CS 579: Computational Complexity. Lecture 3

NL=coNL, Polynomial Hierarchy

Alexandra Kolla

Today

- Randomized log-space
- Alternate characterization of NL
- NL=coNL
- Definition of Polynomial Hierarchy
- Alternate characterization
- Some facts, and when does it collapse

Randomized log-space

- Introduce randomized space-bounded TM (for simplicity only for decision problems).
  - Read-only input tape
  - Read/write work tape
  - Read-only random tape with one-way access (the head can only move from left to right)
- For every fixed input and fixed content of random tape, TM is completely deterministic and either accepts or rejects.

Randomized log-space

- For machine M, input x, random tape content r, denote M(r,x) the outcome of the computation.
- Decision problem L belongs to the class RL if there is a probabilistic TM M that uses O(log n) space on inputs of length n and such that
  - For every $x \in L, Pr_r[M(r,x)\ \ accepts] \geq \frac{1}{2}$
  - For every $x \notin L, Pr_r[M(r,x)\ \ accepts] = 0$

Randomized log-space

- Any constant bigger than zero and smaller than one would work.
- Follows that $L \subseteq RL \subseteq NL$

- Even though we now know that L=SL, it is interesting to see the "old" proof of $SL \subseteq RL$.
- **Theorem**. The problem ST-UCONN is in RL.

An alternate characterization of NL

- We saw alternate definition of NP that used certificates instead of non-determinism.
- Can we do the same for NL?
- Certificates might be poly length.
- Need to assume that they are provided to a log-space machine on a read only tape.

An alternate characterization of NL

- **Definition.** A language L is in NL if there exists a deterministic TM M (verifier) with an additional read-once tape, and a polynomial p:$\mathbb{N}$→$\mathbb{N}$ such that for every $x \in \{0,1\}^*$
  $x \in L \Leftrightarrow \exists u \in \{0,1\}^{p(|x|)}$ s.t. M(x,u)=1

- By M(x,u) we denote the output of M where x is placed on the input tape and u on the special read-once tape, and M uses only O(log(|x|)) space on its work tapes for every input x.

An alternate characterization of NL

- What if we remove the read-once restriction and allow the TM’s to move back and forth on the certificate?
- This changes the class from NL to NP (ex).

NL=coNL

- Analogously to coNP, we define coNL to be the class of languages that are complements of NL languages.
- Complement of STCONN is in coNL, denote it by $\overline{STCONN}$: Given directed graph G and special vertices s,t decide whether t is NOT reachable from s.
- In fact, it is coNL-complete.

NL=coNL

- Will show that there is an NL TM which solves $\overline{STCONN}$.
- Generally, for every "well behaved" s(n), NSPACE(s(n))=coNSPACE(s(n)). (ex)

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

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

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

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

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

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

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.