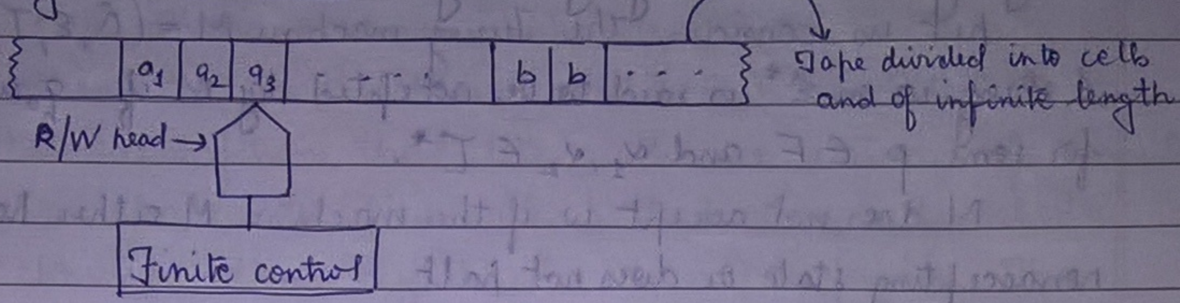
UNIT - 4

Turing Machines :-

① Turing Machine Model :-



A turing machine M is a 7-tuple, namely $(Q, \Sigma, \Gamma, \delta, q_0, b, F)$ where

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

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

$\Sigma \rightarrow$ non-empty set of input symbols

$\delta \rightarrow$ transition function $[\, (q, x)$ into $(q', y, D)\,]$

$q_0 \rightarrow$ initial state

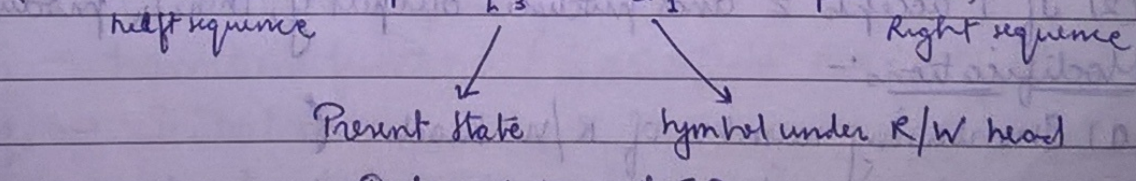
$F \rightarrow$ set of final states

$b \rightarrow$ blank.

② Representation of turing machines :-

(i) Instantaneous description (ID) :-

An ID of a Turing machine M is a string $\alpha\beta\gamma$, where β is the present state of M, the entire input string is split as $\alpha\gamma$, the first symbol of γ is the current symbol 'a' under the R/W head and γ has all the subsequent symbols of the input string, and the string α is the substring of the input string formed by all the symbols to the left of 'a'.



Representation of ID

(ii) Transition diagram (contains states and edges in the form of (q', y, D))

(iii) Transition table. (contains states and tape symbols)

③ **Language acceptability by turing machine :-**

Let us consider the Turing machine $M = (Q, \Sigma, \Gamma, \delta, q_0, F, b)$. A string w in Σ^* is said to be accepted by M if $q_0 w \vdash^* \alpha_1 p \alpha_2$ for some $p \in F$ and $\alpha_1, \alpha_2 \in \Gamma^*$.

M does not accept w if the machine M either halts in a nonaccepting state or does not halt.

④ **Design of Turing Machines:-**

~~(1) The~~ Basic guidelines for designing a turing machine:-

(i) The fundamental objective in scanning a symbol by the R/W head is to 'know' what to do in the future. The machine must remember the past symbols scanned. The turing machine can remember this by going to the next unique state.

(ii) The number of states must be minimized. This can be achieved by changing the states only when there is a change in the written symbol or when there is a change in the movement of the R/W head.

⑤ **Universal Turing machine and other modifications:-**

A universal turing machine (UTM) is a turing machine T_u that works as follows. It is assumed ~~that~~ to receive an input string of the form $e(T)e(z)$, where T is an arbitrary TM, z is a string over the input alphabet of T, and e is ~~the~~ an encoding function whose values are strings in $\{0,1\}^*$. The computation performed by T_u on this input string satisfies these two properties:-

(1) T_u accepts the string $e(T)e(z)$ if and only if T accepts z.

(2) If T accepts z and produces output y, then T_u produces output $e(y)$

Modifications:-

(1) Increasing number of R/W heads &

(2) Making the tape two or three dimensionals

(3) Adding a special purpose memory, such as 'STACKS' one special purpose registers.

⑥ Church's Hypothesis :-

Hypothesis means proposing certain facts. The church's hypothesis or church's turing thesis can be stated as,
"The assumption that the intuitive notion of computable functions can be identified with partial recursive functions."

However, this hypothesis can not be proved. The computability of recursive functions is based on following assumptions-
(1) Each elementry function is computable.
(2) Let f be the computable function and g be the another function which can be obtained by applying an elementry operation to f, then g becomes a computable function
(3) Any functions becomes computable if it is obtained by (1) & (2).

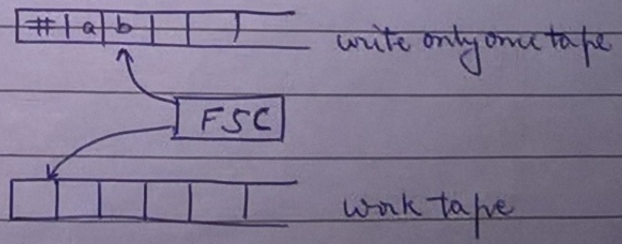
⑦ Composite and Iterated TM :-

When more than one TM combined together to solve the given problem. Such a TM is called composite TM.

The iterated TM is a special case of composite TM in which the output of a TM is given as input to itself.

⑧ Turing machine as enumerators :-

Enumerator is a turing machine with an output printer.
An enumerator E is a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_0, q_{print})$ where
$Q \rightarrow$ finite set of states
$\Sigma \rightarrow$ print tape alphabet
$\Gamma \rightarrow$ work tape alphabet
TWO TAPE DEVICE
$\delta \rightarrow$ transition function $(Q \times \Sigma \times \Gamma \rightarrow Q \times \Sigma \times \{L, R\} \times \Gamma \times \{L, R\})$
$q_0 \rightarrow$ Initial state
$q_{print} \rightarrow$ final state



(9) **Recursively Enumerable language:-**

A language $L \subset \Sigma^*$ is recursively enumerable if there exists a TM M, such that $L = T(M)$.

(10) **Recursive language:-**

A language $L \subseteq \Sigma^*$ is recursive if there exists some TM M that satisfies the following two conditions:-

(i) If $w \in L$ then M accepts w (that is, reaching an accepting state on processing w) and halts.

(ii) If $w \notin L$ then M eventually halts, without reaching an accepting state.

$$\boxed{\text{Recursive language} \subset \text{Recursive enumerable language}}$$

(11) **Properties of recursive & recursively enumerable language-**

(1) If L is a recursive language, then $\bar{L}$ is also a recursive language.

(2) If both L and $\bar{L}$ are recursively enumerable language then L and $\bar{L}$ is also recursive.