4 Pr

Turing Machine

* Introduction :-

[Finite state automata have finite amount of memory & hence cannot ~~to~~ do certain things.

eg :-

No finite state automata can accept language $\{a^n b^n / n \geq 1\}$. Though it is possible to have finite state automata for adding to arbitary long binary number we cannot have finite state automata which can multiple to arbitary long binary number.

To over~~all~~ come this problems the hypothetical m/c was propose by ~~two~~ Turing in 1936 & is known as Turing m/c.]

[Turing m/c is a simple mathematical model of the general purpose computer i.e. Turing m/c is capable of performing any calculation which can be perform by any computing m/c.

*"Turing m/c is an ~~auto~~ autom/c whose temporary storage is tape. This tape is divided into ~~sates~~ cell each of which is capable of holding one symbol. This ~~stage~~ tape is infinite tape. The head which is associated with the tape can read or write a symbol & can move left to right or stay in same position". The turing m/c model is given below,

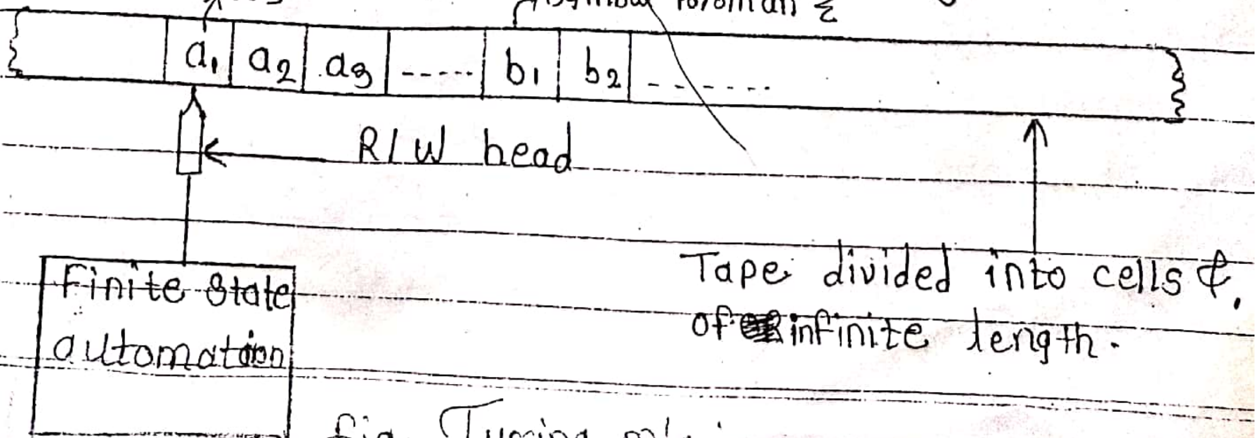


fig - Turing m/c.

In Turing m/c in one move depend~~ant~~s on a symb scan by the tape (read / write) head & the State of the finite control.

- Changes the state
- prints the symbol on the tape ~~s~~cell scan, replacing what was return their.
- Moves its head left or right one cell or revel in same ~~sta~~cell (no move).

* Formal Defn of Turing m/c :-

A Turing m/c 'M' is a 7-tuple as follows

$M = (Q, \Sigma, \Gamma, \delta, q_0, b, F)$

where,

$Q \rightarrow$ is a finite non-empty set of states

$\Sigma \rightarrow$ is a finite non-empty set of input symbols

$\Sigma \subseteq \Gamma$ and $b \notin \Sigma$.

$\Gamma \rightarrow$ is a finite non-empty set of tape symbols.

$\delta \rightarrow$ is a transition function mapping as,

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

$q_0 \rightarrow$ is a starting state, $q_0 \in Q$

$b \rightarrow$ is a blank symbol and $b \in \Gamma$

$F \rightarrow$ is a set of final states ~~+~~ $F \subseteq Q$.

* Language Acceptability By Turing M/c :-

A string 'w' in 'L' (Σ^*) is said to be accepted by turing m/c 'M' if '$q_0 w \vdash^* \alpha_1 p \alpha_2$' for some $p \in F$ & $\alpha_1, \alpha_2 \in \Gamma^*$.

Turing m/c does not accept 'w' if the m/c either halts in a non-accepting states or does not halt.

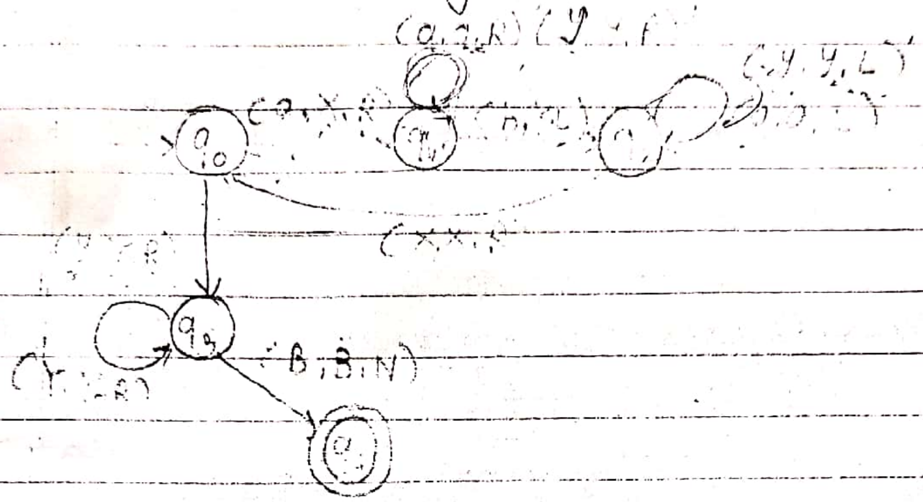
11 | 3 | 15

① Q- Check wheather the string is accepted string aaaabbbb. $a^n b^n$

⇒

...	B	a	a	a	a	b	b	b	b	B	...

Transition diagram is,



Transition Table :-

	a	b	X	Y	B
→ q_0	q_1XR	-	-	q_3YR	-
q_1	q_1aR	q_2YL	-	q_1YR	-
q_2	q_2aL	-	q_0XR	q_2YL	-
q_3	-	-	-	q_3YR	q_4BN
q_4	acceptance state.				

consider,

aaabbb ⊢ B $q_0$aaabbbB

⊢ BX$q_1$aaabbbB

⊢ BXa$q_1$abbbB

⊢ BXaa$q_1$bbbB

⊢ BXaq_2aYbbB

⊢ BX$q_2$aaYbbB

⊢ Bq_2XaaYbbB

⊢ BX$q_0$aaYbbB

⊢ BXXq_1aYbbB.

⊢ BXXaq_1YbbB

Scanned by CamScanner

$\vdash BXXaYq_1bbB$

$\vdash BXXaq_2YYbB$

$\vdash BXXq_2aYYbB$

$\vdash BXq_2XaYYbB$

$\vdash BXXq_0aYYbB$

$\vdash BXXXq_1YYbB$

$\vdash BXXXYq_1YbB$

$\vdash BXXXYYq_1bB$

$\vdash BXXXYq_2YYB$

$\cancel{\vdash BXXXYYq_2YB}$

$\cancel{\vdash BXXXYYYq_2B}$

$\vdash BXXXq_2YYYB$

$\vdash BXXq_2XYYYB$

$\vdash BXXXq_0YYYB$

$\vdash BXXXYq_3YYB$

$\vdash BXXXYYq_3YB$

$\vdash BXXXYYYq_3B$

$\vdash BXXXYYYq_4BN$

$\vdash BXXXYYYBN$

12 + 3 = 15

Representation Of Turing Machine :-

Turing m/c can be represented with,

① Instantaneous description,
② Transition diagram,
③ Transition table.

① Instantaneous Description :-

To define it is the current state & i/ string to be processed should be represented with the tape & FSA having read/write head. At the starting the current state is mostly the initial state of Turing m/c.

" An instantanous description of a turing m/c is a string α, β, δ where 'β' is a present state of 'M'. The entire i/p string is split as α, δ. The first symbol of 'δ' is the current symbol 'a' under read/write head. 'δ' has all the subsequent symbols of the i/p string & the string 'α' is the substring of the i/p string formed by all the symbol to the left of 'a'.

eg :-

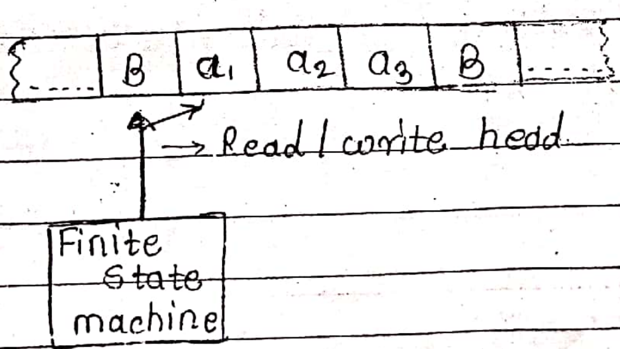


fig - Instantaneous diagram of turing m/c.

(α, β, δ)

The tape contains the string ~~$a_1 a_2 a_3$~~ $a_1 a_2 a_3$ of the turing m/c, the remaining cell are blanks.

& represented by 'B'. The position of the head represents the current state. The 'α' in the above diagram is 'B' & 'δ' is $a_2 a_3 B$.

② Transition Diagram :-

In transition diagram the states are represented by vertices joint by directed edges. Transition of state lables with the form (α, β, δ) where, $\alpha, \beta \in \Gamma$, and $\delta \in \{R, L, N\}$. It means that, $\delta(q_i, \alpha) = (q_j, \beta, \delta)$

eg :-

see the transition diagram of problem ① means $a^4 b^4$

③ Transition Table :-

The 'δ' which is use to represent transition function of turing m/c represented in the form of a table is called transition table, if "$\delta(q, a) = (\delta, \alpha, \beta)$", to represent α, β, δ of transition in transition diagram under 'a- column' & 'q^{th} row' where 'δ' is the resulting state of transition 'α' is the tape symbol to be return & 'β' is movement of head.

eg :-

see the transition of the language $L = \{a^4 b^4\}$ problem no. ①

14-3-15

Unit-5 → PDA

* **Deterministic PDA :-**

The PDA is deterministic which has atmost one move is possible from any instantaneous description.

Formal defn :-

① A PDA $M = (Q, \Sigma, \Gamma, \delta, Z_0, q_0, F)$ is deterministic if, for each q in Q & Z_0 in Γ, whenever $\delta(q, \epsilon, z)$ is non empty, then $\delta(q, a, z)$ is empty for all 'a' in Σ.

For no q in Q, z in Γ & 'a' in $\Sigma \cup \{\epsilon\}$ does $\delta(q, a, z)$ contains more than one element

* **Non-determinastic PDA :-**

The PDA is non-determinatic which has more than one moves are possible from any instantaneous description.

Formal defn :-

Let, $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ be a PDA. the set N(A) accepted by null store is defined by $N(A) = \{ w \in \Sigma^* / (q_0, w, Z_0) \vdash^* (q_0, \epsilon, \epsilon)$ for some $q \in Q \}$

15 3 15

CHAPTER-2

Imp* Difference :-

NFA	DFA
① NFA allows zero or more transition from state to state having i/p symbol from an alphabet and ϵ.	① DFA consist of a finite set of state having transⁿ from state to state that occure on an i/p symbol choosen from an alphabet,
② In NFA they may be zero or more outgoing transition for a ~~sy~~ i/p symbol from an alphabet & ϵ.	② ~~F~~for each i/p symbol their is exactly one transition out of each state.
③ 'δ' is a transition function mapping as, $\delta : Q \times \Sigma \longrightarrow 2^{Q}$	③ 'δ' is a transition function mapping as, $\delta : Q \times \Sigma \longrightarrow Q$
④ Every NFA is not always DFA.	④ Every DFA is always NFA.
⑤ NFA allows ϵ-moves	⑤ DFA has no ϵ-moves.