CMSC 28100

Introduction to Complexity Theory

Autumn 2025

Instructor: William Hoza

~~Which problems~~

~~can be solved~~

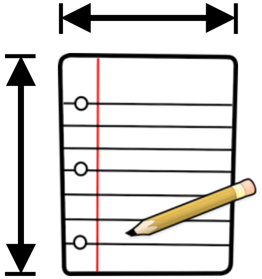
~~through computation?~~

Complexity theory:

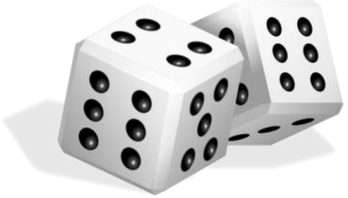
The study of computational resources

Computational resources: Fuel for algorithms

TIME

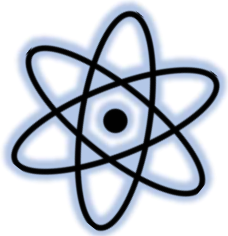


SPACE



RANDOMNESS

ADVICE



QUANTUM PHYSICS

COMMUNICATION

P vs. NP vs. PSPACE vs. EXP

- $\mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE} \subseteq \mathrm{EXP}$
- What we expect: All of these containments are strict
- What we can prove: At least one of these containments is strict. (Why?)

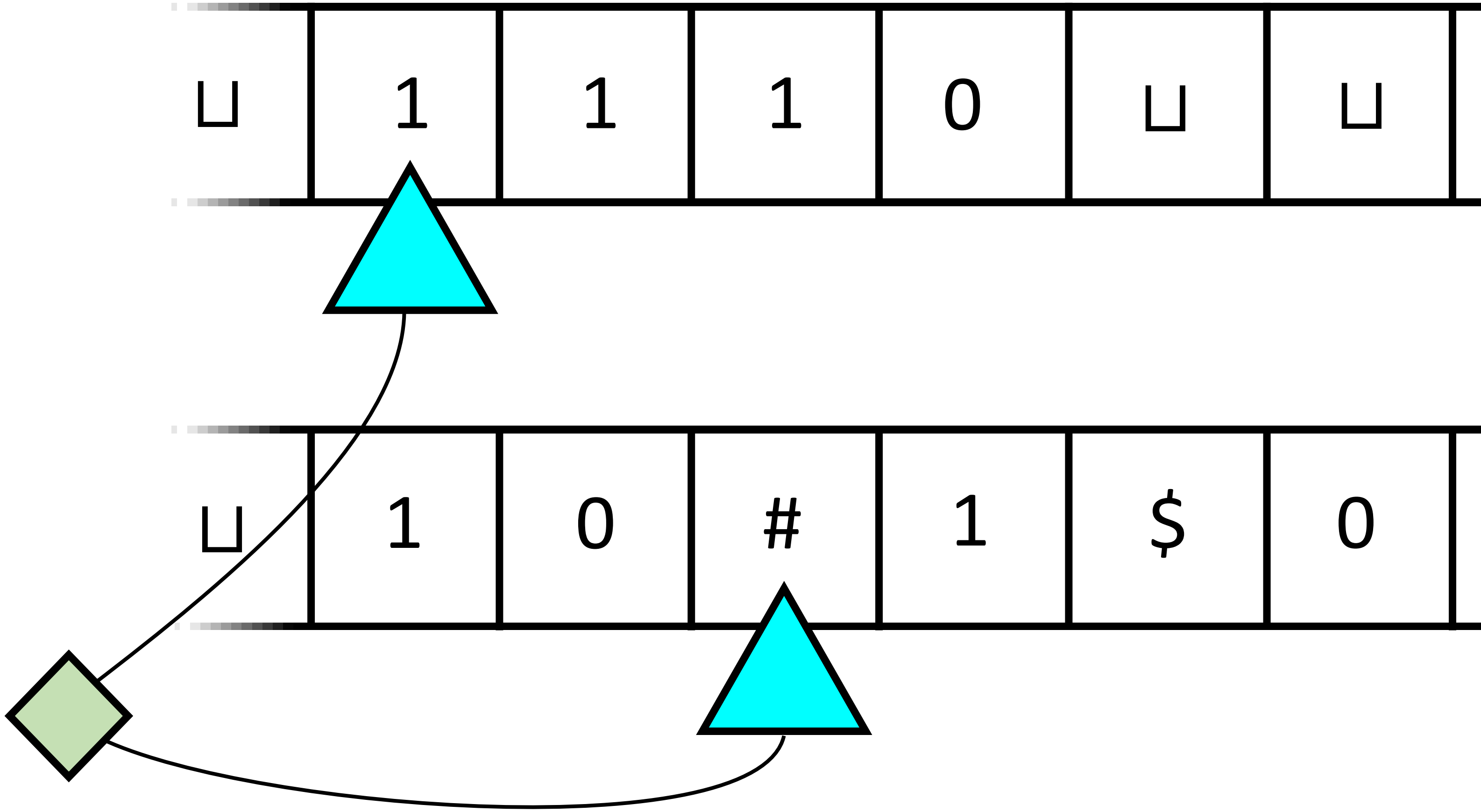
Sublinear-space computation

- Can we solve any interesting problems using $o(n)$ space?
- The one-tape Turing machine is the not the right model of computation for studying sublinear-space algorithms

Sublinear-space computation

Read-only input tape ➞

Read-write work tape ➞

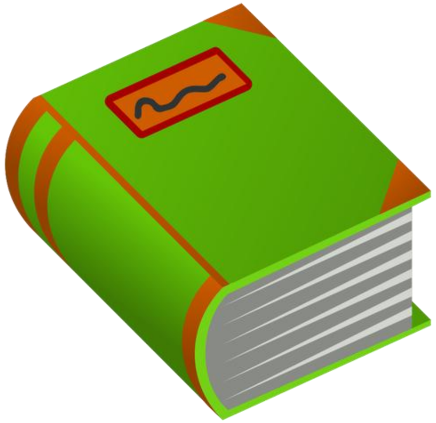


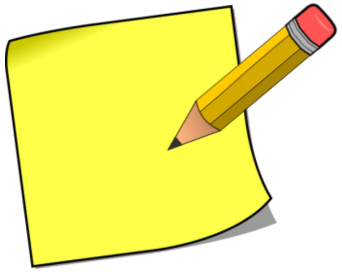
The complexity class $\mathrm{SPACE}(S)$

- Let $Y \subseteq \{0,1\}^*$ and let $S: \mathbb{N} \to \mathbb{N}$ be a function (space bound)
- **Definition:** $Y \in \mathrm{SPACE}(S)$ if there is a two-tape Turing machine M such that:
  - M decides Y
  - M never modifies the symbols written on tape 1
  - Tape 1 head is always located within one cell of the input
  - When the input has length n, the tape 2 head visits $O\big(S(n)\big)$ cells

The complexity class $\mathbf{L}$

- Exercise: $\mathrm{PSPACE} = \bigcup_k \mathrm{SPACE}(n^k)$
- **Definition:** $\mathrm{L} = \mathrm{SPACE}(\log n)$
- L is the set of languages that can be decided in logarithmic space





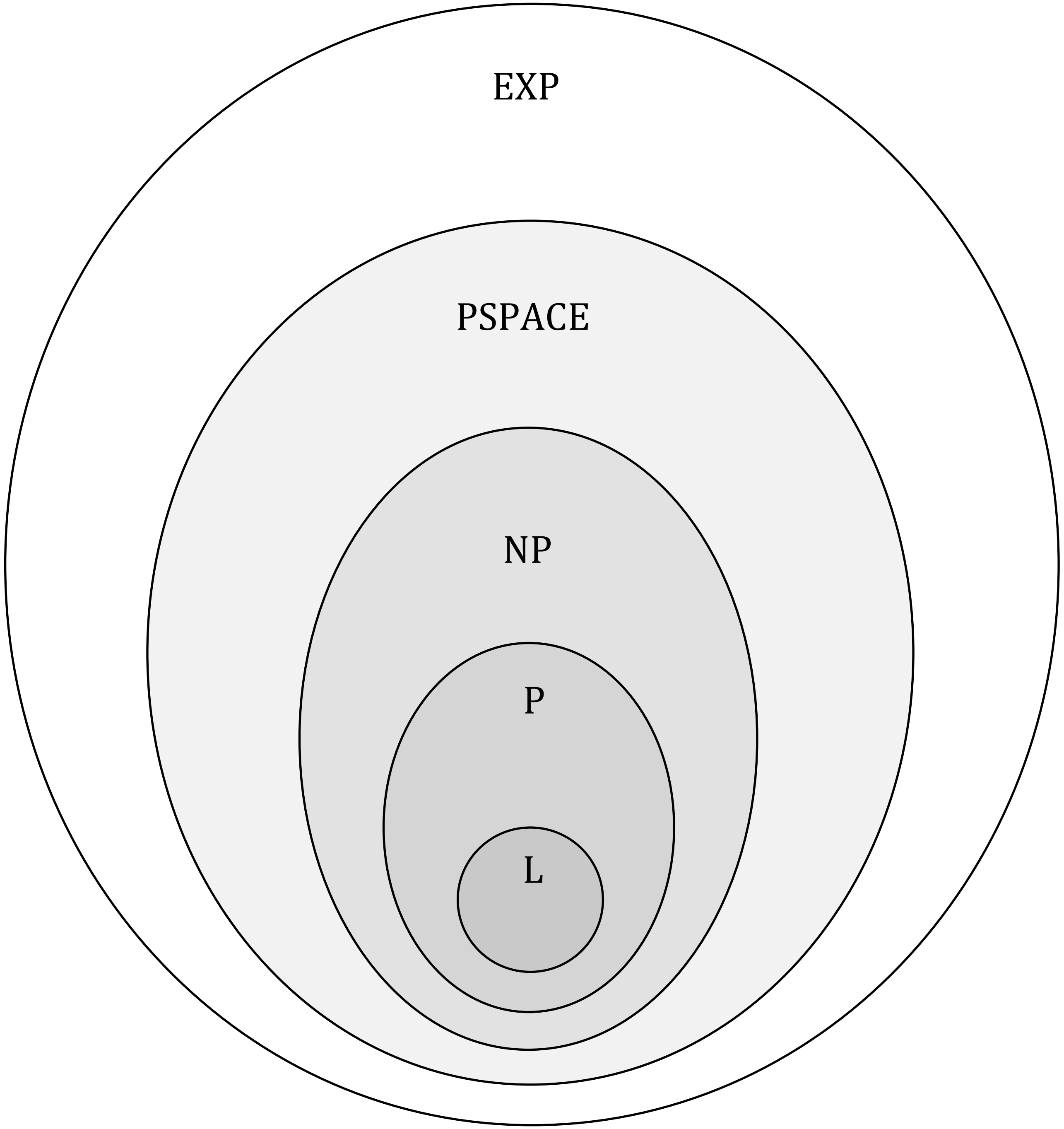
Example: Balanced binary strings

- $\text{BALANCED} = \{x \in \{0,1\}^* : x \text{ has equal numbers of zeroes and ones}\}$
- **Claim:** $\text{BALANCED} \in \text{L}$
- **Proof sketch:** Given $x \in \{0,1\}^n$:
  - Count the number of ones in x
  - Count the number of zeroes in x
  - Check whether the two counts are equal

These counters are only $\log n$ bits each!

$\mathrm{L} \subseteq \mathrm{P}$

- Exercise: Show that $\mathrm{L} \subseteq \mathrm{P}$
- (Similar to the proof that $\mathrm{PSPACE} \subseteq \mathrm{EXP}$)


EXP
PSPACE
NP
P
L

The $\mathbf{L}$ vs. $\mathbf{P}$ problem

- We expect that $\mathrm{L} \neq \mathrm{P}$, but we don't know how to prove it
- $\mathrm{L} = \mathrm{P}$ would mean that every efficient algorithm can be modified so that it only uses a tiny amount of work space

L vs. P vs. NP vs. PSPACE

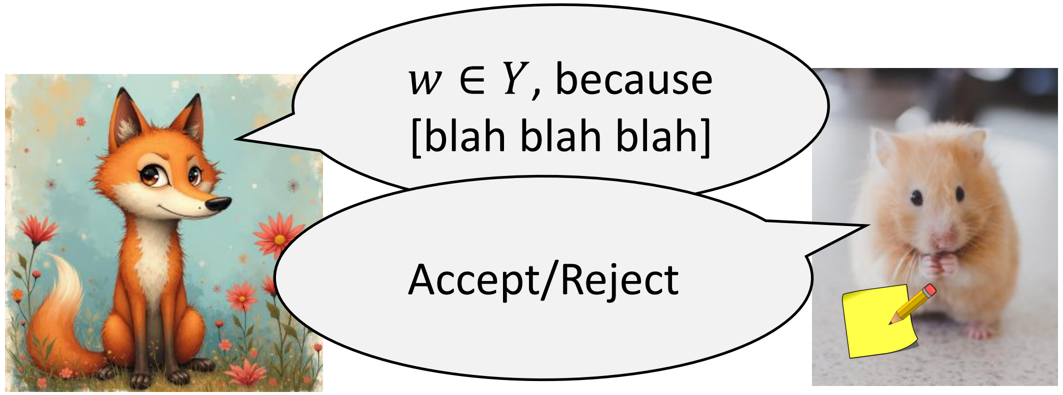
- $\mathrm{L} \subseteq \mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE}$
- What we expect: All of these containments are strict
- What we can prove: At least one of these containments is strict:

Theorem: $\mathrm{L} \neq \mathrm{PSPACE}$

- (Proof omitted)

Nondeterministic log space computation

- **Definition:** NL is the class of languages that can be decided by a nondeterministic log-space Turing machine
- Equivalently, we imagine a log-space verifier
  - Technicality: Verifier reads certificate one time from left to right



The s-t connectivity problem

- $\mathrm{STCONN} = \{\langle G, s, t\rangle : G$ is a digraph, s and t are vertices, and there is a directed path from s to $t\}$
- **Claim:** $\mathrm{STCONN} \in \mathrm{NL}$
- **Proof sketch:** Take a nondeterministic walk through G starting from s for $|V|$ steps. If we ever reach t, accept; otherwise, reject.
- Verifier perspective: Certificate = path from s to t

Two surprises about $\mathbf{NL}$

- We expect that $\mathrm{P} \neq \mathrm{NP}$. In contrast:

Savitch's Theorem: $\mathrm{NL} \subseteq \mathrm{SPACE}(\log^2 n)$

- We expect that $\mathrm{NP} \neq \mathrm{coNP}$. In contrast:

Immerman-Szelepcsényi Theorem: $\mathrm{NL} = \mathrm{coNL}$

Proof of Savitch's theorem

Savitch's Theorem: $\mathrm{NL} \subseteq \mathrm{SPACE}(\log^2 n)$

- Proof step 1: Show that $\mathrm{STCONN} \in \mathrm{SPACE}(\log^2 n)$
- Proof step 2: Show that STCONN is "NL-complete"

Savitch's algorithm

- **Claim (Savitch's algorithm)**: $\mathrm{STCONN} \in \mathrm{SPACE}(\log^2 n)$
- **Proof sketch:** Let's figure out: is there a path from s to t of length at most 2^k?
  1. For all $m \in V$:
     a) Recursively figure out whether there is a path from s to m of length at most 2^{k-1}
     b) Recursively figure out whether there is a path from m to t of length at most 2^{k-1}
     c) If both such paths exist, halt and accept
  2. Halt and reject
- Space complexity is $O(k \log n)$, which is $O(\log^2 n)$ when $k = \lceil \log |V| \rceil$

Proof of Savitch's theorem

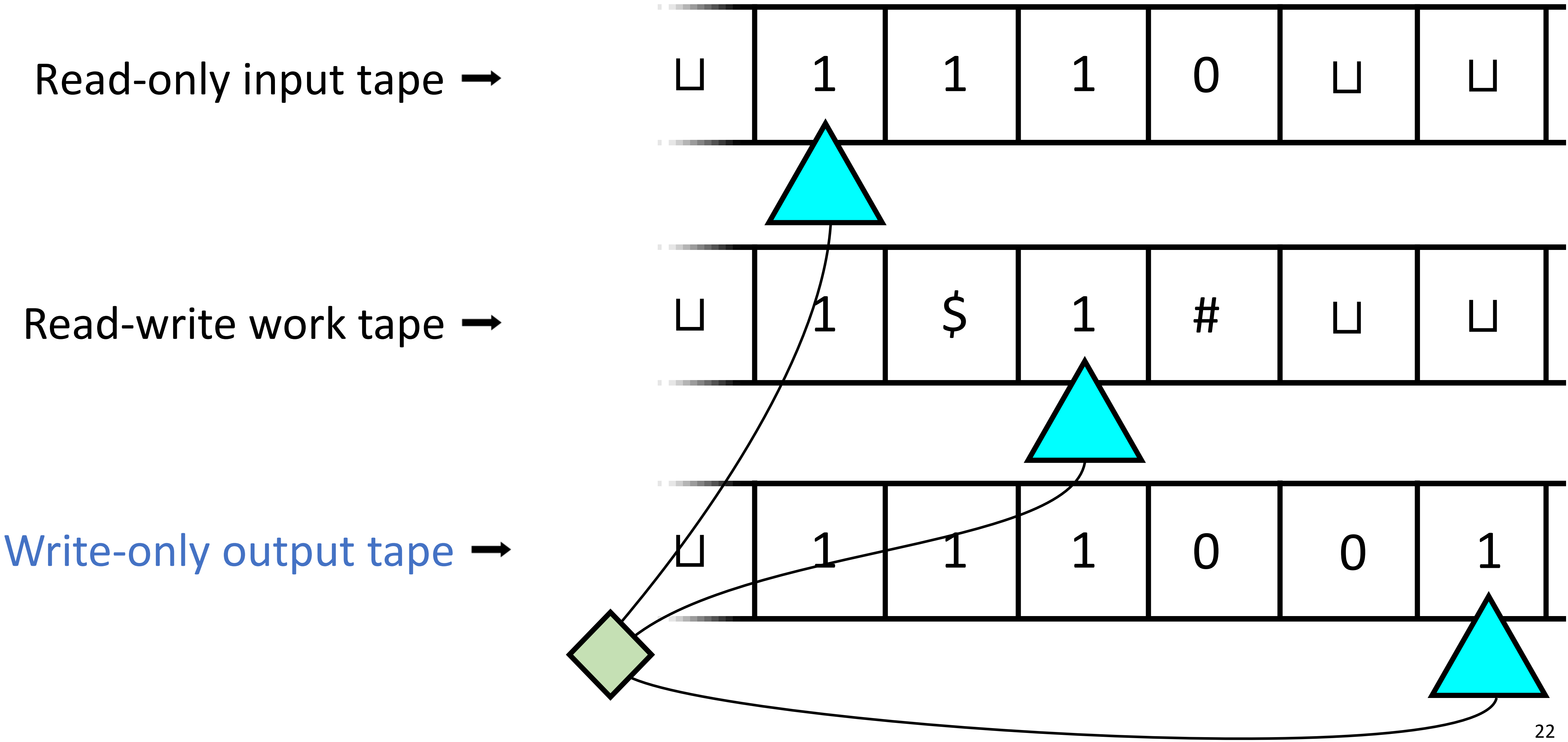
Savitch's Theorem: $\mathrm{NL} \subseteq \mathrm{SPACE}(\log^2 n)$

- Proof step 1: Show that $\mathrm{STCONN} \in \mathrm{SPACE}(\log^2 n)$ ✓
- Proof step 2: Show that STCONN is "NL-complete"

Log-space reductions

- To prove Savitch's theorem, we will use a new type of reduction
- Let $Y_1, Y_2 \subseteq \{0,1\}^*$
- **Definition:** We write $Y_1 \leq_{\mathrm{L}} Y_2$ if there exists $\Psi: \{0,1\}^* \to \{0,1\}^*$ such that
  - For every $w \in Y_1$, we have $\Psi(w) \in Y_2$ ("YES maps to YES")
  - For every $w \notin Y_1$, we have $\Psi(w) \notin Y_2$ ("NO maps to NO")
  - Ψ can be computed in $O(\log n)$ space ⬅ Definition on next slides

Space-bounded "transducer"



Space complexity for string-valued functions

- Let $\Psi: \{0,1\}^* \to \{0,1\}^*$ and let $S: \mathbb{N} \to \mathbb{N}$
- **Def:** We say Ψ is computable in $O(S)$ space if there is a 3-tape TM M such that:
  - If we initialize M with w on tape 1, then it halts with $\Psi(w)$ on tape 3
  - M never modifies tape 1 and M's behavior does not depend on what it reads on tape 3
  - The tape 1 head is always located within one cell of the input
  - When the input has length n, the tape 2 head visits $O\big(S(n)\big)$ cells

NL-completeness

- Let $Y \subseteq \{0, 1\}^*$
- **Definition:** We say that Y is NL-complete if $Y \in \mathrm{NL}$ and for every $Z \in \mathrm{NL}$, we have $Z \leq_{\mathrm{L}} Y$

STCONN is NL-complete

Theorem: STCONN is NL-complete

- **Proof:** We have already shown $\mathrm{STCONN} \in \mathrm{NL}$
- Now let M be a nondeterministic log-space TM that decides Y
- Reduction: $\Psi(w) = \langle G, s, t \rangle$
- Each vertex in G represents a "configuration" of M on w, namely, the internal state, the contents of the work tape, and the locations of heads

$\mathbf{STCONN}$ is $\mathbf{NL}$-complete

- We put an edge from u to v if M can go from u to v in a single step (with w written on input tape)
- We let $s =$ the initial configuration and $t =$ the accepting configuration
- (Without loss of generality, the accepting configuration is unique)
- YES maps to YES ✔ NO maps to NO ✔
- Exercise: The reduction can be computed in $O(\log n)$ space ✔

Proof of Savitch's theorem

Savitch's Theorem: $\mathrm{NL} \subseteq \mathrm{SPACE}(\log^2 n)$

- Proof step 1: Show that $\mathrm{STCONN} \in \mathrm{SPACE}(\log^2 n)$ ✔
- Proof step 2: Show that STCONN is NL-complete ✔
- Proof step 3: Show that $\mathrm{SPACE}(\log^2 n)$ is closed under log-space mapping reductions