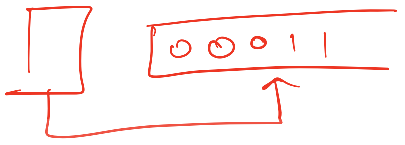
Variants of Turing Machines

Multi-tape Turing Machines :



This is like a Turing Machine but with several tapes and heads. In a single tape machine, there is no reason to stay put.

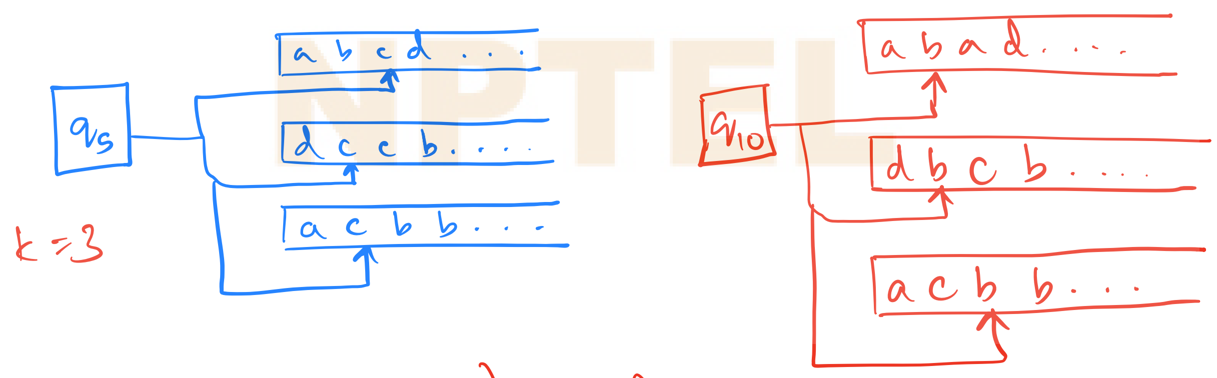
$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$

$M = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$

where $\delta : Q \times \Gamma^k \longrightarrow Q \times \Gamma^k \times \{L, R, S\}^k$

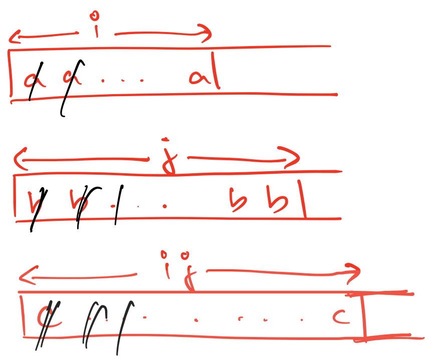
where k is the number of tapes.

$\delta(q_i, a_1, a_2 \ldots a_k) = (q_j, b_1, b_2, \ldots b_k, L, R, \ldots L)$



$\delta(q_5, c, c, c) = \delta(q_{10}, a, b, c, L, S, R)$

$\{a^i b^j c^k \mid i \cdot j = k\}$



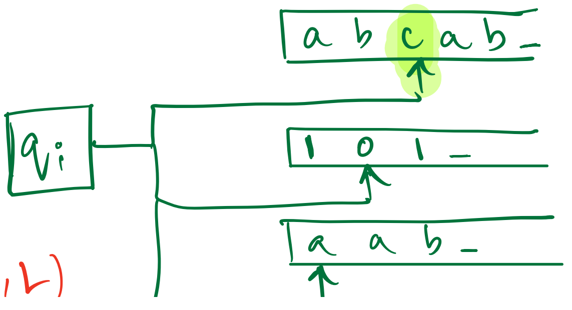
Can strike off one a from tape 1, and then j c's from tape 3. Repeat till we run out of a's. If all c's are also over, then accept. Else reject. If we run out of c's earlier, then reject.

Thus it is easier to check if input is of the form $a^i b^j c^k$ where $k = ij$.

Does multi-tape TM give more power? NO!

Theorem 3.13: Every multi tape TM is equivalent to a single tape TM.

Idea: Represent all the contents in one tape.



$\delta(q_i, c, 0, a)$

$= (q_i, b, 1, b, L, L, L)$

[illegible]

encode into a single tape

are delimiters

q_i: $|\#\ a\ b\ \hat{c}\ a\ b\ \#\ 1\ \hat{0}\ 1\ \#\ \hat{a}\ a\ b\ \#$

q_j: $|\#\ a\ \hat{b}\ b\ a\ b\ \#\ \hat{1}\ 1\ 1\ \#\ \hat{b}\ a\ b\ \#$

The alphabet is now $\Gamma \cup \hat{\Gamma} \cup \{\#\}$

<u>Simulation</u> : On input $w = w_1 w_2 \ldots w_n$.

1. Give input $\# \hat{w}_1 w_2 \ldots w_n \# \hat{\sqcup} \# \hat{\sqcup} \# \ldots \#$

2. Scan till you find the $(k+1)^{th}$ # symbol. Remember the head positions. Simulate δ of M. Make a second pass over the tape to make changes on the k virtual tape contents.

3. If any head needs to move to a blank space on the right, shift the tape contents after inserting $\hat{\sqcup}$.

<u>Corollary 3.15</u>: A language is Turing recognizable $\iff$ a multitape TM recognizes it.

Similarly for decidable languages.