Space Complexity
Space Complexity

• In contrast to time, space complexity measures the memory requirements.

• In the TM model, the # of cells a TM M uses for a problem/language.

• Define the function $S: \mathbb{N} \rightarrow \mathbb{N}$ s.t. $S(n)$ is the maximum # of cells M uses on the work tapes (excluding the input tape) for any input of length $n$.

• Say $M$ runs in space $S(n)$. 
Space Complexity

• Say a language \( L \in \text{SPACE} \ (S(n)) \) if there is a TM \( M \) deciding \( L \) and \( M \) uses space \( O(s(n)) \).

• Similarly, say a language \( L \in \text{NSPACE} \ (S(n)) \) if there is a NDTM \( M \) deciding \( L \) and \( M \) uses space \( O(S(n)) \) (every branch halts and uses space \( O(S(n)) \)).

• Typically, require the function \( s \) to be space-constructible, i.e., \( S(n) \) can be computed in space \( O(S(n)) \).
Space Complexity

- Since work tapes are separated from input tapes, it makes sense to consider $S(n) < n$.

- In contrast to time complexity where we generally require $T(n) \geq n$.

- However, require $S(n) \geq \log n$ so that the TM can at least remember the index of the input cell.
Relation to Time Complexity

• First, we have $\text{DTIME}(S(n)) \subseteq \text{SPACE}(S(n))$. Why?

• A TM can only access one cell per step.

• However, a space $S(n)$ TM can run much longer than $S(n)$ steps, e.g., count from 1 to $2^{S(n)}-1$ (space can be reused).
Relation to Time Complexity

- Theorem: \( \text{DTIME}(S(n)) \subseteq \text{SPACE}(S(n)) \subseteq \text{NSPACE}(S(n)) \subseteq \text{DTIME}(2^{O(S(n))}) \).

- Up to logarithmic terms, the only relation known between time and space complexity

- Any improvement would be a major result.
Configuration Graph

• Let $M$ be a DTM or NDTM.

• Recall the configuration of a TM: state, head positions, all tape contents.

• We can ignore the blank entries of the tapes, since they contain no information.

• So for a space $S(n)$ TM, there are at most $S(n)$ non blank tape entries.
Configuration Graph

• For every space $S(n)$ TM $M$ (DTM or NDTM) and input $x \in \{0, 1\}^*$

• Denote the configuration graph of $M$ on input $x$ by $G_{M,x}$.

• $G_{M,x}$ is a directed graph whose vertices correspond to all possible configurations of $M$.

• There is a directed edge from a configuration $C$ to another configuration $C'$ if $C'$ can be reached from $C$ in one step according to $M$’s transition function.
Configuration Graph

- Can modify $M$ to erase all its work tapes before halting and put all heads at the left end.

- $\exists$ a single accept configuration $C_{accept}$.

- Claim: $M$ accepts input $x$ iff $\exists$ a directed path in $G_{M,x}$ from $C_{start}$ to $C_{accept}$. 
Configuration Graph

Claim: If $M$ uses space $S(n)$, then

1) Every vertex in $G$ can be represented by $cS(n)$ bits for some constant $c$ that depends only on $M$’s alphabet size and # of tapes.

2) There is an $O(S(n))$ size CNF formula $\psi_{M,x}$ s.t. $\forall$ two strings $C$, $C'$, $\psi_{M,x}(C, C')=1$ iff $C$, $C'$ encode two configurations s.t. $C'$ can be reached from $C$ in one step.
Configuration Graph

• Proof of the claim:

• 1) A configuration is completely decided by the state, head positions and non black entries of the tapes. Encode each one of them and combine together. The total size is $O(S(n))$.

• 2) Same idea as in the proof of Cook-Levin: express one step of computation as the AND of many small checks, each of size $O(1)$. The # of checks is $O(S(n))$. 
Proof of the Theorem

• Thm: $\text{DTIME}(S(n)) \subseteq \text{SPACE}(S(n)) \subseteq \text{NSPACE}(S(n)) \subseteq \text{DTIME}(2^{O(S(n))})$.

• $\text{DTIME}(S(n)) \subseteq \text{SPACE}(S(n)) \subseteq \text{NSPACE}(S(n))$ is clear.

• To show $\text{NSPACE}(S(n)) \subseteq \text{DTIME}(2^{O(S(n))})$, first constructing the configuration graph $G_{M,x}$ in time and then use BFS/DFS to search from a path from $C_{\text{start}}$ to $C_{\text{accept}}$. 
Space Complexity Classes

- $\text{PSPACE} = \bigcup_{c \geq 0} \text{SPACE}(n^c)$.
- $\text{NPSPACE} = \bigcup_{c \geq 0} \text{NSPACE}(n^c)$.
- $L = \text{SPACE}(\log n)$.
- $NL = \text{NSPACE}(\log n)$. 
Space Complexity Classes

• Example: SAT ∈ PSPACE. Why?

• Suppose a CNF formula ψ has k variables and is represented by n bits. So k ≤ n.

• Then a DTM M can decide if ψ ∈ SAT by cycling over all $2^k$ assignments using the same space, and check if one is a satisfying assignment. Total space=O(n).

• The same argument applies to all NP languages, so NP ⊆ PSPACE.
More Examples

• EVEN={x: x has an even # of 1’s}

• MULT={\([n]\), \([m]\), \([nm]\)}.

• Both languages are in the class L (can be solved in logspace).

• Seems hard to conceive more complicated problems that can be solved in logspace, however, cannot even rule out SAT \(\in\ L\) (i.e., not known if NP\(\neq L\)).
More Examples

- PATH=\{<G, s, t>: G is a directed graph and \exists a directed path in G from s to t.\}

- Claim: PATH ∈ NL.

- If \exists a path in G from s to t, then \exists a path with length at most n (# of vertices).

- Starting from s, an NDTM N can guess a next vertex each time, and continue the walk for n steps (each step only record the current index). N accepts iff the walk reaches t within n steps.
More Examples

• $\text{PATH} = \{ <G, s, t> : G \text{ is a directed graph and } \exists \text{ a directed path in } G \text{ from } s \text{ to } t. \}$

• Claim: $\text{PATH} \in \text{NL}$.

• Whether $\text{PATH} \in \text{L}$ is an open problem.

• In fact, whether $\text{PATH} \in \text{L}$ is equivalent to whether $\text{NL}=\text{L}$, i.e., $\text{PATH}$ is $\text{NL}$-complete.