Space Complexity

- We now consider the complexity of computational problems in terms of the amount of space, or memory, they require
- Time and space are two of the most important considerations when we seek practical solutions to most problems
- Space complexity shares many of the features of time complexity
- It serves a further way of classifying problems according to their computational difficulty

Space Complexity

Definition

Let M be a deterministic Turing machine, DTM, that halts on all inputs. The space complexity of M is the function $f : \mathbb{N} \to \mathbb{N}$, where $f(n)$ is the maximum number of tape cells that M scans on any input of length n.

Definition

If M is a nondeterministic Turing machine, NTM, wherein all branches of its computation halt on all inputs, we define the space complexity of M, $f(n)$, to be the maximum number of tape cells that M scans on any branch of its computation for any input of length n.

Estimation of space complexity

Let $f : \mathbb{N} \to \mathbb{N}$ be a function. The space complexity classes, $\mathrm{SPACE}(f(n))$ and $\mathrm{NSPACE}(f(n))$, are defined by:

- $\mathrm{SPACE}(f(n)) = \{L \mid L$ is a language decided by an $\mathcal{O}(f(n))$ space DTM$\}$

- $\mathrm{NSPACE}(f(n)) = \{L \mid L$ is a language decided by an $\mathcal{O}(f(n))$ space NTM$\}$

Example

SAT can be solved with the linear space algorithm M_1:
$M_1 =$"On input $\langle \varphi \rangle$, where φ is a Boolean formula:

1. For each truth assignment to the variables $x_1, \ldots, x_n$ of φ:
   (a) Evaluate φ on that truth assignment;
   (b) If φ evaluates to 1, *accept*
2. If φ never evaluates to 1, *reject*."

Another Example

Testing whether a DFA accepts all strings,

$$ALL_{\text{DFA}} = \{\langle A \rangle \mid A \text{ is a DFA and } \mathscr{L}(A) = \Sigma^*\}$$
$$\overline{ALL_{\text{DFA}}} = \{\langle A \rangle \mid A \text{ is a DFA and } \mathscr{L}(A) \neq \Sigma^*\}$$

We show that $\overline{ALL_{\text{DFA}}} \in \text{SPACE}(n)$

Proof idea

Construct D, a deterministic linear space algorithm that decides $\overline{ALL_{\text{DFA}}}$ by enumerating all strings of length n. If A rejects one of them, D stops with "accept". If A accepts all, D will "reject."