Time Complexity Classes

Measuring Complexity

Complexity of a problem = **the efficiency of the best algorithm for the problem**

Measure the efficiency by **time** or **space**, or both

Analyze the efficiency by the **growth of the function that relates the input size and the amount of resources used**

The **worst-case analysis** $\cdots$ analyze the function that maps each nonnegative integer n to the *maximum amount of resources used for solving any input of size* n with the best algorithm known

An alternative is the **average case analysis**

Deterministic Time Complexity Classes

Definition. **Let $t : \mathcal{N} \to \mathcal{N}$ be a function. A Turing machine N is $t(n)$ time (or $t(n)$ time-bounded) if for every $n \in \mathcal{N}$, and for every input x of length n, N on x halts within $t(n)$ steps.**

Definition. **Let $t : \mathcal{N} \to \mathcal{N}$ be a function. Define $\mathrm{TIME}(t(n)) = \{L \mid L$ is decided by an $O(t(n))$ time multi-tape Turing machine $\}$.**

Nondeterministic Time Complexity Classes

Definition. Let $t : \mathcal{N} \rightarrow \mathcal{N}$ be a function. A nondeterministic Turing machine N is $t(n)$ time if for every $n \in \mathcal{N}$, and for every input x of length n, N on x halts within $t(n)$ steps along all computation paths.

Definition. Let $t : \mathcal{N} \rightarrow \mathcal{N}$ be a function. Define $\mathrm{NTIME}(t(n)) = \{L \mid L$ is decided by an $O(t(n))$ time nondeterministic multi-tape Turing machine $\}$.

Relationships Among Models

Theorem. **For every $t(n) \geq n$, each $t(n)$ time multi-tape Turing machine has an equivalent $t(n)^2$ time single-tape Turing machine.**

The proof uses the 1-tape simulation of multi-tape Turing machines.

Relationships Among Models

Theorem. **For every $t(n) \geq n$, each $t(n)$ time nondeterministic Turing machine has an equivalent $2^{O(t(n))}$ time single-tape Turing machine.**

The proof goes as follows:

Step 1 We use the multi-tape version of the 3-tape deterministic simulation of nondeterministic 1-tape Turing machines.

Relationships Among Models

Step 2 We observe that if the nondeterministic machine is a $t(n)$ time machine, then on an input of length n, during examination of length-$t(n)$ computation paths, either we discover:

- the machine indeed accepts, or
- for all length-$t(n)$ paths, the machine rejects the input.

The latter allows to stop simulation with an assertion that the machine does not accept the input.

Relationships Among Models

Step 3 If the maximum number of branches is d, then the time required for the simulation is

$$(1 + d + d^2 + \cdots + d^{t(n)})ct(n)$$

for some constant c. Here $ct(n)$ is an upper bound of the time required to erase the tape, copy the input, and produce the description of next computation path. This is at most

$$2^{(\log d)t(n) + \log(c) + \log t(n)}.$$

This is at most $2^{c' t(n)}$ for some constant c'.

Relationships Among Models

Step 4 The above multi-tape deterministic machine can be simulated by a 1-tape Turing machine with running time of $(2^{c't(n)})^2 = 2^{2c't(n)} = 2^{O(t(n))}$.