PDA

Pushdown Automata

automata + stack

$$(Q, \Sigma, \Gamma, \delta, q_0, F)$$

$\delta: Q \times \{ \Sigma_\cup \{\epsilon\}\} \times (\Gamma \cup \{\epsilon\}) \rightarrow P(Q \times \{\Gamma \cup \{\epsilon\}\})$

$q \qquad \alpha \qquad \times \qquad \gamma \qquad\qquad\qquad\qquad q', \gamma'$

$$\delta(q, \alpha, \gamma) = \{(q_1, \gamma_1), (q_2, \gamma_2)\}$$

If there is no transistion corresponding to input symbol and stack alphabet, then it is called halting (stucks at some position)

Transistion convention $\qquad \alpha, \beta \rightarrow \gamma$

input, stack alphabet

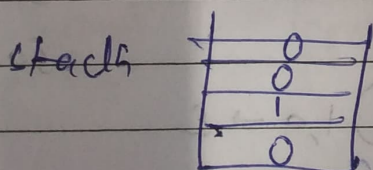eg. $L = \{ ww^R \mid w \in \{0, 1\}\}$

$w = 011$
$w^R = 110$
(reverse of w).

$\underbrace{0\ 1\ 1}_{w} \quad \underbrace{1\ 1\ 0}_{w^R} 1\ 0 \checkmark$

we dont know past or future inputs. So I must guess that current input is end of w.
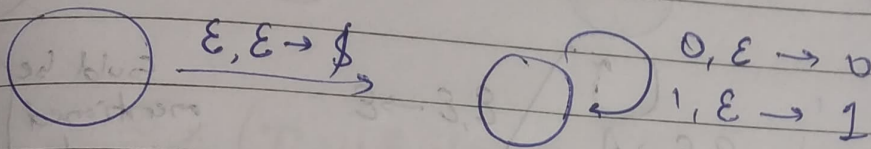
I assume end of w here

eg. 0 1 0 0

0 0 1 0

then remove the stack elements.

Pushstack
	0
	0
	1
	0

(input) w': 0 1 1 1 1 1 0

↓

lets assume split here
(we could be wrong)

$\varepsilon, \varepsilon \to \$$

$0, \varepsilon \to 0$
$1, \varepsilon \to 1$

$\varepsilon, \varepsilon \to \varepsilon \longrightarrow$ used for jumping to check for w^R. we don't

$0, 0 \to \varepsilon$
$1, 1 \to \varepsilon$

know exact mid point, so for every input symbol a thread will jump and checks for w^R.

Looking a thread 011 is processed.

1
1
0
$

and now let's assume it to be a mid point

till 04, 11 → stack

0
$

then for input 1, there is no transistion of 1, 0 → () so the thread is stuck therefore it halts.

eg. $L = \{a^i b^j c^k \mid i,j,k \geq 0$ and $i=j$ or $i=k\}$

thread1 $i=j$

$b, A \to \varepsilon$

$\varepsilon, \varepsilon \to \varepsilon$

$\longrightarrow$ O $\xrightarrow{\varepsilon, \varepsilon \to \$}$ O $\xrightarrow{\varepsilon, \varepsilon \to \varepsilon}$ O $\xrightarrow{\varepsilon, \$ \to \varepsilon}$ ((q))

$a, \varepsilon \to A$

$\varepsilon, \varepsilon \to \varepsilon$

$i=k$

Could be mentioned or not. that If isnt mentioned then thread halts and gets rejected

$a, \varepsilon \to \varepsilon$
$b, \varepsilon \to \varepsilon$
$c, \varepsilon \to \varepsilon$

(q reject)

$b, \varepsilon \to \varepsilon$ ↪ O

$\varepsilon, \varepsilon \to \varepsilon$

$c, A \to \varepsilon$ ↪ O $\xrightarrow{\varepsilon, \$ \to \varepsilon}$ ((q))

input thread $\Rightarrow$ aaa bbb cccc

aaa

going from thread 1
we get input accepted

	A	
	A	
a	A	
	$	

ANOTHER MACHINE MODEL

Turing Machine $\left\{\begin{array}{l}\text{Deterministic (DTM)}\\ \text{Non-Deterministic (NTM)}\end{array}\right.$

Automata + Tape → (similar to RAM module, but it cannot access random positions)

left end

tape head

Can go both forward and backward. or stay at same position

input alphabet.

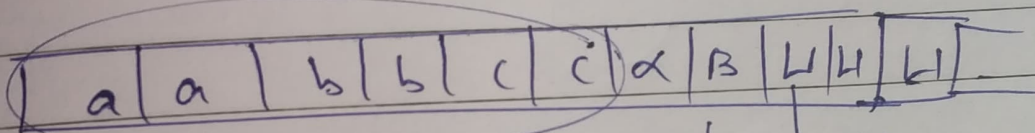$$(Q, \Sigma, T, \delta, q_0, F)$$

tape alphabet

DTM

left or right.

$$\delta: Q \times T \longrightarrow Q \times T \times \{L, R\}$$

NTM

$$\delta: Q \times (T \cup \{\epsilon\}) \longrightarrow Q \times (T \cup \{\epsilon\}) \times (L, R)$$

Here input comes on tape so.

$$\Sigma \subset T$$

a	a	b	b	c	c	α	B	⊔	⊔	⊔	...

↓ input.

→ blank spaces

some other
tape alphabets.