$\mathbb{NL}$

PATH is $\mathbb{NL}$-complete.

$\{\langle G, s, t\rangle \mid$ G is a directed graph and there is a path from s to t $\}$

Can we define $\mathbb{NL}$ in terms of certificates?
($\mathbb{NP}$ can be defined in terms of polytime verifiable certificates.)

<u>Naive (incorrect) defn:</u> (certificate viewpoint)

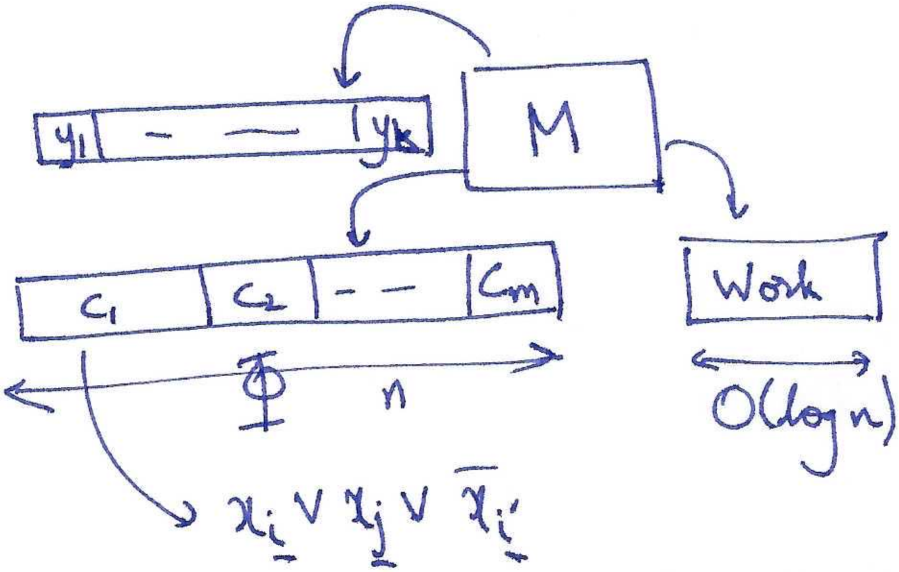
$\mathcal{L} \in \cancel{\mathbb{NL}}$ $\mathbb{NP}$ if $\exists$ polynomial p & logspace ~~poly time~~ machine M
s.t. $\cancel{\forall x \in \mathcal{L}}$ $\forall\, x \in \{0,1\}^*$

$$x \in \mathcal{L} \iff \exists\, y,\ |y| \le p(|x|) \text{ s.t. } M(\langle x, y\rangle) \text{ accepts.}$$

This defn. is incorrect!

In such a setting, we can decide 3SAT!

Given 3CNF Φ and a possible assignment $y \in \{0,1\}^*$, we can check if $\Phi(y) = 1$ in logspace.

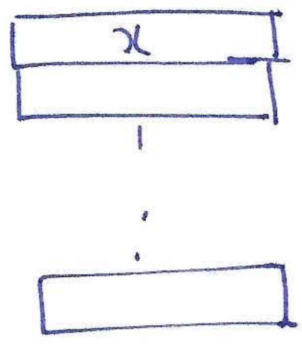


Certificate say: set x_i to bit y_i

For each C_i, machine has to look value of literals in that clause, and check if C_i is satisfied. Machine only need logspace to look up these values.

By carefully looking Cook-Levin theorem / reduction, one can show $\forall\, \mathcal{L} \in \mathbb{NP}$, $\mathcal{L} \leq_L \mathrm{SAT}$

<u>Thm</u>: $\mathcal{L} \in \mathbb{NP}$ iff $\exists$ poly p and logspace machine M s.t. $\forall\, x$, $x \in \mathcal{L} \iff \exists\, y, |y| \leq p(|x|)$, s.t. $M(\langle x, y\rangle)$ accepts.



Φ $\quad z_{ti\alpha} \rightarrow$ Indicator that ~~at~~ time t, i^{th} symbol on tape is α.

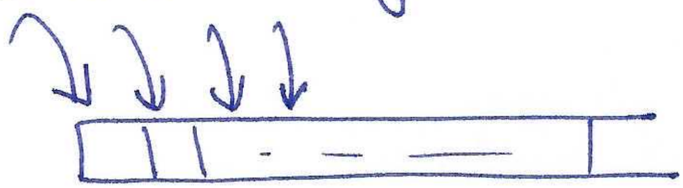
$\Phi = \mathrm{UNIQUE} \wedge \mathrm{START} \wedge \mathrm{ACCEPT} \wedge \mathrm{MOVE}$

Individual clauses in Φ only depend on a constant number of variables.

Only the starting configuration depends on the input.

The right definition

Define a tape to be READ-ONCE if head (on the tape) cannot write on the tape and head only moves right.



Thm

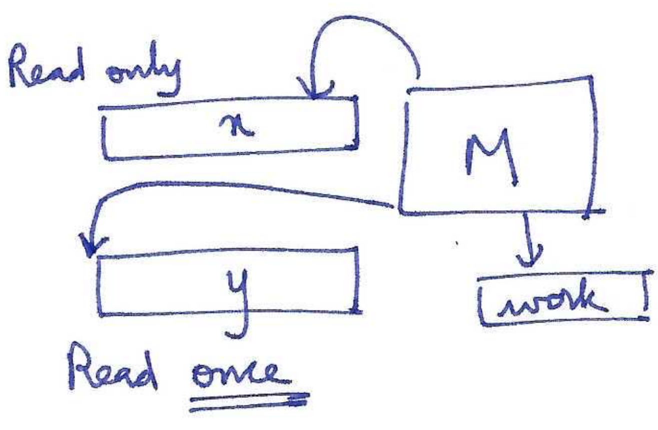
$\mathcal{L} \in \mathbb{NL}$ iff $\exists$ poly p and logspace machine M s.t.

$\forall x \quad x \in \mathcal{L} \iff \exists y, |y| \leq p(|x|)$ st

$M(\langle x, y \rangle)$ accepts and

x is on input tape and

y is on a read-once tape.



Read-once certificate

M cannot store certificate in ~~work~~ workspace.

Logspace verifier
Standard certificate $\Rightarrow \mathbb{NP}$

Read-once certificates
$\equiv$ Non-determinism

Poly time verifier
Standard certificate $\Rightarrow \mathbb{NP}$

Cert.	Verifier: Logspace	Verifier: Poly time
Read-once cert.	$\mathbb{NL}$	$\mathbb{NP}$
Standard cert.	$\mathbb{NP}$	$\mathbb{NP}$

$\mathbb{NP} \longrightarrow$ Cert can be stored in workspace. Now, cert. becomes a standard cert.

<u>Proof.</u> Suppose $L \in \mathbb{NL}$

There is a NTM M running in logspace that
(⇐) decides L. To get the verifier/certificate viewpoint, let certificate be the non-deterministic choices of a run of M.

The verifier M' simply runs M using the non-deterministic choices in the certificate. (M never needs to look back at previous non-deterministic choices.) Certificate is poly(n) sized because $\mathbb{L} \subseteq \mathbb{P}$.

($\Rightarrow$) Consider L with a logspace verifier M' and read-once certificate. Create a NTM M that simulates M' and uses non-determinism to guess the next symbol of certificate. M' decides L. ∎

The Immerman-Szelepcsényi Theorem

$$\mathbb{NL} = \text{co-}\mathbb{NL}$$

Generally, $\mathbb{NSPACE}(s(n)) = \cancel{\mathbb{SP}}\ \text{co-}\mathbb{NSPACE}(s(n))$

$\forall$ ~~s(n) ≥ log n~~ $s(n) = \Omega(\log n)$ (Padding argument)

PATH is $\mathbb{NL}$-complete.

$\overline{\text{PATH}}$ is co-$\mathbb{NL}$-complete.

If we show $\overline{\text{PATH}} \in \mathbb{NL}$, then we prove $\mathbb{NL} = \text{co-}\mathbb{NL}$. (Exercise)

$\overline{\text{PATH}} = \{ \langle G, s, t \rangle \mid$ G is directed and there is NO path from s to t$\}$

PATH $\in \mathbb{NL}$ is "obvious", non-determinism/read-once cert. is the path.

How can non-determinism prove that there is NO path?

Thm: $\overline{\mathrm{PATH}} \in \mathbb{NL}$

Proof: Think in terms of (poly-sized) read-once certificates.
You'll go nuts thinking of a non-deterministic logspace machine.

Define $\overline{\mathrm{PATH}_i} = \{ \langle G,s,t \rangle \mid$ There is no path from s to t of length $\leq i \}$

Let $d(s,t)$ be shortest path distance from s to t

$\overline{\mathrm{PATH}_i} = \{ \langle G,s,t \rangle \mid d(s,t) > i \}$ (dist is ∞ if there is no path)

$\overline{\mathrm{PATH}_{n-1}} = \overline{\mathrm{PATH}}$

We will use certificates (read-once) for $\overline{\mathrm{PATH}_i}$ to construct read-once certificates for $\overline{\mathrm{PATH}_{i+1}}$.

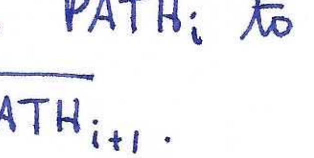
Iterative Counting (G is adjacency matrix)

$\overline{\mathrm{PATH}_0} = \{ \langle G,s,t \rangle \mid s \neq t \} \in \mathbb{L}$

$\overline{\mathrm{PATH}_1} = \{ \langle G,s,t \rangle \mid s \neq t, (s,t) \notin E \}$ $(G = (V,E))$

$\in \mathbb{L}$ $\to$ single lookup in G.

$\overline{\mathrm{PATH}_2} = \{ \langle G,s,t \rangle \mid d(s,t) > 2 \}$



$\Gamma^+(v)$ = outneighborhood

$\Gamma^-(v)$ = in-neighborhood

$\to \Gamma^+(s) \cap \Gamma^-(t) = \phi$

$d(s,t) > 2$ iff $\forall\, v \in \Gamma^+(s)$, $(v,t) \notin E$

checked in logspace

First try: certificate is $\Gamma^+(s)$ is $v_1, v_2, \ldots, v_k$

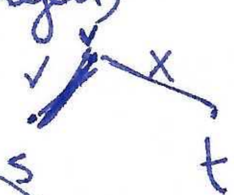
(0) Check $(s,t) \notin E$

Verifier: (1) $\boxed{\text{For } v \text{ in } v_1, \ldots, v_k}$ Read once

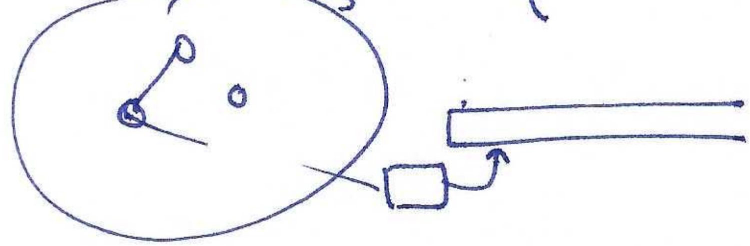
(a) Check if $(s,v) \in E$. (Else reject)

(b) Check if $(v,t) \notin E$. (Else reject)

(2) Accept



What if there is some neighbor v' of s that is NOT in certificate?



The verifier can loop of over all neighbors of s, but it cannot go back and check if they lie in certificate.

~~Cert~~ Certificate is

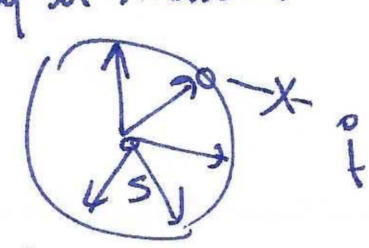
$\overset{\log n}{\longleftrightarrow}$				
d_v^+	v_1	v_2	--	v_k

$v_1 < v_2 < \cdots < v_k$ (prevent repeats)

Verifier: (1) Store d_v^+ from certificate in ~~workspa~~ workspace.

(2) Compute outdegree of v, check if it matches cert. (If not, reject)

(3) For v in $v_1 < v_2 < \cdots < v_k$



(4) Check that $k = d_v^+$. If not, reject