The Class P

Time Complexity Class

- Definition: time complexity class, $TIME(t(n))$

  Let $t : N \to N$.

  TIME$(t(n)) = \{L \mid L$ is a language decided by a $O(t(n))$ time TM $\}$

- The complexity of a problem depends on the models of computation.

- Example: $\{0^k 1^k \mid k \geq 0\}$

Complexity Relationship

- Theorem.

  Let $t(n)$ be a function, $t(n) \geq n$.

  $t(n)$ time multitape TM $\equiv O(t^2(n))$ time single-tape TM.

  Proof Idea: Simulating one step of the multitape TM uses at most $O(t(n))$ steps on the single-tape TM.

- Theorem.

  Let $t(n)$ be a function, $t(n) \geq n$.

  $t(n)$ time nondeterministic TM $\equiv 2^{O(t(n))}$ time deterministic TM.

  Proof Idea: Simulating $t(n)$ steps of the nondeterministic TM uses at most $2^{O(t(n))}$ steps on the single-tape TM.

The Class P

- Polynomial differences in running time are considered to be insignificant.
- Exponential differences are considered to be large.
- Definition: The class P of languages
  - decidable in polynomial time
  - on a deterministic single-tape TM

  $P = \cup_k \text{TIME}(n^k)$
- Features of the class P
  - invariant for all models polynomially equivalent to the deterministic single-tape TM.
  - roughly those problems that are realistically solvable.

Problems in P

- A polynomial time algorithm
  - A polynomial upper bound on the number of stages.
  - Each stage has a polynomial upper bound on the number of steps.
- A reasonable encoding method: polynomial time
- An example

  $PATH = \{< G, s, t > \mid G$ is a directed graph that has a directed path from s to t $\}$.

Another Example of Problems in P

- Theorem: Every CFL is a member of P.

  Proof. Give a polynomial time algorithm for each CFL

  - Let L be a CFL generated by CFG G in CNF.
  - Any derivation of a string w has $2|w| - 1$ steps.
  - Use dynamic programming technique: solve smaller problems first.