Space Complexity

The biggest difference between time and space is that you can’t reuse time.
(Merrick Furst)

Definitions

The space complexity of a program $M$ is the function $f : \mathbb{N} \rightarrow \mathbb{N}$, where $f(n)$ is the maximum amount of space that $M$ uses on any input of length $n$.

Let $f : \mathbb{N} \rightarrow \mathbb{R}^+$ be a function.

$SPACE(f(n)) = \{ L \mid L$ is a language decided by an $O(f(n))$ space deterministic program.$\}$

$NSPACE(f(n)) = \{ L \mid L$ is a language decided by an $O(f(n))$ space nondeterministic program.$\}$

Examples

SAT is decidable in linear space $SPACE(n)$

$M_1 = \text{On input } \phi$, where $\phi$ is a Boolean formula:
1. For each truth assignment to the vars. of $\phi$:
   1a. Evaluate $\phi$ on that truth assignment
   2. If $\phi$ ever evaluated to true, accept, else reject.

$ALL_{NFA} = \{ \langle M \rangle \mid M$ is an NFA and $L(M) = \Sigma^* \}$

$ALL_{NFA}$ is decidable in $NSPACE(n)$:

$M_2 = \text{On input } M$, where $M$ is an NFA:
1. Run $M$ for $2^q$ steps, $q = \text{number of states}$.
   1a. Choose input nondeterministically.
   1b. Keep track of the possible states at each step.
   2. Accept if some step did not have accept state.
Key Results about PSPACE

Savitch’s theorem: $\text{NSPACE}(f(n)) \subseteq \text{SPACE}(f(n)^2)$

PSPACE is the class of languages that are decidable in polynomial space. That is, $\text{PSPACE} = \bigcup_{k \in \mathbb{N}} \text{SPACE}(n^k)$.

PSPACE = NPSPACE due to Savitch’s theorem.

Example PSPACE-complete problem:
Given an assignment to a set of boolean variables, and rules for changing values, such as:
if $x_a \land x_b$ then assign $x_c \leftarrow T$, $x_d \leftarrow F$
Is there a sequence of rules to make $x_g = T$?

Sublinear-Space Complexity

While PSPACE is interesting, it contains many hard problems (so we think).

To ensure tractable problems, consider sublinear space bounds.

An input of length $n$ already takes up linear space, so for sublinear space to make sense:
1. The input is restricted to be read-only.
2. Read-write memory is restricted to $o(n)$ space.

Consider graph algorithms that mark/unmark vertices. How much r/w space is needed?

Log-Space Complexity

$L$ is the class of languages that are decidable in logarithmic space using a deterministic program. In other words, $L = \text{SPACE}(\log n)$.

How much r/w space is needed to determine if an DFA $M$ accepts $w$?

$NL$ is the class of languages that are decidable in logarithmic space using a nondeterministic program. In other words, $NL = \text{NSPACE}(\log n)$.

How much r/w space to nondeterministically determine if an NFA $M$ accepts $w$?