18.405J/6.841J Advanced Complexity Theory	February 11, 1998
Lecture 4	
Lecturer: Daniel A. Spielman	*Scribe: Chris Peikert*

Polynomial Hierarchy

We will make two equivalent set of definitions of some complexity classes, and prove their equivalence. The first is presented here:

Definition 1 *Let $\Sigma_k \mathit{TIME}(f(n))$ be the class of languages L accepted by an Alternating Turing Machine (ATM) that begins in an existential ($\exists$) state, alternates between existential and universal states at most $k-1$ times, and halts (on all branches) within time $O(f(n))$. We also define*

$$\Sigma_k P = \bigcup_{i=1}^{\infty} \Sigma_k \mathit{TIME}(n^i).$$

Note that $NP = \Sigma_1 P$, since a NTM remains in an existential state over its entire computation.

Definition 2 $\Pi_k \mathit{TIME}(f(n))$ *and* $\Pi_k P$ *are defined similarly, but the ATM must begin in a universal ($\forall$) state.*

Note that co-$NP = \Pi_1 P$, and in general it can be shown directly that $\Pi_k P =$ co-$\Sigma_k P$. For consistency, we define the special cases $\Pi_0 P = \Sigma_0 P = P$. These definitions are reasonable if we think of the deciders as using *no* quantifiers at all, so that their execution is deterministic.

Note that, by definition, $\Sigma_k P \subseteq \Pi_{k+1} P$ and $\Pi_k P \subseteq \Sigma_{k+1} P$, since a machine can simply alternate quantifiers once at the beginning of its computation. This prompts the following definition:

Definition 3 *The polynomial hiearchy $PH = \bigcup_k \Sigma_k P = \bigcup_k \Pi_k P$.*

By the PSPACE-completeness of TQBF, it's easy to see that PH $\subseteq$ PSPACE, but it's an open question whether PSPACE $\subseteq$ PH.

There may be a temptation to say that PSPACE$(f(n)) \subseteq PH$ because $TQBF \subseteq \Sigma_{f(n)}\mathrm{TIME}(f(n))$, but this reaoning is flawed because in PH the number of alternations *may not* grow with n.

We now present an alternate set of definitions:

Definition 4 *First, $\Sigma_0 P = \Pi_0 P = P$. Next,*

$$\begin{aligned}
\Sigma_k P &= \{A \mid \exists\, B \in \Pi_{k-1}P,\ c > 0 \text{ such that } x \in A \iff \exists\, y \text{ such that } (x,y) \in B \text{ and } |y| \leq |x|^c\} \\
\Pi_k P &= \{A \mid \exists\, B \in \Sigma_{k-1}P,\ c > 0 \text{ such that } x \in A \iff \forall\, y \text{ such that } (x,y) \in B \text{ and } |y| \leq |x|^c\}
\end{aligned}$$

It should be clear that $\Sigma_1 P = NP$. These definitions generalize the notion of a "certificate" or "witness", which should be familiar from the standard interpretation of NP.

Theorem 5 *The two definitions of $\Sigma_k P$ are equivalent.*

Proof We can inductively "unwind" the second definition, if k is even:

$$\begin{aligned}
\Sigma_k P \quad = \quad & \{A \mid \exists\, B \in P, c_1, \ldots, c_k \geq 1 \text{ such that } x \in A \iff \\
& \exists\, y_1 \quad |y_1| \leq |x|^{c_1} \\
& \forall\, y_2 \quad |y_2| \leq (|x| + |y_1|)^{c_2} \\
& \ldots \\
& \forall\, y_k \quad |y_k| \leq \left(|x| + \sum_{i=1}^{k-1} |y_i|\right)^{c_k} \\
& \text{ such that } (x, y_1, y_2, \ldots, y_k) \in B\}
\end{aligned}$$

If k is odd, we get a similar definition. To show that definition 2 includes definition 4, we note that a $\Sigma_k P$ machine (from definition 2) can solve the conditions for membership in A, because the y_i values are short enough to guess in polynomial time, and because there are few enough of them.

To show that definition 4 includes definition 2, consider an ATM M, beginning in an $\exists$ state, changing state at most $k-1$ times, and running in polynomial time. Then we can produce a machine N with inputs of the form $(x, y_1, y_2, \ldots, y_k)$. N will simulate M, but it will use bits from y_1 in the first collection of existential states, the bits from y_2 in the next collection of universal states, etc. In other words, N simulates M by choosing an execution path based on the first bit of y_1, then at M's next $\exists$ state it branches on the second bit of y_1, etc. until M goes into a $\forall$ state, at which point it uses the bits of y_2, and so on. Then we have M accepts $x \iff \exists\, y_1 \forall\, y_2 \cdots \forall\, y_k$ N accepts $(x, y_1, y_2, \ldots, y_k)$ so $B = \mathrm{L(N)} \in \Sigma_k P$ (from definition 4). ■

Finally, we see another way of looking at $\Sigma_k P$ and $\Pi_k P$, based on *oracle Turing Machines.*

Definition 6 *An oracle TM (OTM) $M^?$ with oracle (a language) A has an oracle tape on which it can write a string w and enter an "oracle state" and, in one step, return to state "oracle yes" if $w \in A$ or "oracle no" if $w \notin A$.*

Note that $P^A = \{L \mid \exists$ a poly-time OTM $M^?$ that, on oracle A, accepts L$\}$. Consider that $P^{\mathrm{3SAT}} \supseteq NP$ (which is trivial to show), and also $P^{\mathrm{3SAT}} \supseteq$ co-NP since a machine can reject if the oracle replies "yes", and accept otherwise.

Similarly, $NP^A = \{L \mid \exists$ a nondet poly-time OTM $M^?$ that, on oracle A, accepts L$\}$. We will prove that that $NP^{\mathrm{3SAT}} = \Sigma_2 P$. More generally, we define the language $\exists\mathrm{QBF}_k$, which is the language of true QBFs, for which the first quantifier is $\exists$, and which alternate quantifiers at most $k-1$ times. Similarly, we will show that $NP^{\exists\mathrm{QBF}_k} = \Sigma_{k+1} P$.

Also, $\exists\mathrm{QBF}_k$ is complete for $\Sigma_k P$, but the proof will be done later (see the solutions to last year's problem sets for a preview).

Theorem 7 *If $\Sigma_k P = \Pi_k P$ for some $k \geq 1$, then $\Sigma_j P = \Sigma_k P\ \forall\ j > k$, in which case we say that "the PH collapses." (It's widely believed that PH does* not *collapse.)*

Proof We will use definition 4. Let $A \in \Sigma_{k+1} P \Rightarrow \exists\, c > 0, B \in \Pi_k P$ such that $x \in A \iff \exists\, y : |y| \leq |x|^c$ such that $(x, y) \in B$. If $\Pi_k P = \Sigma_k P, \exists\, c' > 0, C \in \Pi_{k-1} P$ such that $x \in A \iff \exists\, y : |y| \leq |x|^c$ and $\exists\, y' : |y'| \leq (|x| + |y|)^{c'}$ such that $(x, y, y') \in C$. But then we can just combine y and y' into one string, which implies that $A \in \Sigma_k P$, and the proof is complete. ■