

Space Complexity

Model of computation

- As before, our model of computation will be the Turing machine (we'll consider both deterministic and non-deterministic TMs)
- One difference: we don't want to count the space required to store the input; so our TMs will have a read-only input tape, and a read/write work tape; only the space used on the work tape will be counted
- **Definition:** If M is a TM, and $w \in \Sigma^*$, then the space used by M on input w , denoted $s_M(w)$, is the number of cells on the work tape that are visited by M during its computation on w . Also, the worst-case space complexity of M on inputs of size n is

$$S_M(n) = \max\{s_M(w) \mid |w| = n\}$$

The class of languages decidable by a deterministic TM running in space $O(f(n))$ is denoted $\text{DSPACE}(f)$.

$$\text{DSPACE}(f) = \left\{ L \subseteq \Sigma^* \mid \exists \text{ TM } M \text{ s.t. } L(M) = L \text{ and } S_M(n) \in O(f(n)) \right\}$$

- **Definition:** If M is an NTM, and $w \in \Sigma^*$, then the space used by M on input w , denoted $s_M(w)$, is the maximum space used by any computation of M on input w . We denote by $S_M(n)$ the worst-case space complexity of M on inputs of size M , and define $\text{NSPACE}(f)$ to be the class of languages decidable by a non-deterministic Turing machine with worst-case space complexity in $O(f(n))$.

$$\text{NSPACE}(f) = \left\{ L \subseteq \Sigma^* \mid \exists \text{ NTM } M \text{ s.t. } L(M) = L \text{ and } S_M(n) \in O(f(n)) \right\}$$

Interesting space classes

- **Definition:**

$$\mathbf{L} = \text{DSPACE}(\log n)$$

$$\mathbf{NL} = \text{NSPACE}(\log n)$$

$$\mathbf{PSPACE} = \bigcup_{k>0} \text{DSPACE}(n^k)$$

$$\mathbf{NPSPACE} = \bigcup_{k>0} \text{NSPACE}(n^k)$$

- **Fact:** $\mathbf{L} \subseteq \mathbf{NL} \subseteq \mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{PSPACE} \subseteq \mathbf{NPSPACE}$

- **Proof:** Most of the inclusions are obvious. We'll prove $\mathbf{NL} \subseteq \mathbf{P}$, and $\mathbf{NP} \subseteq \mathbf{PSPACE}$.
- $\mathbf{NL} \subseteq \mathbf{P}$: Assume that $M = (Q, \Sigma, \Gamma, q_0, q_{accept}, q_{reject}, \delta)$ is a non-deterministic Turing machine with $S_M(n) \leq c \log n$. We must define a polynomial time, deterministic TM M' with $L(M') = L(M)$. Let $w \in \Sigma^*$ be an input of length n . The main observation is that the number of possible configurations of M on input w is bounded by a polynomial in n . To completely specify a configuration of M during its computation on w , we must specify:
 1. The current state of M . There are $|Q|$ possible states.
 2. The position of the input tape-head. We may assume that the input tape-head never moves past the ends of the input, so there are about n possible positions (technically $n + 2$, but we'll ignore the extra constant).
 3. The position of the work tape-head. By assumption there are at most $c \log n$ possible positions.
 4. The contents of the work tape. This is equal to the number of strings of length $c \log n$ over the alphabet Γ , which is

$$|\Gamma|^{c \log n} = 2^{c \log |\Gamma| \log n} = n^{c \log |\Gamma|}$$

Thus the total number of configurations which M may reach on input w is at most

$$|Q| \cdot cn \log n \cdot n^{c \log |\Gamma|} \leq n^k$$

for some constant k , as c , $|Q|$, and $|\Gamma|$ are all constants. The idea is to consider the "configuration graph" of M on input w , i.e. a graph $G_w = (V_w, E_w)$, where V_w contains the configurations which M may reach on input w , and E_w consists of the edges (C, C') where $C \vdash_M C'$. The deterministic machine M' does a DFS in the configuration graph starting from $I_M(w)$, and accepts iff an accepting configuration is reachable. As $|V_w| \leq n^k$, the time required is polynomial in n^k , and therefore polynomial in n .

- $\mathbf{NP} \subseteq \mathbf{PSPACE}$: Assume that $M = (Q, \Sigma, \Gamma, q_0, q_{accept}, q_{reject}, \delta)$ is a non-deterministic Turing machine and that $T_M(n) \leq p(n)$ for some polynomial p . We must construct a deterministic, polynomial-space TM M' with $L(M') = L(M)$. Let $B = \max\{|\delta(q, \sigma)| \mid q \in Q, \sigma \in \Gamma\}$ be the maximum number of transitions M has available at any stage. Then define M' = "on input w ...
 1. Let $n = |w|$
 2. For each $b_1 b_2 \dots b_{p(n)} \in \{1, \dots, B\}^{p(n)}$, simulate M on input w using b_i to decide which non-deterministic move to make in the i -th step. If M accepts then accept.

3. If M rejected in every iteration above, then reject.

Since p is polynomial, the amount of space required to store $b_1 \cdots b_{p(n)}$ is $O(p(n) \log B) = O(p(n))$, since B is constant. Also, since M runs in polynomial time, then a computation of M only uses polynomial space: at most $p(n)$ (space is never larger than time, since to visit t cells of a tape requires at least t moves). So one iteration of the loop requires polynomial space $O(p^2(n))$. Each subsequent iteration can re-use this space, so the total space is polynomial in n .

Savitch's theorem

- **Theorem (Savitch):** If $S(n) \geq \log n$, then $\text{NSPACE}(S) \subseteq \text{DSPACE}(S^2)$.
- We will see the proof of this theorem next class. As a consequence, we get the following surprising corollary.
- **Corollary: PSPACE = NSPACE.**
- **Proof:** We already know that **PSPACE** \subseteq **NSPACE**, so it suffices to prove that **NSPACE** \subseteq **PSPACE**. So assume $L \in \text{NSPACE}$, and prove that $L \in \text{PSPACE}$. Since $L \in \text{NSPACE}$ there is a constant k such that $L \in \text{NSPACE}(n^k)$. By Savitch's theorem, $L \in \text{DSPACE}((n^k)^2) = \text{DSPACE}(n^{2k})$. Since k is constant, so is $2k$, and so $L \in \text{PSPACE}$.
- Recall that in the case of time complexity, the analogous question of whether **P** = **NP** is unresolved, and the answer is believed to be negative: it is believed that non-determinism actually provides more computational power with respect to computing in polynomial time. On the other hand, the above result shows that non-determinism does not provide any additional power with respect to computing in polynomial space.
- Notice that Savitch's theorem does not imply that **L** equals **NL**. It does imply that

$$\text{NL} \subseteq \mathbf{L}^2$$

where $\mathbf{L}^2 = \text{DSPACE}(\log^2 n)$.