1. Complexity Relationships among Models

given that $t(n) \geq n$

Th Every $O(t(n))$ multitape TM has equivalent $O(t^2(n))$ single-tape TM.

Proof

M - multitape TM, S - single-tape TM

S requires $O(n)$ time to set up tape.

M performs $t(n)$ constant operations => S performs $t(n)$ operations.

However, for every operation of M, S needs to traverse its tape ($O(t(n))$ time)

=> S total time: $O(n) + O(t^2(n))$, since $t(n) \geq n$ = $O(t^2(n))$

given that $t(n) \geq n$

Th Every $O(t(n))$ single-tape NDTM has equivalent $2^{O(t(n))}$ single-tape DTM.

Proof

N - single-tape NDTM, D - single-tape DTM

Every branch of N has length $O(t(n))$.

Assuming a node can have at most b children, N has at most $b^{t(n)}$ leaves

D does concurrent breadth-first traversal of N => $O(t(n) b^{t(n)}) = 2^{O(t(n))}$

2. Class P

Def. Complexity of problem - worst-case complexity of best algorithm

Def. A is best algorithm for X if $\forall$ algorithm B for X, $\forall n$, $T_A(n) \leq T_B(n)$

Def. $TIME(t(n)) = \{L \mid L$ is decided by (single-tape) DTM in $O(t(n))$-time$\}$

Strong Church-Turing Thesis: Every physically realisable algorithmic process can be simulated on single-tape DTM with polynomial overhead.

Def. Algorithm with $f(n)$ time complexity is

-> polynomial-time if $f(n) \in O(n^c)$, $c > 0$ / $f(n) = n^{O(1)}$

-> exponential-time if $f(n) \in O(2^{n^c})$, $c > 0$ / $f(n) = 2^{n^{O(1)}}$

Def. $P = \bigcup_{k \geq 0} TIME(n^k) = \{L \mid L$ is decided by DTM in polynomial time$\}$

Th Every Context-Free Language is a member of P.

3. Class NP

Def. $NTIME(t(n)) = \{L \mid L$ is decided by NDTM in $O(t(n))$ time$\}$

Def. $NP = \bigcup_{k \geq 0} NTIME(n^k) = \{L \mid L$ is decided by NDTM in polynomial time$\}$

P - solution can be found in polynomial time

NP - solution can be verified in polynomial time

4. Karp Reduction

Def. $f: A \to B$ polynomial-time mapping s.t. $x \in A$ can be decided by deciding $f(x) \in B$.

$X \le Y$: $Y \in P \to X \in P$, $Y \in NP \to X \in NP$, $X \in$ NP-hard $\to Y \in$ NP-hard

Def. Problem A is NP-hard iff $\forall X \in NP$, $X \le A$.

Def. Problem X is NP-Complete iff $X \in NP$ and $X \in$ NP-hard