq \mathsf{PolySize} \implies \mathsf{PH} = \Sigma_2^p$.

Proof:

■

Proposition 3.7 *(A. Meyer).* $\mathsf{EXP} \subseteq \mathsf{PolySize} \implies \mathsf{EXP} = \Sigma_2^p$.

Proof:

■