Log Space TM
- read-only input tape (input of length n)
- work tape, need \( O(\log n) \) space

Log space transducer is a TM with
- read-only input tape, read-once
- write-only, move-right only output tape
- work tape, need \( O(\log n) \) space
- and always terminates

Example: \( 0^n \) 1 \( \) can be decided by a log space TM, since we're just counting, and that takes log space.

The complexity class \( L = \{ L \mid L=TC^0(\log n) \} \) for log space TM \( M \).

\( NL = \{ L \mid L=TC^0(\log n) \} \) for a logspace nondeterministic TM \( N \).

Open problem: Is \( L = NL \)? - Savitch's theorem only shows \( NL \) is \( O(\log^2 n) \).

\( PATH = \{ \langle G, s, t \rangle \mid G \text{ is a directed graph with a path from } s \in V(G) \text{ to } t \in V(G) \} \).

\( PATH \in NL \): \( N = \) On input \( \langle G, s, t \rangle \),
- Doesn't work! \( \langle \text{Non-det. pick a sequence of vertices } v_1, \ldots, v_n \rangle \).
- Could be linear. \( \langle \text{Verify } v_1 = s, v_n = t, \text{ and } (v_{i-1}, v_i) \in E(G) \rangle \).

New attempt: check one by one:
- \( N = \) On input \( \langle G, s, t \rangle \),
  - Let \( v_1 = s \).
  - For \( i = 2 \) to \( n \),
    - Non-det. pick \( v_i \).
    - Check that \( (v_{i-1}, v_i) \in E(G) \).
    - If \( v_n = t \), accept.
    - Erase \( v_{i-1} \).
  - Reject.
PATH $\in$ coNL:
Main idea: Given $<G, s, t>$ and $C$, guess a promise that $C =$ # vertices reachable from $s$.
Guess the $C$ vertices and verify they're reachable from $s$.
If not, reject.
Non-det, guess $v_1, \ldots, v_C$; Reject if $t \notin \{v_1, \ldots, v_C\}$.
Run $N$ from before on $<G, s, v_i>$ to make sure
the $v_i$'s are connected.
If so, accept, otherwise reject.

Where does $C$ come from? We can guess it non-deterministically, since
then we'd miss vertices.
Let $C_i =$ # vertices reachable from $s$ by a path of length $i$.
So, $C = C_n$. $C_0 = 1$, $C_2 =$ out-degree of $s$,
For each $v \in V(G)$,
For $C_i$, we use $C_{i-1}$. Non-det, one-to-one, guess which vertices
are at distance $i-1$ from $s$.
* For each of them, verify correctness of your guess. (incorrect)
* For each of them, if there is an edge into $v$ or $v$ is
one of them, increment $C_i$ (start at $C_i = 0$).

Thm: NL $\subseteq$ coNL

A log space reduction is computable by a log-space transducer.
We say $A \leq_L B$ if there is a log space transducer $T$ s.t.
$x \in A \iff T(x) \in B$. ($A$ is log space reducible to $B$).

Thm: If $A \leq_L B$ and $B \in L$, then $A \in L$.

We choose the definition of logspace reduction so this theorem works.
NL $\subseteq$ P, so $A \leq_L B$ for $A, B \in NL$, and
polynomial reducibility isn't helpful.
Why do we want a transducer?

At PATH = \{ \langle G \rangle \mid G \text{ is a directed graph with a path from } 1 \text{ to } n \}.

If there's any justice, logspace reducibility should be s.t.

At PATH \subseteq PATH.

Logspace TM can only write O(log n) bits on output tape, but reduction wants to write \langle G, 1, n \rangle.

A logspace transducer can output \langle G, 1, n \rangle, since write-only output tape can be any length.

pt of A \leq B \# and B \in L \implies A \in L:

Want to construct M_A, logspace TM for A, from M_B, logspace TM for B, and T, logspace transducer s.t. x \in A \iff T(x) \in B.

TRY 1: On input x,

- Compute T(x) = y
- Run M_B on input y

Doesn't work, since y could be very long: too much space!

TRY 2: On input x,

- Compute y_1, (first symbol of y)
- By running T until we get y_1
- Involve M_B, give it y_1
- If M_B points at y_i and wants to move left, run T on x, throw away y_1, y_2, and get y_{i+1}, then simulate next step of M_B.
- If M_B wants to move right, run T(x), ignoring y_1, ..., y_i, get y_{i+1}, then simulate next step of M_B.

Thm: If A \leq B and B \leq C, then A \leq C.

Def: A is NL-hard if \forall L \in NL, L \leq A.

Def: A is NL-complete if A is NL-hard and A \in NL.
PATH is NL-hard. Let $N$ be any logspace NTM

WT's, $L(N) \subseteq PATH$

Idea:

$N$ has a path from $s$ to $t$ iff $N$ accepts $w$

$
< G, s, t >
$

Vertices of $G$ are configurations of $N$ on input $w$.

There are poly $|w|$ of them.

Edges of $G$: $(u, v) \in E(G)$ if config $v$ is reachable

from config $u$ using a transition of $N$.

$s =$ config, corresponding to start

$t =$ an accepting config. (w.l.o.g., $N$ erases contents

of work tape before accepting, so $t$ is unique)

Proof that this is a logspace reduction: exercise.

Randomized computation:

Randomized TM: extra read-only, non-right/stay-only

infinite tape, pre-filled with random bits

the usual input/work tape.

Example: Input: Boolean formula $\phi$

promise: either $\frac{2}{3}$ of assignments to $x_1, \ldots, x_n$ satisfy $\phi$

or $\leq \frac{1}{3}$ do:

Pr[$M$ accepts $w$] = \# of distinct accepting random tapes of length $\ell(w)$

\[
\frac{1}{2 \ell(w)}
\]

Runtime of $M$ is $\ell(w)$