$NL = coNL$

Theorem (Immerman & Szelepcsényi, 1987): $NL = coNL$.

Proof:

- It suffices to show $\overline{\text{Path}} \in NL$.
- I.e. design a logspace-algorithm A s.t. for an input instance (G, s, t), $\exists$ sequence of "guesses" u for A with $A(\langle G, s, t \rangle, u) = 1$ iff $\nexists$ path $s \rightsquigarrow t$ in G.
- Idea: path counting!

- Let $n := |V(G)|$, and C_i be the set of vertices reachable from s in $\leq i$ steps.

$\triangleright$ A logspace-machine can easily certify whether a vertex v is in C_i.

- Now, the plan is to design logspace-machines that could certify:

(i) a vertex $v \notin C_i$, given $|C_i|$, and
(ii) $|C_i| = c$, given c and $|C_{i-1}|$.
(Clearly, this proves $\overline{\text{Path}} \in NL$.)

- Certifying $v \notin C_i$, given $|C_i|$:

The certificate is simply a list of vertices $v_1 < v_2 < \dots < v_{|C_i|}$ in increasing order, all in C_i, and none equal to v.

To do this the algorithm guesses v_j, v_{j+1} & checks: $v_j < v_{j+1}$ & $v_j, v_{j+1} \in C_i$. Finally, it counts that $|C_i|$ many v_j's were guessed.

- Certifying $v \notin C_i$, given $|C_{i-1}|$:

The certificate is again a list $v_1 < \dots < v_{|C_{i-1}|}$ in increasing order, all in C_{i-1}, none equal to v or <u>its neighbour</u>.

Clearly, this certificate is guessable

& checkable by a logspace-machine.

• Certifying $|C_i| = c$, given $|C_{i-1}|$:

For every vertex v, $v \in C_i$ resp. $v \notin C_i$ are certifiable in logspace, given the value $|C_{i-1}|$.

Thus, the machine can go through each vertex v & count the correct value $|C_i|$.

• This finishes the proof of $\overline{\text{Path}} \in NL$. $\square$

Corollary: $\forall S(n) = \Omega(\lg n)$, $\text{Nspace}(S) = \text{coNspace}(S)$.

Proof:

• Let M be a $S(n)$-space NDTM deciding L.

▷ The configuration graph G of $M(x)$ is $S(n)$-space computable (assuming $S(n) > \lg n$ so that x can be read!).

- We have: $x \notin L$ iff $\langle G, C_{start}, C_{accept} \rangle \notin$ Path.

- Now, invoke the previous proof to deduce $\bar{L} \in$ Nspace (S). $\square$

The Polynomial Hierarchy

- The PH is a generalization of NP, coNP & lies well "below" Pspace.

- Consider the following optimization qn.:
  <u>MinDNF</u> $:= \{\phi \mid \phi$ is a DNF formula not equiv. to any smaller DNF $\}$.

- Alternatively, it is:
  $\{\phi \mid \forall \text{DNF } \psi, |\psi| < |\phi|, \exists s, \psi(s) \neq \phi(s)\}$.

- It seems to be beyond NP, coNP as it uses <u>two</u> different quantifiers.

- On the other hand, it does not seem as hard as QBF!

- This motivates a new class:

<u>Defn</u>: • A language $L \in \underline{\Pi_2^P}$ if $\exists$ poly-time TM M & a constant c s.t. $\forall x \in \{0,1\}^*$,

$$x \in L \text{ iff } \forall u \in \{0,1\}^{|x|^c}, \exists v \in \{0,1\}^{|x|^c}, M(x,u,v) = 1.$$

• A language $L \in \underline{\Sigma_2^P}$ if $\exists$ poly-time TM M & a constant c st. $\forall x \in \{0,1\}^*$, $x \in L$ iff $\exists u \in \{0,1\}^{|x|^c}$, $\forall v \in \{0,1\}^{|x|^c}$, $M(x,u,v) = \underline{1}$.

▷ Clearly, $\Sigma_2^P = \text{co-}\Pi_2^P$.

Proposition: (i) $\mathrm{MinDNF} \in \Pi_2^P$.

(ii) $NP \cup coNP \subseteq \Sigma_2^P \cap \Pi_2^P$.

(iii) $\Sigma_2^P \cup \Pi_2^P \subseteq \mathrm{Pspace}$.

- Why stop at <u>two</u> quantifiers !?

- We can define Σ_i & Π_i by alternating $\forall/\exists$ i times :

• $L \in \underline{\Sigma_i}$ if $\exists$ poly-time TM M & a $c>0$ s.t. $\forall x$, $x \in L$ iff

$$\exists u_1 \, \forall u_2 \cdots\cdots Q_i u_i \quad M(x, u_1, \ldots, u_i) = 1 .$$

strings in $\{0,1\}^{|x|^c}$

$Q_i :=$ $\exists$ resp. $\forall$ if i is <u>odd</u> resp. <u>even</u>.

• $\underline{\Pi_i}$ is defined in a similar way except that the quantifier-sequence begins with a '$\forall$'.

• Conventionally, $\underline{\Sigma_0} = \underline{\Pi_0} := P$.

Defn: • The *polynomial hierarchy* is:

$$\underline{PH} := \bigcup_{i \geq 0} \Sigma_i .$$

Proposition: (1) $\Sigma_1 = NP$, $\Pi_1 = coNP$.

(2) $\forall i \geq 0$, $\Sigma_i \subseteq \Sigma_{i+1}$, $\Pi_i \subseteq \Pi_{i+1}$.

(3) $\forall i \geq 0$, $\Pi_i = co\text{-}\Sigma_i$.

(4) $\forall i \geq 0$, $\Sigma_i \cup \Pi_i \subseteq \Sigma_{i+1} \cap \Pi_{i+1}$.

(5) $PH = \bigcup_{i \geq 0} \Pi_i$.

(6) $PH \subseteq Pspace$.

OPEN: We do not know whether it is indeed a hierarchy?

I.e. $\Sigma_0 \subsetneq \Sigma_1 \subsetneq \cdots$?

No class on Thu. Midsem Wed, 19-Feb @ 12:30-14:30 @ L3