Certificate NL

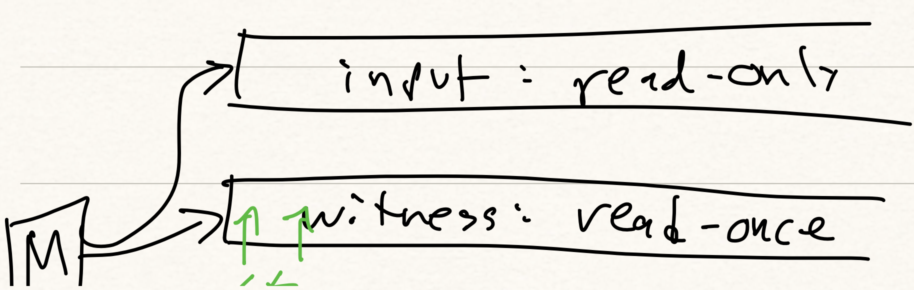
Lecture 11

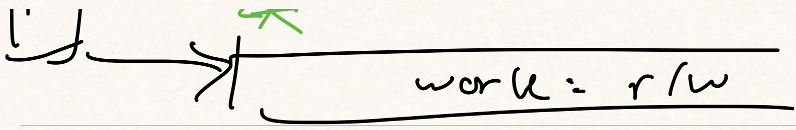
L, NL : input tape not counted for space, only worktape.

Certif. NL :

1. Don't count poly-sized witness/certificate in space usage
2. You can only read the witness <u>once</u>

NL Verifier





Definition: $L \in NL$ if $\exists$ DTM M w/ a special read-once witness tape & polynomial p s.t.

$\forall x \in \{0,1\}^*$

$$x \in L \iff \exists w \in \{0,1\}^{p(|x|)} \quad M(\underbrace{x}_{\text{read-only tape}}, \underbrace{w}_{\text{read-once tape}}) = 1$$

and M uses $O(\log |x|)$ space on work tape.

NL = coNL

- Recall: we don't believe NP = coNP

NP: True for atleast 1 certificate

coNP: True $\forall$ certificates

$\text{coNL} := \{ L : \overline{L} \in \text{NL} \}$

- <u>Know</u>: $\overline{\text{PATH}}$ is coNL-complete

$\text{PATH} := \{ (G, s, t) : \exists$ path from s to t in directed graph G $\}$

$\overline{\text{PATH}} := \{ (G, s, t) :$ no paths from s to t. $\}$

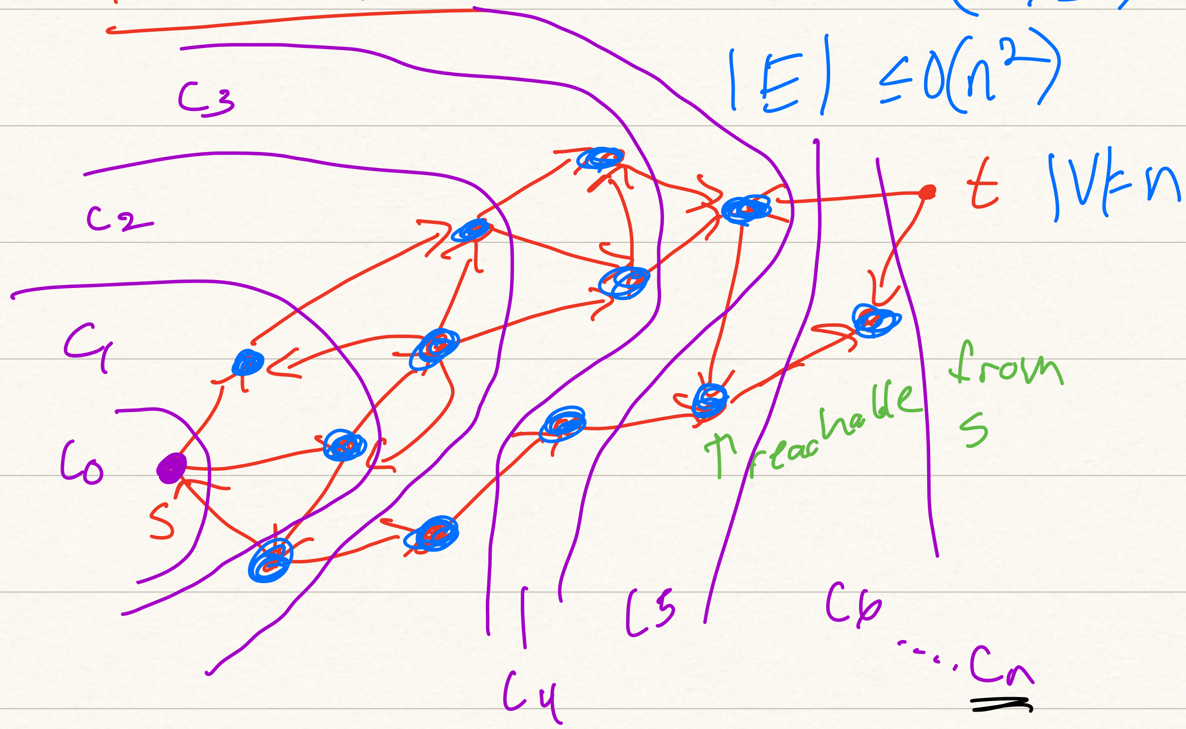
<u>Theorem</u>: $\overline{\text{PATH}} \in \text{NL}$

<u>Proof</u>: We'll give a certificate certifying no path from

s to t.

Main Idea If $G=(V,E)$

$|E| \leq O(n^2)$



we will construct set of vertices reachable from s in log space.

- First, we say $v \in V$ is reachable from s if $\exists$

path from s to v.

- Let $C_i \subset V$ be the set of $v \in V$ s.t. v is reachable from s in at most i steps.
  - Notice: $C_{i-1} \subseteq C_i$ $\forall i \geq 1$
  - * If $|V| = n$, then $V = \{1, \ldots, n\}$
- Let $i \in [n]$ and let $v \in C_i$.

P1 We give a read-once certificate showing "$u \in C_i$"

- Certificate $(v_0, \underset{\substack{\uparrow \\ v_i \in V(G)}}{v_1}, \ldots, v_k)$ len-k path

① $v_0 = s$ (log-space check equality)

② $\forall j$: check $(v_{j-1}, v_j) \in E(G)$

π π input,

check in log space
just comparing $\log(n)$-bit strings.

not counted towards space usage

③ $\forall_j$: check if $u = v_j$

④ Count up to k, check $k \leq i$

- Given P1, we'll build 2 more complicated certificates.

D1 Certificate for u & C_i given we know $|C_i| = s_i$.

↳ has been verified

D2 A certificate that $s_i = |C_i|$ given we know $s_{i-1} = |C_{i-1}|$

• **Fact**: $|C_0| = 1$ $\quad C_0 = \{s\}$

(D1) **Certificate**: certify $v \notin C_i$ given $s_i = |C_i|$.

- (P1) gives us a certificate for $u \in C_i$.
  - Let $w_{u,i}$ be this certificate
- Make new certificate

$$\overline{w}_{v,i} = (w_{u_1,i}, w_{u_2,i}, \ldots, w_{u_{s_i},i})$$

$v \notin C_i$

$$\{u_1, u_2, \ldots, u_{s_i}\} = C_i$$

$$u_i < u_{i+1} \quad \forall i \in [s_i - 1]$$

Checking $\overline{w}_{v,i}$:

① Verify all $w_{u_j,i}$ are valid

② Check $u_j < u_{j+1}$ $\forall j$

③ Check $v \neq u_j$ $\forall j$

④ Exactly $\boxed{|s_i|}$ certificates are given

We assume we have this.

D1.5 Certify $v \in C_i$ given $s_{i-1} = |C_{i-1}|$.

- Certificate is exactly the same but for C_{i-1}

$$\overline{w}_{v,i} = (w_{u_1,i-1}, w_{u_2,i-1}, \ldots, w_{u_{s_{i-1}},i-1})$$

Verification is nearly the same

1. Verify all $w_{u_j, i-1}$ are valid
2. Check $u_j < u_{j+1}$ $\forall j$ ✓
3. Check $v \neq u_j$ and $\forall j$ $v \notin \text{neighbors}(u_j)$
4. Exactly $\boxed{s_{i-1}}$ certificates are given

We assume we have this.

D2: Certify $s_i = |C_i|$ given $s_{i-1} = |C_{i-1}|$

- We have seen Certificate for

① $u \in C_i \quad \forall u$

② $v \notin C_i \quad \forall v$

- Given $s_{i-1} = |C_{i-1}|$ + above, build a certificate for $s_i = |C_i|$
- Construct $\hat{w}_i$
  - n certificates, one for every vertex

$$\hat{w}_i = (\alpha_{1,i}, \alpha_{2,i}, \dots, \alpha_{n,i})$$

$$\alpha_{j,i} = \begin{cases} w_{j,i} & \text{if } j \in C_i \\ \overline{w}_{j,i} & \text{if } j \notin C_i \end{cases}$$

- Certification:

① Check that # of $w_{j,i}$ is s_i

② Use D1.5 to certify $\bar{w}_{j,i}$

↳ we are only given $s_{i-1} = |C_{i-1}|$

Final Certificate

$w = (\hat{w}_1, \hat{w}_2, \ldots, \boxed{\hat{w}_n}, \bar{w}_{t,n})$

certify $s_2 = |C_2|$ given $s_1 = |C_1|$

certify $s_1 = |C_1|$ given $s_0 = |C_0| = 1$

$\widehat{W}_n$: $s_n = |C_n|$

$\overline{W}_{t,n}$: $t \notin C_n$ given

$s_n = |C_n|$ $\square$

- **Corollary** : $\forall$ space-constructible S

$$NSPACE(S) = coNSPACE(S)$$

The Polynomial Hierarchy

- Sometimes, non-determinism not strong enough

- **Example** : INDSET =

$\{ (G, k) : G \text{ has indset of size} \geq k \}$

Exact-INDSET :=

$\{ (G, k) : \text{largest indset in } G \text{ has size } k \}$

$\underbrace{(G, k)}_{x} \in \text{EIND} \iff$

$\exists$ indset of siz k s.t.

$\forall$ other indsets have size $\leq k$.

$\iff \exists w_1 \forall w_2 \; M(x, w_1, w_2) = 1$

The class Σ_2^p:

$L \in \Sigma_2^p$ if $\exists$ DTM M (the verifier) and poly q s.t. $\forall x \in \{0,1\}^*$

$$x \in L \leftrightarrow \exists w_1 \in \{0,1\}^{q(|x|)} \; \forall w_2 \in \{0,1\}^{q(|x|)} \; M(x, w_1, w_2) = 1$$

Ex: MIN-EQ-DNF ::= $\{(\varphi, k) :$ $\exists$ DNF ψ of size $\leq k$ that is equivalent to $\varphi\}$

reverse CNF

OR of ANDs

$\exists$ DNF ψ of size $\leq k$ s.t.

$\forall u \quad \varphi(u) = \psi(u)$

<u>In fact</u>:

$$NP \subset \Sigma_2^P$$
$$coNP \subset \Sigma_2^P$$