Computational Complexity. Lecture 6

Polynomial Hierarchy

Alexandra Kolla

Today

- Definition of Polynomial Hierarchy
- Alternate characterization
- Some facts, and when does it collapse

The polynomial hierarchy

- Difference between NP and coNP is questions of the form "does there exist" (simple, efficient proofs) and "for all" (don't seem to have simple and efficient proofs).
- Formally, decision problem A is in NP iff there is poly-time procedure V(.,.) and polynomial bound p(.) such that

$$x \in A \Leftrightarrow \exists y \colon |y| \leq p(|x|) \land V(x, y) = 1$$

- Decision problem A is inco NP iff there is poly-time procedure V(.,.) and polynomial bound p(.) such that

$$x \in A \Leftrightarrow \forall y \colon |y| \leq p(|x|) \land V(x, y) = 1$$

Stacking quantifiers

- Suppose you had a decision problem A which asked

$$x \in A \Leftrightarrow \exists z\ s.t. |z| \leq p(|x|) \forall y\ s.t. |y| \leq p(|x|), V(x,z,y)$$

Example: given Boolean formula f, over variables $x_1, x_2, \ldots, x_n$ is there formula f' which is equivalent to f and is of size at most k?

- Member of the second level of the polynomial hierarchy $\sum_2$

The polynomial hierarchy

- Starts with familiar classes at level 1: $\sum_1 = NP$ and $\prod_1$=coNP.

- For all i, it includes two classes $\sum_i$ and $\prod_i$

A$\in \sum_i \Leftrightarrow \exists y_1 \forall y_2 \ldots Q y_i \, V_A(x, y_1, \ldots, y_i)$

B$\in \prod_i \Leftrightarrow \forall y_1 \exists y_2 \ldots Q' \, y_i \, V_B(x, y_1, \ldots, y_i)$

For clarity, I omitted the p(.) conditions but they are still there.

The polynomial hierarchy

- Easy to see that : $\prod_k = co\sum_k$.

- For all i<k, $\prod_i \subseteq \sum_k$, $\sum_i \subseteq \sum_k$, $\sum_i \subseteq \prod_k$, $\prod_i \subseteq \prod_k$ (ex)

An alternate characterization

- PH characterized in terms of “oracle machines”
- Oracle has certain power and can be consulted as many times as desired. Every consultation costs only one computational step at a time.
- Syntactically, let A be some decision problem and $\mathcal{M}$ a class of TM. Then $\mathcal{M}^A$ is the class of machines obtained from $\mathcal{M}$ by allowing instances of A to be solved in one step.

An alternate characterization

- If C is a complexity class, then $\mathcal{M}^{\mathrm{C}} = \bigcup_{A \in \mathrm{C}} \mathcal{M}^{A}$.

- If L is complete for C and the machines in $\mathcal{M}$ are powerful enough to compute poly-time computations, then $\mathcal{M}^{\mathrm{C}} = \mathcal{M}^{L}$.

An alternate characterization

- **Theorem.** $\sum_2 = NP^{3SAT}$

An alternate characterization

- **Theorem**. For every i>1, $\sum_i = NP^{\sum_{i-1}}$ (ex)

Additional properties

Here are some facts about PH that we will not prove:

- $\sum_i$ and $\prod_i$ have complete problems for all i.
- A $\sum_i$-complete problem is not in $\prod_j$, j<i, unless $\sum_i = \prod_j$.
- A $\sum_i$-complete problem is not in $\sum_j$, j<i, unless $\sum_i = \sum_j$
- Suppose $\sum_i = \prod_i$ for some i. Then $\sum_j = \prod_j = \sum_i = \prod_i$ for all j≥i .
- Suppose that $\prod_i = \prod_{i+1}$ for some i. Then$\sum_j = \prod_j = \prod_i$ for all j≥i .

Additional properties

Theorem. (Special case of (3) above)

Suppose NP=coNP. Then for every $i \geq 2$, $\sum_i$=NP.