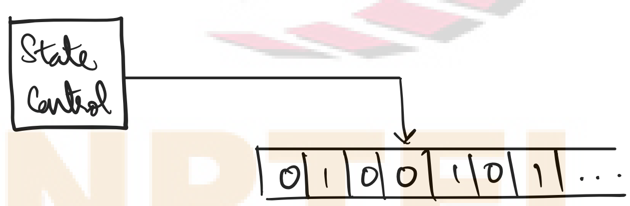
Computability Theory

Now we see Turing Machines - which are an abstract model of modern day computers. The main features of Turing machines are:

- → Read / write head
- → Head can move left or right.
- → Infinite tape (to one side)
- → Special accept / reject states which immediately come into effect.



Consider the language D.

$$D = \{ w \# w \mid w \in \{0,1\}^* \}$$

D is not a CFL. Can a Turing machine recognize D?

On tape : $\cancel{0}\cancel{1}\cancel{1}\cancel{0}\cancel{0}\#\cancel{0}\cancel{1}\cancel{1}\cancel{0}\cancel{0}\sqcup\sqcup\sqcup\ldots$

Blank

Reject $\begin{cases} \cancel{0}\cancel{1}\cancel{1}\cancel{0}\#\cancel{0}\cancel{1}\cancel{1}\cancel{0}\underline{0} \\ \cancel{0}\cancel{1}\cancel{1}\cancel{0}\underline{1}\#\cancel{0}\cancel{1}\cancel{1}\cancel{0} \end{cases}$

Reject

$\cancel{0}\cancel{1}100\#\cancel{0}\underline{0}100$

TM can do the following :

1. "Strike off" the first position from the left side, which is not already struck off. Remember this symbol, go past # and cross check with the first symbol in the right of # that is not already struck off. If not the same, reject. Else continue, or repeat. If no symbol is found after #, or no # symbol is found, then reject.

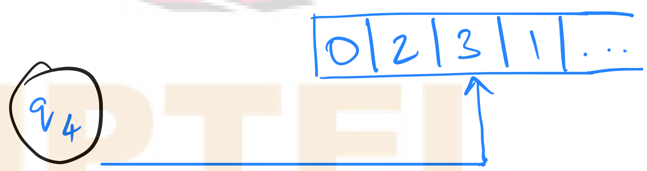
2. When all the symbols before # has been struck off, check if any more symbols remain after # (that are not struck off). If yes, reject. Else, accept.

We usually do not go into this level of detail. Only the high level idea will be explained. Most of the things can be accomplished, though the details could be tedious.

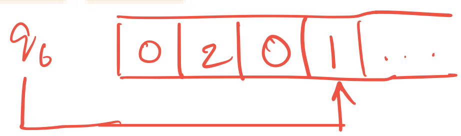
Q: How can a TM recognize $\{ww \mid w \in \{0,1\}^*\}$?

Formal Definition of Deterministic TM (Def 3.3): A Turing machine is a 7-tuple $(Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$ where Q, Σ, Γ are finite sets, such that

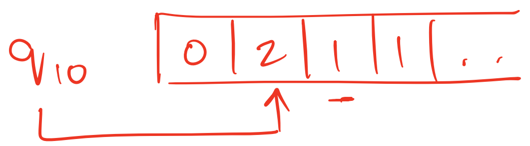
* $\Sigma \subseteq \Gamma$, $\sqcup \in \Gamma$, $\sqcup \notin \Sigma$.
* $q_{accept} \neq q_{reject}$
* $\delta(Q, \Gamma) \longrightarrow Q \times \Gamma \times \{L, R\}$

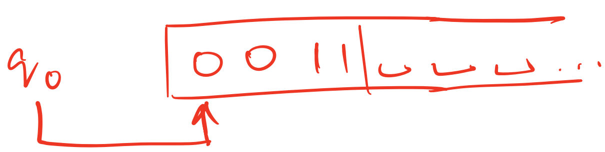


$\delta(q_4, 3) = (q_6, 0, R)$



$\delta(q_4, 3) = (q_{10}, 1, L)$





* The input is placed on the tape to the left most end. $w = w_1 w_2 \ldots w_n \in \Sigma^*$. The rest of the cells are blank ($\sqcup$). The blanks indicate the end of the input.

* δ indicates what to write and where to move to.

* Computation starts from the left most end of the tape. If the tape head ever tries to move to the left from the left most end, it remains there.

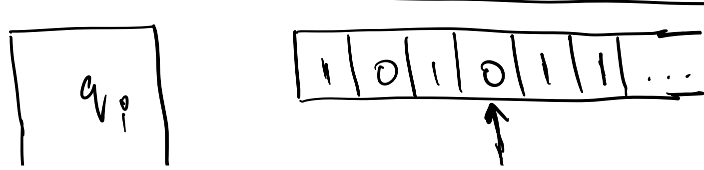
* Computations end when q_{accept} | q_{reject} is reached. Else it can also go on forever without halting. (looping).

Configuration : A configuration of a TM consists of three things

→ State : q_i

→ Head Position : 4^{th}

→ Tape Contents : $101011\sqcup\sqcup\ldots$

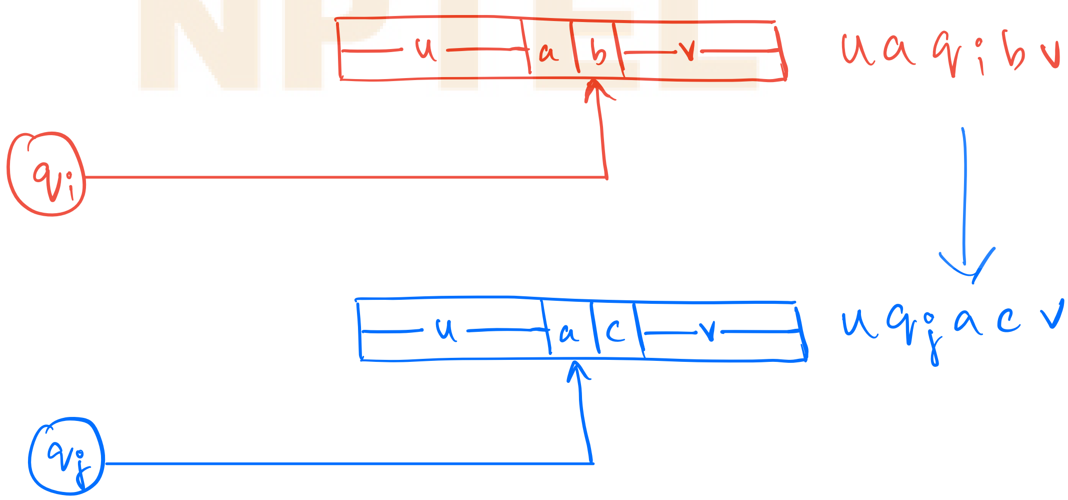


Denoted by : **$101\,q_i\,011 \sqcup \sqcup \ldots$**

Denotes that head is in 4^{th} position and state is q_i

We say **C_1 yields C_2** if the TM can go from configuration C_1 to configuration C_2 in one single step.

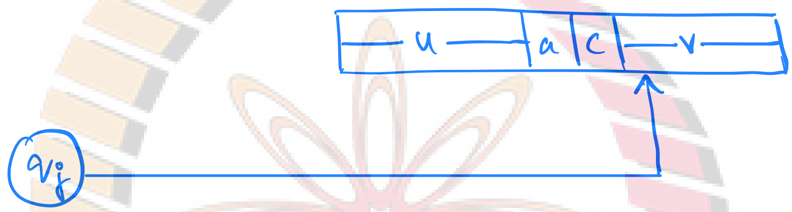
Let $u\,a\,q_i\,b\,v$ and $u\,q_j\,a\,c\,v$ be two different configurations, where $u, v \in \Gamma^*$, $a, b, c \in \Gamma$ and $q_i, q_j \in Q$.



So we say $u\,a\,q_i\,b\,v$ yields $u\,q_j\,a\,c\,v$.

This happens when $\delta(q_i, b) = (q_j, c, L)$

If instead we had $\delta(q_i, b) = (q_j, c, R)$



$u\,a\,c\,q_j\,v$

The TM accepts w if there is a sequence of configurations $C_1, C_2, \dots C_k$ such that

1. C_1 is the start configuration: $q_0 w$
2. $C_i \rightarrow C_{i+1}$ for all i
3. C_k is an accepting configuration (the state in C_k is q_{accept}).

The language recognized by M is denoted $L(M)$.

$$L(M) = \{ w \mid w \text{ is accepted by } M \}.$$