CS 205

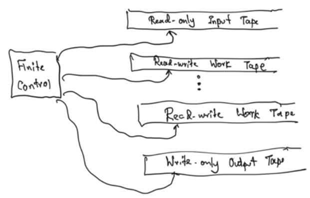
Lecture 36

13/4/21

Computational Complexity & Intractability

Complexity : Focuses on the amount of resources needed to solve a problem
- time
- space

Multi-tape TM : Computational model



Measuring Complexity

$t: \mathbb{N} \to \mathbb{N}$

$s: \mathbb{N} \to \mathbb{N}$

Time Complexity : We say $L \in \text{TIME}(t(n))$ if there is a multi-tape DTM which accepts L and uses no more than $t(|x|)$ steps on input x.

Note: Here x is any string in Σ^*.

Space Complexity : $L \in \text{SPACE}(s(n))$ if there is a multi-tape DTM which accepts L and uses no more than a total of $s(|x|)$ cells of its work tapes during its computation on input x.

Important : We consider only TMs that halt on all inputs i.e., we are talking about recursive/decidable languages.

Ex $L = \{ww^R \mid w \in \{0,1\}^*\}$

Single-tape TM : $O(n^2)$

Two-tape TM : $O(n)$

Nondeterministic TMs

Time Complexity : (1) A NTM M is said to be $t(n)$-bounded, if on any input w, no sequence of nondeterministic choices causes M to make more than $t(|w|)$ moves.

(2) $L \in \text{NTIME}(t(n))$ if there is a $t(n)$-time bounded NTM M that accepts L.

Relation between TIME $(t(n))$ and NTIME $(t(n))$

(1) $\text{TIME}(t(n)) \subseteq \text{NTIME}(t(n))$

(2) $\boxed{\text{NTIME}(t(n)) \subseteq \text{TIME}(2^{t(n)})}$

Complexity Classes

(1) L is in class P if there is some polynomial $T(n)$ s.t. $L \in \text{TIME}(T(n))$.

This is equivalent to saying: $P = \bigcup_{k \geq 1} \text{TIME}(n^k)$

(2) L is in class NP if there is some polynomial $T(n)$ s.t. $L \in \text{NTIME}(T(n))$.

Same as $NP = \bigcup_{k \geq 1} \text{NTIME}(n^k)$

Alternative Definition of NP

(1) A verifier for a language L is a DTM M where

$$L = \{w \mid M \text{ accepts } \langle w, c\rangle \text{ for some string } c\}$$

- We measure the time complexity of a verifier in terms of the length of w.
- A poly-time verifier runs in polynomial time in $|w|$.

(2) L is polynomially verifiable if it has a poly-time verifier.

(3) NP is the class of all languages that have poly-time verifiers.

Ex (1) Problems in P : Sorting, Searching, Shortest Path, Matching, ...

(2) NP : SAT, IS, VC, 3COLOUR,

Q: $P = NP$?

Why is class P is important?
- Seems to capture the class of efficiently solvable problems
- Not sensitive to problem encoding and computational model

$f: \Sigma^* \to \Sigma^*$ can be computed in PTIME

- Poly-time reductions $L_1 \leq_p L_2$ $\quad w \in L_1 \Leftrightarrow f(w) \in L_2$
- NP-complete : A problem L is NP-complete if
  (1) $L \in NP$
  (2) Any $X \in NP$ satisfies $X \leq_p L$.
- Cook-Levin Thm : SAT is NP-complete.