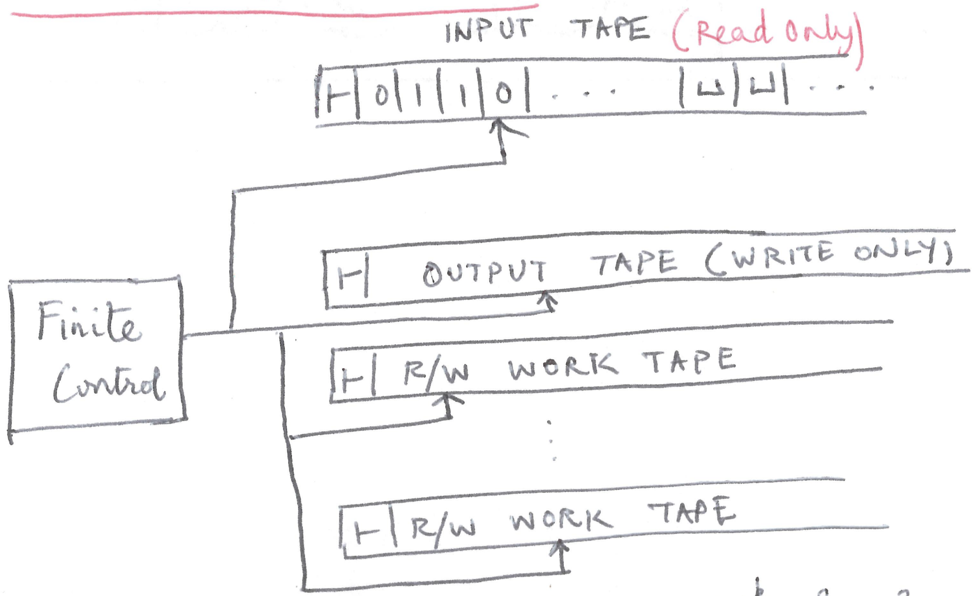
TURING M/C MODEL



$$\delta : Q \times \Gamma \times \Gamma^k \longrightarrow Q \times (\Gamma \times \{L, R\})^k \times \{L, R\} \times (\Gamma \cup \{\epsilon\})$$

$$\text{Configuration} \in Q \times \underbrace{\{1..n\}}_{\substack{\text{Inp head pos.}\\ \text{(input size} = n)}} \times \underbrace{\mathbb{N}^k}_{\text{Head pos of } k \text{ work-Tapes.}} \times (\underbrace{\Gamma^* \sqcup^\omega}_{k \text{ work-tapes}})^k$$

Lecture 13: Time and Space Complexity Classes

Date: October 5, 2023.

Time and Space Bounds: Let $T : \mathbb{N} \to \mathbb{N}$ and $S : \mathbb{N} \to \mathbb{N}$ be functions.

- A (deterministic/nondeterministic) Turing machine M is said to **run in time** $T(n)$ (or is $T(n)$ **time bounded**) if on all inputs x, **all** computations of M on x take at most $T(|x|)$ steps before halting.
- A(deterministic/nondeterministic) Turingmachine$M$ is said to **runs in space** $S(n)$ (or is $S(n)$ **space bounded**) if on all inputs x, **all** computations of M on x use at most $S(|x|)$ worktape cells.

Time/Space Classes:

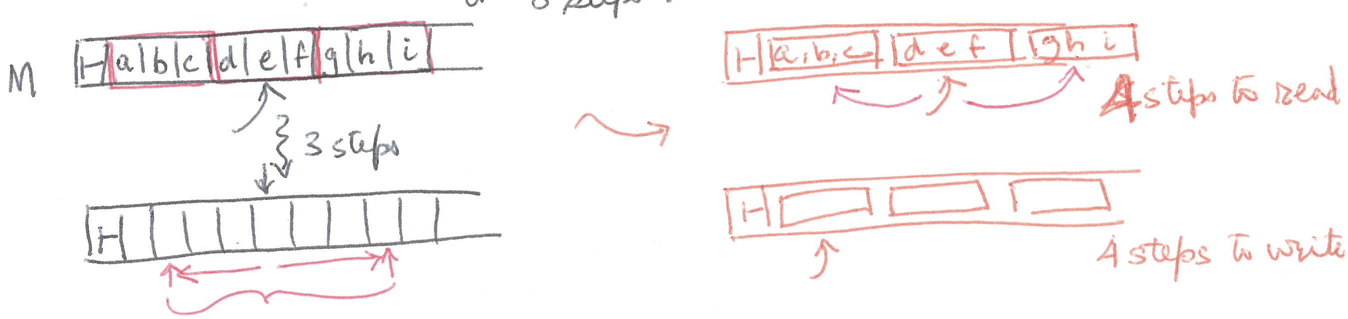
$$\begin{aligned}
\mathrm{DTIME}(T(n)) &= \{\mathbf{L}(M) \mid M \text{ is a DTM running in time } T(n)\}\\
\mathrm{NTIME}(T(n)) &= \{\mathbf{L}(M) \mid M \text{ is a NTM running in time } T(n)\}\\
\mathrm{DSPACE}(S(n)) &= \{\mathbf{L}(M) \mid M \text{ is a DTM running in space } S(n)\}\\
\mathrm{NSPACE}(S(n)) &= \{\mathbf{L}(M) \mid M \text{ is a NTM running in space } S(n)\}
\end{aligned}$$

If $\mathcal{A}$ is a collection of problems then co-$\mathcal{A} = \{L \mid \overline{L} \in \mathcal{A}\}$.

Theorem 1 (Linear Speedup/Compression). *For $T(n) \geq n+1$ and any constant $c \geq 1$*

$$\begin{aligned}
\mathrm{DTIME}(cT(n)) &\subseteq \mathrm{DTIME}(T(n))\\
\mathrm{NTIME}(cT(n)) &\subseteq \mathrm{NTIME}(T(n))\\
\mathrm{DSPACE}(cS(n)) &\subseteq \mathrm{DSPACE}(S(n))\\
\mathrm{NSPACE}(cS(n)) &\subseteq \mathrm{NSPACE}(S(n))
\end{aligned}$$

$A \in \mathrm{DTIME}(cT(n))$. $\exists$ DTM M s.t $A = L(M)$ and M is $cT(n)$- Time bounded
New Algo for A M': Simulate M s.t every k steps of M will be simulated in 8 steps.



Order Notation: $f(n) = O(g(n))$ if there are constants c, n_0 such that for all $n > n_0$, $f(n) \leq cg(n)$. $g(n)$ is said to be an **asymptotic upper bound** of $f(n)$.

$f(n) = \Omega(g(n))$ if there are constants c, n_0 such that for all $n > n_0$, $f(n) \geq cg(n)$. $g(n)$ is said to be an **asymptotic lower bound** of $f(n)$.

Lemma 2. DTIME($T(n)$) $\subseteq$ NTIME($T(n)$) *and* DSPACE($S(n)$) $\subseteq$ NSPACE($S(n)$).

Deterministic TM is a special Nondeterministic TM.

Lemma 3. DTIME($T(n)$) $\subseteq$ DSPACE($T(n)$).

In $T(n)$-steps you can only write in $T(n)$-new cells.

Lemma 4. *For* $S(n) \geq \log n$, DSPACE($S(n)$) $\subseteq$ DTIME($2^{O(S(n))}$).

$A \in$ DSPACE$(S(n))$. $\exists$ DTM M s.t. $A = L(M)$ and M is $S(n)$-space bounded.

New Algo for A: Simulate M. If M takes "too much" time then reject the input.

Computation of M on input x ($|x| = n$).

$C_0 \vdash C_1 \vdash C_2 \cdots\cdots$

$\uparrow$ initial configuration.

\# states

size of M's Tape alphabet.

$$\# \text{ Configurations of } M \text{ on inputs of length } n = c\, n\, k\, S(n)\, b^{kS(n)} \leq d^{S(n)} = 2^{O(S(n))}$$

If the computation of M takes more that $d^{S(n)}$-steps then it is looping and so M will not accept the input.

New Algo for A:

Input x

Run M for $d^{S(n)}$-steps

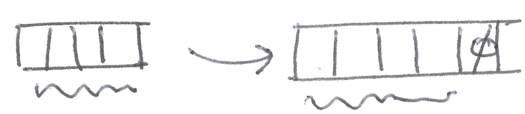
Accept if M does

else reject.

marking off $S(n)$ cells in worktape and then counting in base d.

Assume $S(n) = 1$

When M uses more than S cells, M' will mark-off an additional cell.



Lemma 5. NTIME($T(n)$) $\subseteq$ DSPACE($T(n)$). [NTIME($T(n)$) $\subseteq$ NSPACE ($T(n)$)]

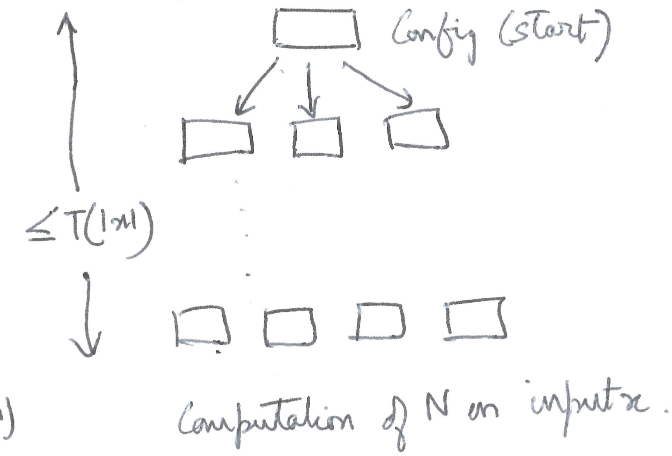
$A \in$ NTIME ($T(n)$) $\exists$ NTM N s.t. $A = L(N)$ and N is $T(n)$-time bounded.

Deterministic algo is "simulate N" on the input.

Det(N): Input x.

For all choices in lexicographic ordering.

Run N on x using the current choice.



Storage: Choice $\rightarrow T(n)$

Current config of N $\rightarrow T(n)$

$O(T(n))$

Lemma 6. *For* $S(n) \geq \log n$, NSPACE($S(n)$) $\subseteq$ DTIME($2^{O(S(n))}$).