Lecture 9: Polynomial-Time Hierarchy, Time-Space Tradeoffs

Feb 3, 2016

Lecturer: Paul Beame *Scribe: Paul Beame*

1 The Polynomial-Time Hierarchy

Last time we defined problems

$$EXACT\text{-}INDSET = \{[G,k] \mid \text{ the largest independent set of } G \text{ has size } = k\},$$

$$MINDNF = \{[\varphi,k] \mid \varphi \text{ is a DNF that has an equivalent DNF of size } \leq k\},$$

and the complexity classes Σ_2^P and its dual Π_2^P. Σ_2^P and Π_2^P were defined in analogy with NP and coNP except that there are two levels of quantifiers with alternation $\exists\forall$ and $\forall\exists$, respectively. We observed that $MINDNF \in \Sigma_2^P$ and $EXACT\text{-}INDSET \in \Sigma_2^P \cap \Pi_2^P$.

More generally we have the definition:

Definition 1.1. *Σ_k^P is the set of $A \subseteq \{0,1\}^*$ such that there exist polynomials $p_1,\ldots,p_k$ and polynomial-time verifier V such that*

$$x \in A \;\Leftrightarrow\; \exists y_1 \in \{0,1\}^{p_1(|x|)} \forall y_2 \in \{0,1\}^{p_2(|x|)} \ldots Q_k y_k \in \{0,1\}^{p_k(|x|} \; (V(x,y_1,\ldots,y_k)=1)$$

where $Q_k = \exists$ if k is odd and $Q_k = \forall$ if k is even.
$\Pi_k^P = \{\overline{L} \mid L \in \Sigma_k^P\}$; alternatively, it is the set of $B \subseteq \{0,1\}^*$ such that there exist $p_1,\ldots,p_k$ and V such that

$$x \in B \;\Leftrightarrow\; \forall y_1 \in \{0,1\}^{p_1(|x|)} \exists y_2 \in \{0,1\}^{p_2(|x|)} \ldots Q_k y_k \in \{0,1\}^{p_k(|x|} \; (V(x,y_1,\ldots,y_k)=1)$$

where $Q_k = \forall$ if k is odd and $Q_k = \exists$ if k is even.

In general one often says that there are k *alternations*[1] in each of these definitions where k is the number of quantifier blocks in them. The following properties are all immediate from the definitions.

Proposition 1.2. *1. $\Sigma_k^P \subseteq \Pi_{k+1}^P$ and $\Pi_k^P \subseteq \Sigma_{k+1}^P$.*

2. $\mathsf{NP} = \Sigma_1^{\mathsf{P}}$ and $\mathsf{coNP} = \Pi_1^{\mathsf{P}}$.

[1] even though there are only $k-1$ switches from one kind of quantifier to the other

3. $\mathsf{P} = \Sigma_0^{\mathsf{P}} = \Pi_0^{\mathsf{P}}$.

Definition 1.3. *The* polynomial-time hierarchy $\mathsf{PH} = \bigcup_{\mathsf{k}} \Sigma_{\mathsf{k}}^{\mathsf{P}} = \bigcup_{\mathsf{k}} \Pi_{\mathsf{k}}^{\mathsf{P}}$.

We can define the notion of completeness for each of these classes in the same way as we have defined it for other classes above P.

Definition 1.4. *B is* Σ_k^P-complete *(respectively* Π_k^P-complete*) iff*

1. *$B \in \Sigma_{\mathsf{k}}^{\mathsf{P}}$ (respectively Π_k^P), and*
2. *For all $A \in \Sigma_{\mathsf{k}}^{\mathsf{P}}$ (respectively Π_k^P), $A \leq_P B$.*

We see that natural restrictions of $TQBF$ form the complete problems for all levels

Definition 1.5. *Define $\Sigma_k SAT$ to be the set of quantified Boolean formulas of the form*

$$\exists \vec{x}_1 \forall \vec{x}_2 \cdots Q_k \vec{x}_k \varphi(\vec{x}_1, \ldots \vec{x}_k)$$

that evaluate to true. $\Pi_k SAT$ is the dual set of the form

$$\forall \vec{x}_1 \exists \vec{x}_2 \cdots Q_k \vec{x}_k \varphi(\vec{x}_1, \ldots \vec{x}_k)$$

that evaluate to true.

The previous arguments for the proof of the Cook-Levin theorem immediately extend to show the following:

Proposition 1.6. *$\Sigma_k SAT$ is* $\mathsf{Sigma}_{\mathsf{k}}^{\mathsf{P}}$*-complete and $\Pi_k SAT$ is $\Pi_{\mathsf{k}}^{\mathsf{P}}$-complete.*

Umans showed the following via a much more difficult proof.

Fact 1.7. *$MINDNF$ is Σ_2^P-complete.*

Observe also that the PH does not have a complete problem unless $\mathsf{PH} = \Sigma_{\mathsf{k}}^{\mathsf{P}}$ for some k. Any complete problem must be in $\Sigma_{\mathsf{k}}^{\mathsf{P}}$ for some fixed k and hence all of would be contained in it.

Theorem 1.8.

1. *$forall k \geq 1$ if $\Sigma_k^P = \Pi_k^P$ then $\mathsf{PH} = \Sigma_{\mathsf{k}}^{\mathsf{P}} \cap \Pi_{\mathsf{k}}^{\mathsf{P}} = \Sigma_{\mathsf{k}}^{\mathsf{P}}$. (In this case we say that PH "collapses to level k".*
2. *If $\mathsf{P} = \mathsf{NP}$ then $\mathsf{PH} = \mathsf{P}$.*

Proof. We first prove part 2. The proof is by induction that $\Sigma_k^P \subseteq \mathsf{P}$. The base case Σ_1^P subset P follows by assumption Assume that Σ_k^P is in P. Let $A \subseteq \Sigma_{k+1}^P$. We first assume that $k+1$ is odd. Then there are polynomials $p_1, \ldots, p_{k+1}$ and polynomial time verifier V such that

$$x \in A \Leftrightarrow \exists y_1 \in \{0,1\}^{p_1(|x|)} \forall y_2 \in \{0,1\}^{p_2(|x|)} \ldots \exists y_{k+1} \in \{0,1\}^{p_{k+1}(|x|} (V(x, y_1, \ldots, y_{k+1}) = 1).$$

Since $\mathsf{P} = \mathsf{NP}$ there is a polynomial-time algorithm W such that $W(x, y_1, \ldots, y_k) = 1$ iff $\exists y_{k+1} \in \{0,1\}^{p_{k+1}(|x|)} (V(x, y_1, \ldots, y_{k+1}) = 1$. By using the former instead of the latter we get that A is in Σ_k^P and hence in P by the inductive hypothesis. If $k+1$ is even then the $k+1$-st quantifier is $\forall$ which we can also express as $\neg\exists\neg$. We apply the $\mathsf{P} = \mathsf{NP}$ assumption to find a polynomial-time algorithm for $\exists y_{k+1} \in \{0,1\}^{p_{k+1}(|x|)} V(x, y_1, \ldots, y_{k+1}) \neq 1$ and complement its answer to obtain the same result.

The proof for part 1 uses a similar idea. For any $A \in \Sigma_i^P$ for $i > k$, since $\Sigma_k^P = \Pi_k^P$ we can replace the last k quantifiers by their dual. Rather than removing the last quantifier as in part 2, this will lead to two quantifiers of the same type next to each other of variables y_{i-k} and y_{i-k-1}. This can be described in terms of a single variable y' having bit-length the sum of those for the other two. This is one less alternation and so $\Sigma_i^P \subseteq \Sigma_{i-1}^P$. By induction $\mathsf{PH} \subseteq \mathsf{\Sigma_k^P}$ which implies the claim. $\square$

We now give an alternative characterization of the levels of PH using oracles for complete problems.

Theorem 1.9. $\mathsf{\Sigma_2^P} = \mathsf{NP^{SAT}}$ *and* $\mathsf{\Pi_2^P} = \mathsf{coNP^{SAT}}$. *More generally,* $\mathsf{\Sigma_{k+1}^P} = \mathsf{NP^{\Sigma_k SAT}}$.

Proof. We prove the case $\mathsf{\Sigma_2^P} = \mathsf{NP^{SAT}}$; the rest of the cases are similar.
$\underline{\mathsf{\Sigma_2^P} \subseteq \mathsf{NP^{SAT}}}$: Let $A \in \mathsf{\Sigma_2^P}$. Then there are q_1, q_2 and V such that

$$x \in A \Leftrightarrow \exists y_1 \in \{0,1\}^{q_1(|x|)} \neg\exists y_2 \in \{0,1\}^{q_2(|x|)} (V(x, y_1, y_2) \neq 1).$$

The $\mathsf{NP^{SAT}}$ algorithm guesses y_1 and calls the SAT oracle on the formula expressing $V(x, y_1, y_2) \neq 1$ and flips its answer. Therefore $A \in \mathsf{NP^{SAT}}$.
$\underline{\mathsf{NP^{SAT}} \subseteq \mathsf{\Sigma_2^P}}$: Let $A \in \mathsf{NP^{SAT}}$. Let $M_A^?$ be a polynomial-time oracle NTM and let $T(n)$ be its polynomial running time. The computation of M_A^{SAT} on input x is a tree that has branches of length $T(n)$. There are two sources of branching of M_A^{SAT}: the nondeterministic choices of $M_A^?$ itself, and the answers to the up to $T(n)$ calls to the SAT oracle, each of which may depend on the previous calls. To show that $A \in \Sigma_2^P$, we use the existentially quantified variables to guess: (1) the nondeterministic guesses $\vec{g}$ of $M_A^?$ on input x, (2) all of the formulas $\vec{\varphi}$ that are asked as questions that $M_A^?$ asks of the SAT oracle, (3) the answers $\vec{a}$ to each of the formulas asked to the SAT oracle, and (4) the satisfying assignments $\vec{\alpha}$ for each of the formulas φ_i for which the answer $a_i = 1$. There are universally quantified variables for potential assignments $\vec{\beta}$ for each of the formulas φ_i for which $a_i = 0$. The polynomial-time verifier then checks that (a) the computation is accepting, (b) that $\varphi_i(\alpha_i) = 1$ for each i such that $a_i = 1$, and (c) that $\varphi_i(\beta_i) = 0$ for each i such that $a_i = 0$. Therefore $A \in \Sigma_2^P$.

The same method works at higher levels also, using a $\Sigma_k SAT$ oracle instead of a SAT oracle. $\square$

2 Time-Space Tradeoffs for SAT

Definition 2.1. *Let* $\mathsf{DTIME\text{-}SPACE}(\mathsf{T}(\mathsf{n}), \mathsf{S}(\mathsf{n}))$ *be the set of languages* L *that are decided by a TM* M *that runs in time* $O(T(n))$ *and space* $O(S(n))$.

Now $\mathsf{DTIME\text{-}SPACE}(\mathsf{T}(\mathsf{n}), \mathsf{S}(\mathsf{n})) \subseteq \mathsf{DTIME}(\mathsf{T}(\mathsf{n})) \cap \mathsf{DSPACE}(\mathsf{S}(\mathsf{n}))$ but we do not know that the two are equal. For example we know that $PATH \in DTIME(n^2)$ and $PATH \in \mathsf{NL} \subseteq \mathsf{DSPACE}(\log^2 \mathsf{n})$ but we do not know whether or not $PATH$ is in $\mathsf{DTIME\text{-}SPACE}(\mathsf{n}^{\mathsf{O}(1)}, \log^{\mathsf{O}(1)} \mathsf{n})$.

Theorem 2.2 (Fortnow,Fortnow-Lipton-Van Melkebeek-Viglas)**.** *If* $(1+\varepsilon+2\delta)(1+\varepsilon) < 2$ *then*

$$\mathsf{NTIME}(\mathsf{n}) \not\subseteq \mathsf{DTIME\text{-}SPACE}(\mathsf{n}^{1+\varepsilon}, \mathsf{n}^{\delta}).$$

Before giving the proof we show that

Corollary 2.3. *For every* $\gamma > 0$, $SAT \notin \mathsf{DTIME\text{-}SPACE}(\mathsf{n}^{\sqrt{2}-\gamma}, \mathsf{n}^{\mathsf{o}(1)})$.

Proof. For $\gamma > 0$, we choose $\delta = \gamma/4$ and $\varepsilon = \sqrt{2}-1-2\gamma$. Then $(1+\varepsilon+2\delta)(1+\varepsilon) < (\sqrt{2}-\gamma)^2 < 2$. If the statement of the corollary is false, as we discussed in the simulation of Turing machines by circuits (and then formulas), every language in $\mathsf{NTIME}(\mathsf{n})$ is reducible to SAT in time $n \log^{O(1)} n$ using formulas of size $O(n \log n)$, and space $\log^{O(1)} n$. Therefore if SAT could be solved in the claimed time and space bounds it would violate the theorem with the above parameters. ☐

Proof of Theorem 2.2. Let $(1+\varepsilon+2\delta)(1+\varepsilon) < 2$ and suppose that

$$\mathsf{NTIME}(\mathsf{n}) \subseteq \mathsf{DTIME\text{-}SPACE}(\mathsf{n}^{1+\varepsilon}, \mathsf{n}^{\delta}).$$

We will show that this will imply a violation of the nondeterministic time hierarchy theorem. As we have seen in padding arguments we can substitute any time and space constructible function $g(n)$ for n. It follows that

$$\mathsf{NTIME}(\mathsf{n}^2) \subseteq \mathsf{DTIME\text{-}SPACE}(\mathsf{n}^{2+2\varepsilon}, \mathsf{n}^{2\delta}).$$

Suppose that $L \in \mathsf{DTIME\text{-}SPACE}(\mathsf{n}^{2+2\varepsilon}, \mathsf{n}^{2\delta})$, and let M_L be the associated TM deciding L that runs in time $c_T n^{2+2\varepsilon}$ and space $c_S n^{2\delta}$. By definition, $x \in L \Leftrightarrow$
$\exists$ a vector y describing a sequence of configurations $C_0, C_1, \ldots, C_{n^{1+\varepsilon}}$ of M_L, each of which is expressible in $O(n^{2\delta})$ bits, such that C_0 is the initial configuration of M_L on input x, $C_{n^{1+\varepsilon}}$ is an accepting configuration of M_L such that
$\forall\, i \in \{1, \ldots, n^{1+\varepsilon}\}$ there is a computation of M_L of length at most $c_T n^{1+\varepsilon}$ begining with C_{i-1} and ending in C_i.

The last part of the computation after the $\forall$ is described by a function $V(x, y, i)$ that is computed by a TM that runs in time $n^{1+\varepsilon+2\delta}$. Including the $\forall$, this can be expressed as $\neg\exists\neg V(x, y, i)$ and the part its first $\neg$ is a computation in $NTIME(n^{1+\varepsilon+2\delta})$. By padding with function $g(n) = n^{1+\varepsilon+2\delta}$, the assumption that $\mathsf{NTIME(n)} \subseteq \mathsf{DTIME\text{-}SPACE}(\mathsf{n}^{1+\varepsilon}, \mathsf{n}^{\delta})$ which implies that $\mathsf{NTIME(n)} \subseteq \mathsf{DTIME}(\mathsf{n}^{1+\varepsilon})$, also implies that $NTIME(n^{1+\varepsilon+2\delta}) \subseteq \mathsf{DTIME}(\mathsf{n}^{(1+\varepsilon+2\delta)(1+\epsilon)})$. Therefore the entire part of the computation beginning with the $\forall$ quantifier can be done in $\mathsf{DTIME}(\mathsf{n}^{(1+\varepsilon+2\delta)(1+\epsilon)})$.

Adding in the existentially quantified part, it follows that $L \in \mathsf{NTIME}(\mathsf{n}^{(1+\varepsilon+2\delta)(1+\epsilon)})$. Therefore $\mathsf{NTIME}(\mathsf{n}^2) \subseteq \mathsf{NTIME}(\mathsf{n}^{(1+\varepsilon+2\delta)(1+\epsilon)})$ which contradicts the nondeterministic time hierarchy theorem since $(1 + \varepsilon + 2\delta)(1 + \epsilon) < 2$. $\square$