

DEFINITION 1. A *k-string Turing machine with input and output* is a *k*-string Turing machine with a read-only input string and write-only output string. The cursor of the first string does not move backward except when it encounters $\sqcup$. When it encounters $\sqcup$, it moves backward.

DEFINITION 2. Suppose that for a *k*-string Turing machine M_k and an input x ,

$$(s, \triangleright, x, \triangleright, \epsilon, \dots, \triangleright, \epsilon) \xrightarrow{M_k^*} (H, w_1, u_1, \dots, w_k, u_k),$$

where H is one of the halting states. Then the space required by M_k on x is $\sum_{i=1}^k |w_i u_i|$. If M_k is with input and output, then the space required is: $\sum_{i=2}^{k-1} |w_i u_i|$.

DEFINITION 3. Let $f : \mathbb{N} \rightarrow \mathbb{N}$ and let M be a Turing machine; we say M operates within space bound $f(n)$ if the space required for any input x is less than or equal to $f(|x|)$.

DEFINITION 4. $\text{SPACE}(f(n))$: Languages decided by deterministic Turing machines with input and output that operate within space bound $f(n)$.

DEFINITION 5. $L = \text{SPACE}(\log n)$.

DEFINITION 6. A *nondeterministic Turing machine* is a quadruple $N = (K, \Sigma, \Delta, s)$, where

$$\Delta \subset (K \times \Sigma) \times [(K \cup \{h, \text{"yes"}, \text{"no"}\}) \times \Sigma \times \{\leftarrow, \rightarrow, -\}].$$

Notation: We will use **NDTM** for nondeterministic Turing machine and **DTM** for deterministic Turing machine.

REMARK 0.1. (1) Notice that if N is a NDTM, then for each combination of state and input symbol, there may be no action, one action, or more than one action.

(2) Notice that if M is a DTM, then for each combination of state and input symbol, there exists exactly one transition.

(3) If Δ is a function, then the NDTM becomes a DTM.

DEFINITION 7. NDTM's have applications in logic and AI.

DEFINITION 8. Let N be a NDTM; we say

configuration (q, w, u) *yields* configuration (q', w', u') in *one step*, denoted by

$$(q, w, u) \xrightarrow{N} (q', w', u')$$

if there exists a one-step transition of the machine from (q, w, u) to (q', w', u') .

Similar definitions for

$$(q, w, u) \xrightarrow{N^k} (q', w', u'),$$

and

$$(q, w, u) \xrightarrow[N^*]{1} (q', w', u')$$

DEFINITION 9. Let N be a NDTM with alphabet Σ , let L be a language over $\Sigma - \{\sqcup\}$, and let $x \in (\Sigma - \{\sqcup\})^*$. We say

- (1) N accepts x if there is a sequence of choices that result in a "yes" output state.
- (2) N rejects x if there is no sequence of choices leading to acceptance.
- (3) N decides L if $x \in L$ iff N accepts x .

DEFINITION 10. Let N be a NDTM and let Σ be an alphabet for M . Let L be a language over $\Sigma - \{\sqcup\}$ and let f be a function from the nonnegative integers to the nonnegative integers. We say that N decides L within time $f(n)$, if

- (1) N decides L .
- (2) $\forall x \in (\Sigma - \{\sqcup\})^*$, if $(s, \triangleright, x) \xrightarrow{N^k} (q, w, u)$, then $k \leq f(|x|)$. In other words, N does not have computation paths greater than $f(n)$.

DEFINITION 11. Let N_k be a k -string NDTM with input and output and let Σ be an alphabet for M . Let L be a language over $\Sigma - \{\sqcup\}$ and let f be a function from the nonnegative integers to the nonnegative integers. We say that N decides L within space $f(n)$, if

- (1) N decides L .
- (2) $\forall x \in (\Sigma - \{\sqcup\})^*$, if

$$(s, \triangleright, x, \triangleright, \epsilon, \dots, \triangleright, \epsilon) \xrightarrow{N_k^*} (q, w_1, u_1, \dots, w_k, u_k),$$

then $\sum_i = 2^{k-1} |w_i u_i| \leq f(|x|)$.

REMARK 0.2. N decides L within space $f(n)$ is similar to the definition for M_k decides L within space $f(n)$, where M_k is a k -string Turing machine with input and output.

DEFINITION 12. **NTIME**($f(n)$): Complexity class of languages that are decided by NDTM's operating within time $f(n)$.

DEFINITION 13. $NP = \text{NTIME}(n^k)$.

Handout2.ps

THEOREM 14. If L is decided by a NDTM N in time $f(n)$, then it is decided by a 3-string DTM in time $\mathcal{O}(c^{f(n)})$, where $c > 1$ is a constant depending on N .

REMARK 0.3. The above theorem says that any NDTM can be simulated by a DTM with exponential loss of efficiency.