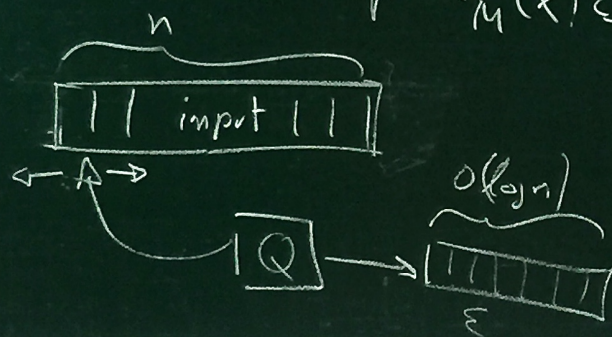


15 NC¹, L, NL, and AC¹

NC¹, L, NL, co-NL, AC¹

$L := \{L \subseteq \Sigma^* \mid \exists \text{DTM } M$
 $M(x) = X_c(x) \text{ and}$
 $\text{space}_M(x) \in O(\log n)\}$



- DTM can only move within the input on the input tape
- cannot write on the input
- can move left & right

$$\begin{aligned}
 \# \text{conf} &\in n \cdot |Q| \cdot |\Sigma|^{c \cdot \log n} \\
 &= O(n^c)
 \end{aligned}$$

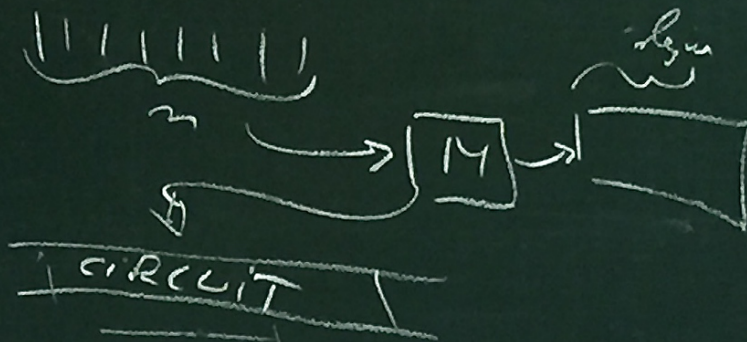
15 NC¹, L, NL, and AC¹

NC^1 depth: $O(\log n)$
 size: $O(n^c)$

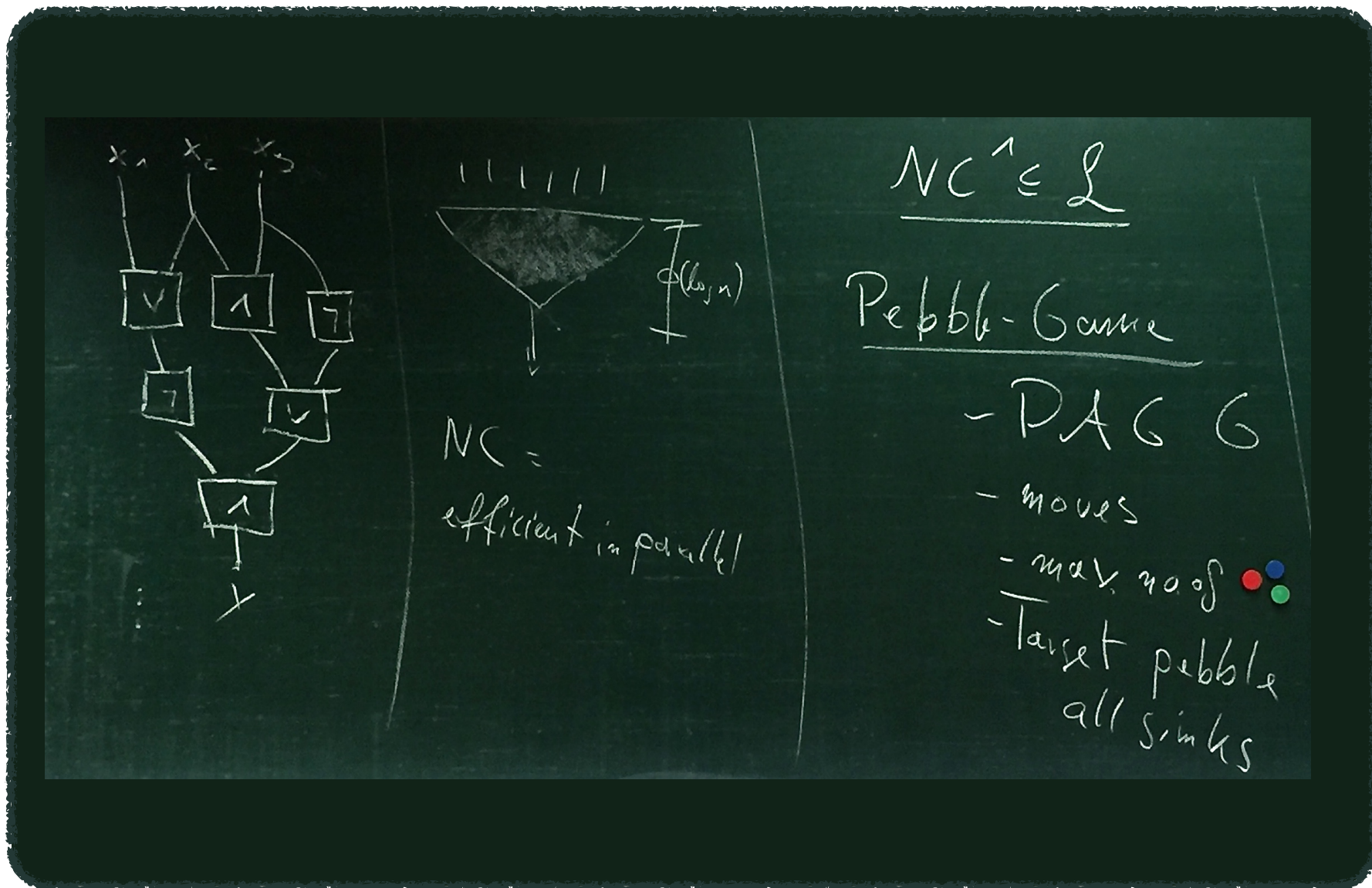
- Boolean fan-in bounded circuit.

Base $\Omega = \{\text{and, or, not}\}$

- log space constructible



15 NC¹, L, NL, and AC¹



x_1 x_2 x_3

\vee \wedge \neg

\neg \vee

\wedge

x

$\log_2 n$

NC = efficient in parallel

$NC^1 \in \mathcal{L}$

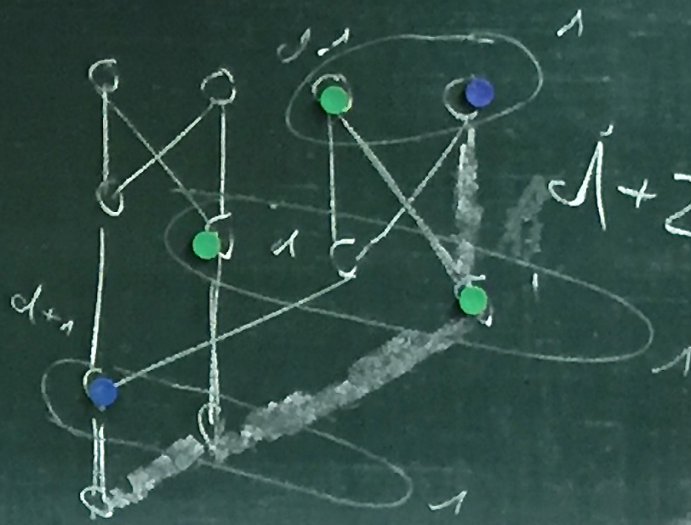
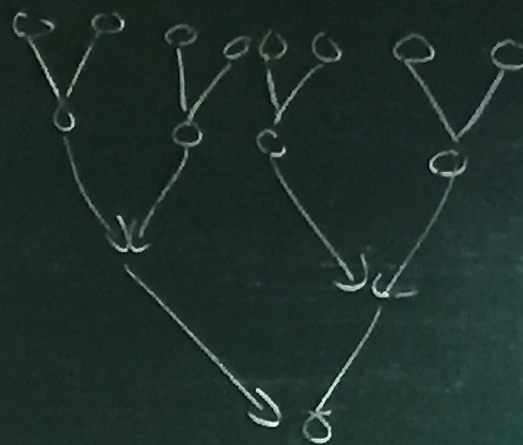
Pebble-Game

- DAG G
- moves
- max. no. of ● ● ●
- Target pebble all sinks

15 NC¹, L, NL, and AC¹

Moves

1. You can put a pebble on a source
2. You can remove a pebble
3. if all direct predecessors of a node have a pebble you can pebble the node.



Lemma Every DAG G
 with fanin 2 can be
 pebbled with $d+2$
 pebbles.

Proof

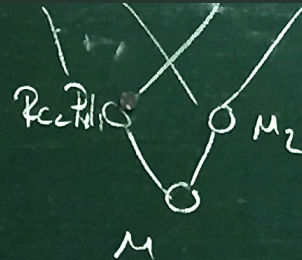
RecPebble(u)

if u is leaf then pebble u

else RecPebble(u_1); u_1 is left parent.

Remove all pebbles used the except pebble in

RecPebble(u_2); Remove pebbles too
 Pebble u .

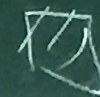


By induct. on
 the claim follows

Log space machine M

- Use RecPebbl on the circuits
- Every time we put a pebble on a node we can compute the value.
- Encode the recursion path binary.

l r l l r l

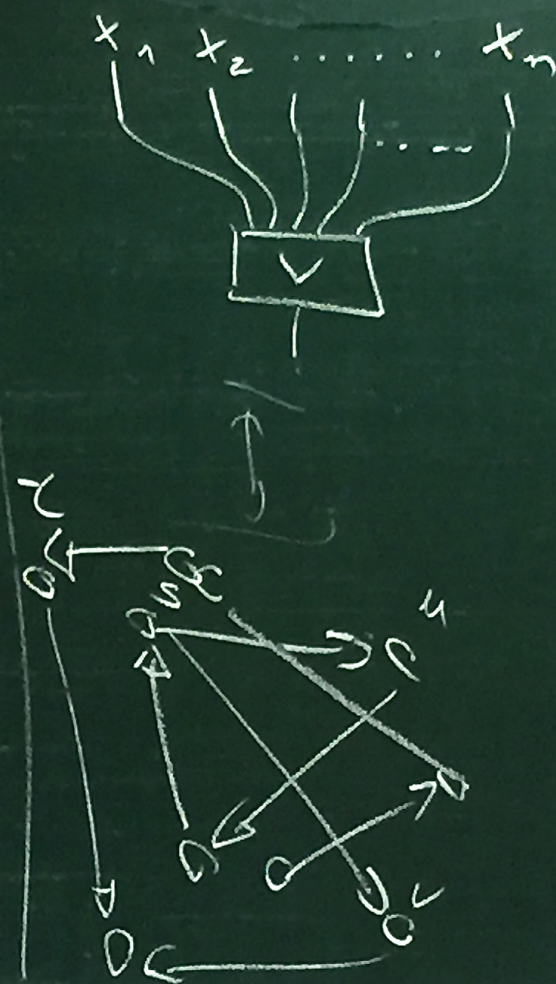


15 NC¹, L, NL, and AC¹

$$L \subseteq NL \subseteq AC^1$$

AC¹

- logspace const. circuits
- not fan in bounded
- Base $\{ \text{and, or, not} \}$
- depth $O(\log n)$
- size $O(n^c)$



$PATH = \{(G, s, t) \mid \text{dir. graph } G, s, t \in V(G)\}$

 1} path from s to t }

Theorem

PATH \in NL-complete

1. PATH \in NL? ✓

$(u, v) / (v, u) (u, w) \dots (s, u)$

1. guess the path iteratively and store only the two current nodes
2. stop if more than n guesses.

15 NC^1 , L, NL, and AC^1

$PATH \in NL$ - hard $AC^0 \subseteq TC^0 \subseteq NC^1 \subseteq L \subseteq NL \subseteq AC^1 \subseteq TC^1 \subseteq NC^2$

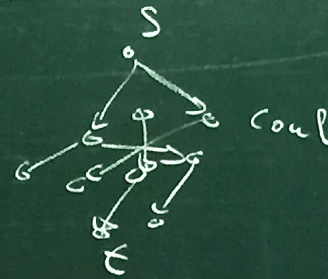
$\forall L \in NL. L \leq_{log} PATH$

L has a logspace bounded NTM: M

- initial configuration s

- accepting configuration t

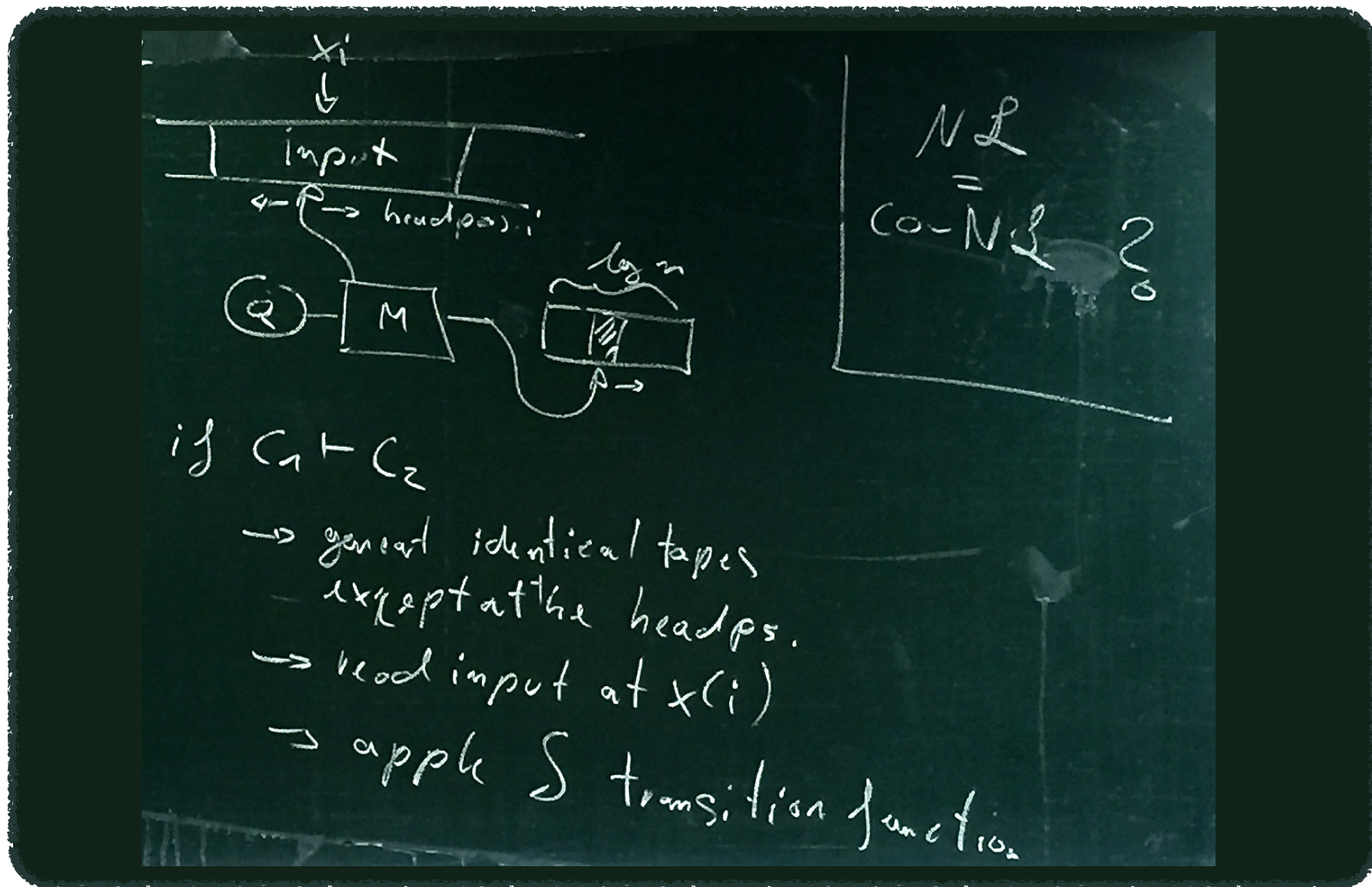
- transition $C_1 \vdash C_2, E \Rightarrow (C_1, C_2)$



\exists path in G from s to t?

How much space do we need to compute (C_1, C_2)
 $O(\log n)$

15 NC^1 , L, NL, and AC^1



x_i
 ↓
 input
 ← head pos. i →
 Q → M → $\log n$
 A →

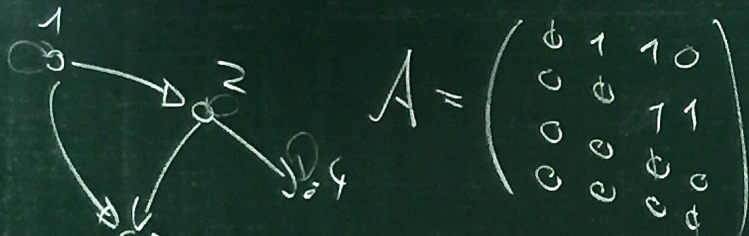
$NL = \overline{Co-NL} \approx \Sigma_0^1$

if $C_1 \vdash C_2$

- general identical tapes except at the head pos.
- head input at $x(i)$
- apply δ transition function

$$NL \subseteq AC^1$$

Proof PATH $\in AC^1$



$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$a_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{if } (i,j) \notin E \end{cases}$$

$$A^k = \bigvee_{k \in V} a_{ik} \wedge a_{kj} = \begin{cases} 1 & \exists \text{ path of length } k \\ 0 & \text{else} \end{cases}$$

$$A^2 = A \cdot A$$

$$(A \vee I)^k = \begin{cases} 1 & \exists \text{ path of length at most } k \\ 0 & \text{elsewhere} \end{cases}$$

$$I = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

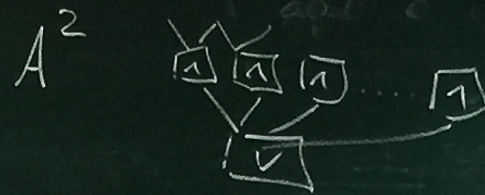
15 NC¹, L, NL, and AC¹

$$A^2 = \sum_{k \in V} a_{ik} \cdot a_{kj} = \begin{cases} 1 & \exists \text{ path of length 2} \\ 0 & \text{else} \end{cases}$$

$$A^2 = A \cdot A$$

$$(A \vee I)^k = \begin{cases} 1 & \exists \text{ path of length at most } k \\ 0 & \text{elsewhere} \end{cases}$$

$$I = \begin{pmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{pmatrix}$$



$$(A \vee I)^n = \left(\underbrace{\left(\left(\left(A \vee I \right)^2 \right)^2 \right)^2 \right)^2}_{x = \log_2 n}$$

$$x = \log_2 n$$

\Rightarrow Transition closure $\in AC^1$

\Rightarrow PATH $\in AC^1$