Complexity Theory

Functions $T : \mathbb{N} \to \mathbb{N}$ and $S : \mathbb{N} \to \mathbb{N}$ that are non decreasing i.e $\forall n, m \quad n < m \Rightarrow T(n) \leq T(m)$ and $S(n) \leq S(m)$.

Time Bounded TM A deterministic/nondeterministic TM M runs in time $T(n)$ if on any input u, <u>all</u> computations of M on u take at most $T(|u|)$ steps.

Space Bounded TM A deterministic/nondeterministic TM M uses space $S(n)$ if on any input u, <u>all</u> computations of M on u use at most $S(|u|)$ work tape cells.

- A cell is used if it is written to at least once during a computation
- Space can be reused.

$\mathrm{DTIME}(T(n)) = \{A \mid \exists$ DTM M that runs in time $T(n)$ and $L(M) = A\}$

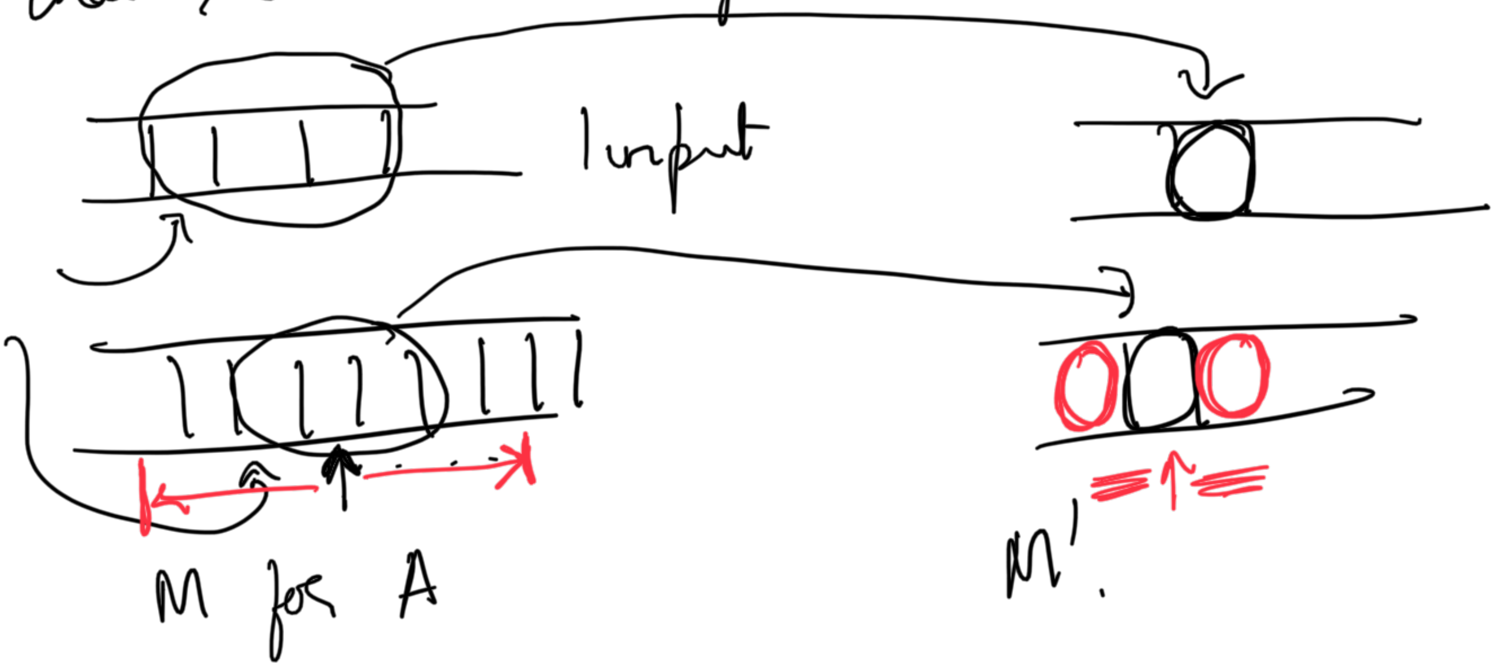
$\mathrm{NTIME}(T(n)) = \{A \mid \exists$ NTM M that runs in time $T(n)$ and $L(M) = A\}$

$\mathrm{DSPACE}(S(n)) = \{A \mid \exists$ DTM M that uses space $S(n)$ and $L(M) = A\}$

$\mathrm{NSPACE}(S(n)) = \{A \mid \exists$ NTM M that uses space $S(n)$ and $L(M) = A\}$

Linear Speedup Theorem For any $A \in DTIME(T(n))$ (or $NTIME(T(n))$) and $c > 0$, there is a deterministic (nondeterministic) TM that solves A in time $cT(n) + n$.

Linear Compression Theorem For any $A \in DSPACE(S(n))$ (or $NSPACE(S(n))$) and $c > 0$, there is a deterministic (nondeterministic) TM that solves A in space $cS(n)$.



Since constant factors don't matter we use O-notation.

Definition A function $f(n)$ is $O(g(n))$ if there is $\exists c\ \exists n_0 \in \mathbb{N}.\ \forall n \geq n_0\ f(n) \leq c\,g(n)$

Invariance Thesis Any mechanical procedure can be simulated on a TM with at most a polynomial slow-down and no change in space requirements

Robust Complexity Classes

- Our statements to platform independent
- Closed under function composition.
- Complexity Classes to contain important natural problems.

$L = \text{DSPACE}(\log n)$ $\quad$ $NL = \text{NSPACE}(O(\log n))$

$P = \bigcup_{k \in \mathbb{N}} \text{DTIME}(n^k)$ $\quad$ $NP = \bigcup_{k \in \mathbb{N}} \text{NTIME}(n^k)$

$\text{PSPACE} = \bigcup_{k \in \mathbb{N}} \text{DSPACE}(n^k)$ $\quad$ $\text{NPSPACE} = \bigcup_{k \in \mathbb{N}} \text{NSPACE}(n^k)$

$\text{EXP} = \bigcup_{k \in \mathbb{N}} \text{DTIME}(2^{n^k})$ $\quad$ $\text{NEXP} = \bigcup_{k \in \mathbb{N}} \text{NTIME}(2^{n^k})$

Proposition $\text{DTIME}(T(n)) \subseteq \text{DSPACE}(T(n))$

$\text{NTIME}(T(n)) \subseteq \text{NSPACE}(T(n))$

Proof In any single step a TM can write to at most one new cell.

Proposition $\text{DTIME}(T(n)) \subseteq \text{NTIME}(T(n))$

$\text{DSPACE}(S(n)) \subseteq \text{NSPACE}(S(n))$

Proof Every deterministic TM is special NTM.

Theorem $\text{NSPACE}(S(n)) \subseteq \text{DTIME}(n 2^{O(S(n))})$

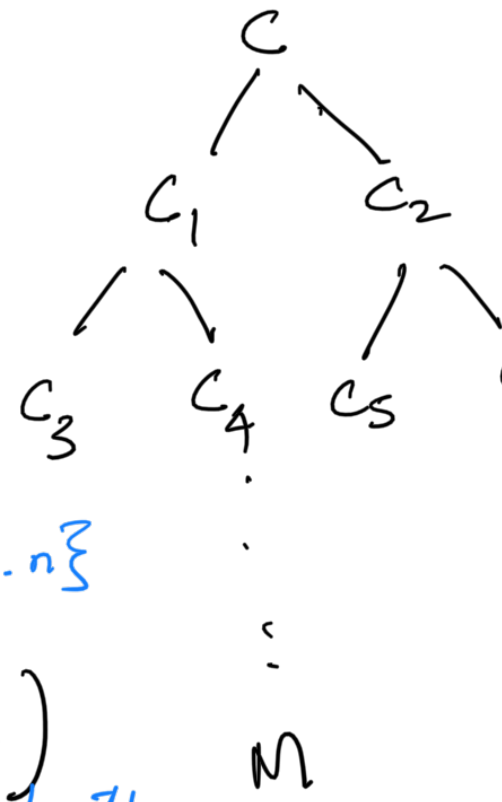
In other words, $S(n) \geq \log n$,

NSPACE$(S(n)) \subseteq$ DTIME$(2^{O(S(n))})$

Proof Let A be a problem solved a NTM M in space $S(n)$. M has 1 work tape

Configuration of M

- State of M
- Contents of work tape
- Head positions on the input & work-tape



$(q,$ input head position$,$ work tape head$,$ work-tape$)$

input head position $\in \{1..n\}$

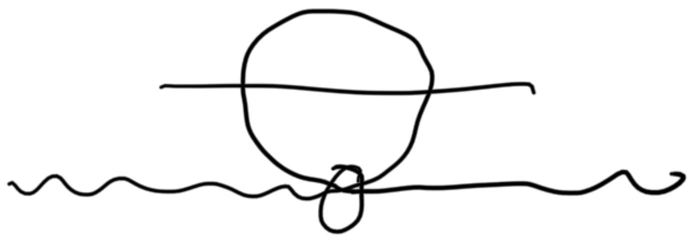
work tape head $\in \{1, .. S(n)\}$

work-tape: string of length $S(n)$

configuration of M on input of length n

$= Q \cdot n \cdot S(n) \cdot c^{S(n)} = n \cdot 2^{O(S(n))}$

If M has an accepting computation on input then M has an accepting computation of length $\leq n\, 2^{O(S(n))}$.



Configuration Graph

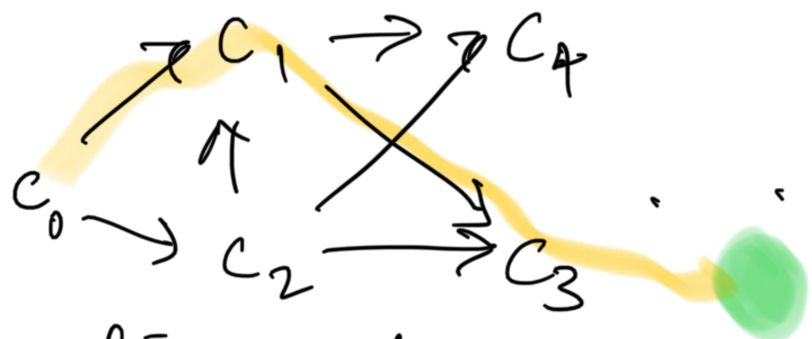
Directed graph. Vertices are configurations of M on input u. And there is an edge from c to c' if the TM M

can go from c to c' in one step.

Algorithm for A: Given input u, determine if $u \in A$.

determine if M accepts u.

- "Construct" configuration graph of M on u
- Determine if there is a path from the initial configuration to an accepting configuration.



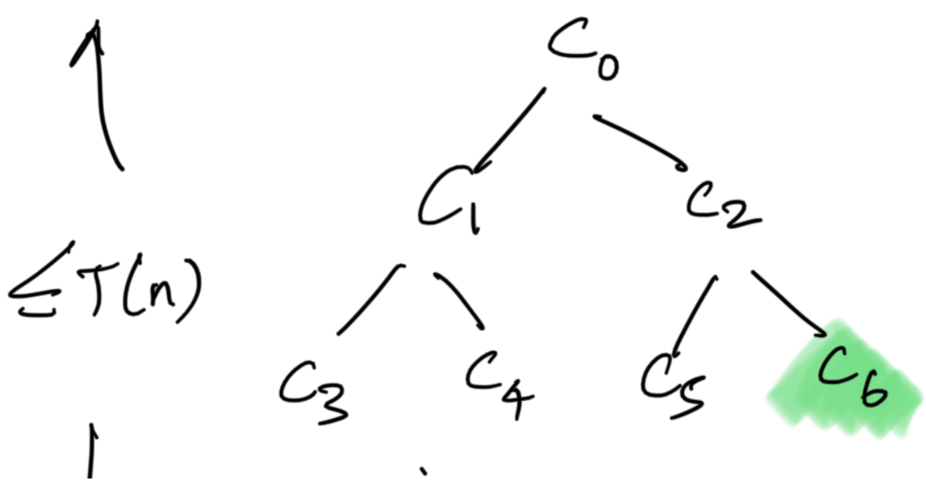
Running DFS/BFS on the configuration graph.

$$\text{Time} \le |\text{Configuration Graph}|$$
$$= n\, 2^{O(S(n))}.$$

Theorem $NTIME(T(n)) \subseteq DSPACE(T(n))$

Proof Let NTM M running in $T(n)$ time that solves A

Consider input u of length n.



To determine $w \in A$, checking if one of the configurations in the computation tree of M on w is accepting.

Run DFS on the computation tree.

Space Size × hight call stack

$$T(n) \times T(n) = T^2(n)$$