Subject Name: **Theory of Computation**

Subject Code: **IT-5001**

Semester: **5th**

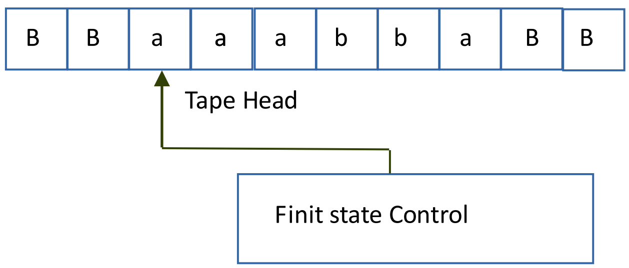
**IT-5001 Theory of computation**

**Unit-V : Turing Machine**

### 5.1 Introduction

Turing machine is considered as a simple model of a real computer. Turing machines can be used to accept all context-free languages, but also languages such as $L = \{a^m b^n c^{mn} : m \geq 0, n \geq 0\}$which is not class of language comes under regular and context free. Every problem that can be solved on a real computer can also be solved by a Turing machine.

### 5.2 Description of Turing Machine




1. There are k tapes, for some fixed k ≥ 1 of infinite length. Each tape is divided into cells, Each cell stores a symbol belonging to a finite set of tape symbols/ alphabets Γ. B is called blank symbol which also belongs to Γ.2. If a cell contains B, then this means that the cell is actually empty. (The given diagram is TM of single tape ).
2. Each tape has a tape head which can move along the tape, one cell per move. It can also read the cell it currently scans and replace the symbol in this cell by another tape symbol.
3. There is a finite state control, which can be in any one of a finite number of states. The finite set of states from Q. The set Q contains three special states: a start (initial ) state, an accept state, and a reject state.

### 5.3 Working or operation of Turing Machine

The Turing machine performs a sequence of computation steps. In one such step, it does the following:

1. Immediately before the computation step, the Turing machine is in a state q of Q, and tape heads is on a certain cell.
2. Depending on the current state q and the symbol that are read by the tape heads,
   (a) The Turing machine switches to a state p of Q (which may be equal to p),
   (b) Each tape head writes a symbol of Γ in the cell it is currently scanning (this symbol may be equal to the symbol currently stored in the cell), and
   (c) Each tape head either moves one cell to the left, moves one cell to the right, or stays at the current cell.

### 5.4 Formal Definition of Turing Machine

A Deterministic Turing machine is a 7-tuple

M = (Q,Σ, Γ, δ, q0, B, F )

where

1. Q is a finite set of states,
2. Σ is a finite set of input alphabet; the blank symbol B is not contained in Σ,
3. Γ is a finite set of tape alphabet; this alphabet also contains the blank symbol B, and Σ ⊆Γ,
4. δ is called the transition function, which maps: Q × Γ into Q × Γ × D .
   (q,X) into (p,Y,D)

Where q : current state
X : Tate symbol in scanning cell
p : Next state after transition
Y : Tape symbol written into the cell , replacing X
D : Direction in which head moves Left, Right or stay in scanning cell.

5. q0 is start state , element of Q
6. B is Blank sym δ bol element of Γ,
7. Set of final states which is subset of Q.

**5.5 Instantaneous Description of Turing Machine**

It is notation to current configuration of Turing Machine. IDs are string of following type.

$X_1 X_2 X_3$--------------- $X_{i-1}$ q $X_i X_{i+1}$ –-------------------- $X_n$

1. q is current state of Turing Machine.
2. $X_i$ symbol being scanned cell. Please note q is at left side of $X_i$.
3. $X_1 X_2$-------------- $X_n$ portion of tape between leftmost and rightmost blank.

Ids are used for displaying of computation sequence for input string.

**Note :** If head is to the left of leftmost blank then there will be some blank between q and first non-blank symbol.

q B B B $X_1 X_2 X_3$------------

Similarly if head is to the right of rightmost blank then tere will be some blank between q and last non-blank symbol.

$X_i X_{i+1}$ –-------------------- $X_n$ B B B q

**5.6 Computation by Turing Machine**

Consider TM T = ({$q_1$ $q_2$ $q_3$ $q_4$ $q_5$},{0,1},{0,1,b}, δ, $q_1$ , b, {$q_5$})
Transition Function δ is given by following transition table

| Present State | Tape Symbol | | |
|---|---|---|---|
| | B | 0 | 1 |
| → $q_1$ | ($q_2$ ,1, L) | ($q_1$ ,0, R) | - |
| $q_2$ | ($q_3$ ,b, R) | ($q_2$ ,0, L) | ($q_2$ ,1, L) |
| $q_3$ | - | ($q_0$ ,b, R) | ($q_5$ ,b, R) |
| $q_4$ | ($q_5$ ,0, R) | ($q_4$ ,0, R) | ($q_4$ ,1, R) |
| $q_5$ | ($q_2$ ,0, L) | - | - |

**Computation sequence for input string w = 00**

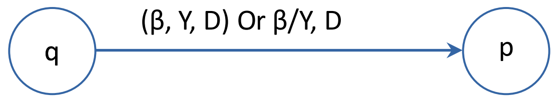
Initial ID : $q_1$00

$q_1$00 \|- 0$q_1$0 \|- 00$q_1$ \|- 0$q_2$01 \|- $q_2$001 \|- $q_2$b001 \|- $q_3$001 \|- $q_4$01 \|- 0$q_4$1 \|- 01$q_4$ \|- 010$q_5$ \|- 01$q_2$00 \|-*- $q_5$000

**5.7 Transition Digram of Turing Machine**

For transition function $\delta(q, \beta) = (p, Y, D)$ The transition digram will have




Students are expected to draw transition diagram of TM as given in section 5.6.

**5.8 Language accepted by Turing Machine**

Let M = (Q,Σ, Γ, δ, q0, B, F ) be a Turing Machine . A string w of input alphabet said to be accepted by TM if $q_0 w$ $\vdash^* \beta_1 p \beta_2$ for some p in F and $\beta_1, \beta_2$ are string in Γ and TM halt here.
String which accepted by TM belongs to Language recognized by TM .

String will be said not accepted by TM if TM either halt in non-accepting state or TM does not halt.

4.2.1
129
**5.9 Examples of Turing machines : Accepting palindromes using one tape**
**(Source : Introduction to Theory of Computation :Anil Maheshwari**
**Michiel Smid )**

We will show how to construct a Turing machine with one tape, that decides whether or not any input string w ∈ {a, b} ∗ is a palindrome. Recall that the string w is called a palindrome, if reading w from left to right gives the same result as reading w from right to left. Examples of palindromes are abba, baabbbbaab, and the empty string ǫ.

**Start of the computation:**

The tape contains the input string w, the tape head is on the leftmost symbol of w, and the Turing machine is in the start state q 0 .

**Idea:** The tape head reads the leftmost symbol of w, deletes this symbol and "remembers" it by means of a state. Then the tape head moves to the rightmost symbol and tests whether it is equal to the (already deleted) leftmost symbol.

- If they are equal, then the rightmost symbol is deleted, the tape head moves to the new leftmost symbol, and the whole process is repeated.
- If they are not equal, the Turing machine enters the reject state, and the computation terminates.

The Turing machine enters the accept state as soon as the string currently stored on the tape is empty.

We will use the input alphabet Σ = {a, b} and the tape alphabet Γ = {a, b, B}. The set Q of states consists of the following eight states:

$q_0$ : Start state; tape head is on the leftmost symbol
$q_a$ : Leftmost symbol was a; tape head is moving to the right
$q_b$ : Leftmost symbol was b; tape head is moving to the right
$q'_a$ : Reached rightmost symbol; test whether it is equal to a, and delete it
$q'_b$ : Reached rightmost symbol; test whether it is equal to b, and delete it
$q_2$ : Test was positive; tape head is moving to the left
$q_{accept}$ : Accept state
$q_{reject}$ : Reject state

The Turing Machine will be
M ({$q_0, q_a, q_b, q'_a, q'_b, q_2, q_{accept}, q_{reject}$},{a,b},{a,b,B}, δ , $q_0$ , B , { $q_{accept}$} )

The transition function δ is specified by the following instructions:

| | | |
|---|---|---|
| $q_0a \rightarrow q_aBR$ | $q_aa \rightarrow q_aaR$ | $q_ba \rightarrow q_baR$ |
| $q_0b \rightarrow q_bBR$ | $q_ab \rightarrow q_abR$ | $q_bb \rightarrow q_bbR$ |
| $q_0B \rightarrow q_{accept}$ | $q_aB \rightarrow q'_aBL$ | $q_bB \rightarrow q'_bBL$ |
| $q'_aa \rightarrow q_2BL$ | $q'_ba \rightarrow q_{reject}$ | $q_2a \rightarrow q_2aL$ |
| $q'_ab \rightarrow q_{reject}$ | $q'_bb \rightarrow q_2BL$ | $q_2b \rightarrow q_2bL$ |
| $q'_aB \rightarrow q_{accept}$ | $q'_bB \rightarrow q_{accept}$ | $q_2B \rightarrow q_0BR$ |

Studnets should go through the computation of this Turing machine for some sample inputs, for example abba, b, abb and the empty string (which is a palindrome).

**5.10 Examples of Turing machines : Accepting palindromes using two tapes**
**(Source : Introduction to Theory of Computation :Anil Maheshwari**
**Michiel Smid )**

We again consider the palindrome problem, but now we use a Turing with two tapes.

**Start of the computation:**

The first tape contains the input string w and the head of the first tape is on the leftmost symbol of w. The second tape is empty and its tape head is at an arbitrary position. The Turing machine is in the start state $q_0$ .

**Idea:**

First, the input string w is copied to the second tape. Then the head of the first tape moves back to the leftmost symbol of w, while the head of the second tape stays at the rightmost symbol of w. Finally, the actual test starts: The head of the first tape moves to the right and, at the same time, the head of the second tape moves to the left. While moving, the Turing machine tests whether the two tape heads read the same symbol in each step.

The input alphabet is Σ = {a, b} and the tape alphabet is Γ = {a, b, B}.

The set Q of states consists of the following five states:

$q_0$ : Start state; copy w to the second tape

$q_1$ : w has been copied; head of first tape moves to the left

$q_2$ : Head of first tape moves to the right; head of second tape moves to the left; until now, all tests were positive

$q_{accept}$ : Accept state

$q_{reject}$ : Reject state

The Turing Machine will be

$M = (\{q_0, q_1, q_2, q_{accept}, q_{reject}\},\{a,b\},\{a,b,B\}, \delta, q_0, B, \{q_{accept}\})$

The transition function δ is specified by the following instructions:

| | | |
|---|---|---|
| $q_0aB \rightarrow q_0aaRR$ | $q_0bB \rightarrow q_0bbRR$ | $q_0BB \rightarrow q_1BBLL$ |
| $q_1aa \rightarrow q_1aaLN$ | $q_1ab \rightarrow q_1abLN$ | $q_1ba \rightarrow q_1baLN$ |
| $q_1bb \rightarrow q_1bbLN$ | $q_1Ba \rightarrow q_2BaRN$ | $q_1Bb \rightarrow q_2BbRN$ |
| $q_1BB \rightarrow q_{accept}$ | $q_2aa \rightarrow q_2aaRL$ | $q_2ab \rightarrow q_{reject}$ |
| $q_2ba \rightarrow q_{reject}$ | $q_2bb \rightarrow q_2bbRL$ | $q_2BB \rightarrow q_{accept}$ |

Students should run this Turing machine for some sample inputs.

**5.10 Examples of Turing machines : Accepting $a^nb^nc^n$ using one tape**
**(Source : Introduction to Theory of Computation :Anil Maheshwari**
**Michiel Smid )**

We will construct a Turing machine with one tape that accepts the language $\{a^nb^nc^n : n \geq 0\}$.

Please note that this language is not context- free.

**Start of the computation:**

The tape contains the input string w, and the tape head is on the leftmost symbol of w. The Turing machine is in the start state $q_0$ .

**Idea:**

Repeat the following Stages 1 and 2, until the string is empty.

**Stage 1.** Walk along the string from left to right, delete the leftmost a,delete the leftmost b, and delete the rightmost c.

**Stage 2.** Shift the substring of b and c one position to the left; then walk back to the leftmost symbol.

The input alphabet is Σ = {a, b, c} and the tape alphabet is Γ = {a, b, c, B}

For Stage 1, we use the following states:

$q_0$ : Start state; tape head is on the leftmost symbol
$q_a$ : Leftmost a has been deleted; have not read b
$q_b$ : Leftmost b has been deleted; have not read c
$q_c$ : Leftmost c has been read; tape head moves to the right
$q'_c$ : Tape head is on the rightmost c
$q_1$ : Rightmost c has been deleted; tape head is on the rightmost symbol or B
$q_{accept}$ : Accept state
$q_{reject}$ : Reject state

The transitions for Stage 1 are specified by the following Transition Functions:

| | | | |
|---|---|---|---|
| $q_0a \rightarrow q_aBR$ | $q_0b \rightarrow q_{reject}$ | $q_0c \rightarrow q_{reject}$ | $q_0B \rightarrow q_{accept}$ |
| $q_aa \rightarrow q_aaR$ | $q_ab \rightarrow q_bBR$ | $q_ac \rightarrow q_{reject}$ | $q_aB \rightarrow q_{reject}$ |
| $q_ba \rightarrow q_{reject}$ | $q_bb \rightarrow q_bbR$ | $q_bc \rightarrow q_ccR$ | $q_bB \rightarrow q_{reject}$ |
| $q_ca \rightarrow q_{reject}$ | $q_cb \rightarrow q_{reject}$ | $q_cc \rightarrow q_ccR$ | $q_cB \rightarrow q'_cBL$ |
| $q'_cc \rightarrow q_1BL$ | | | |

For Stage 2, we use the following states:

$q_1$ : As above; tape head is on the rightmost symbol or on 2
$q^c$ : Copy c one cell to the left
$q^b$ : Copy b one cell to the left
$q_2$ : done with shifting; head moves to the left

Additionally, we use a state $q'_1$ which has the following meaning: If the input string is of the form $a^i$ bc, for some i ≥ 1, then after Stage 1, the tape contains the string $a^{i-1}$ BB, the tape head is on the B immediately to the right of the a, and the Turing machine is in state $q_1$ . In this case, we move one cell to the left; if we then read B, then i = 1, and we accept; otherwise, we read a, and we reject.

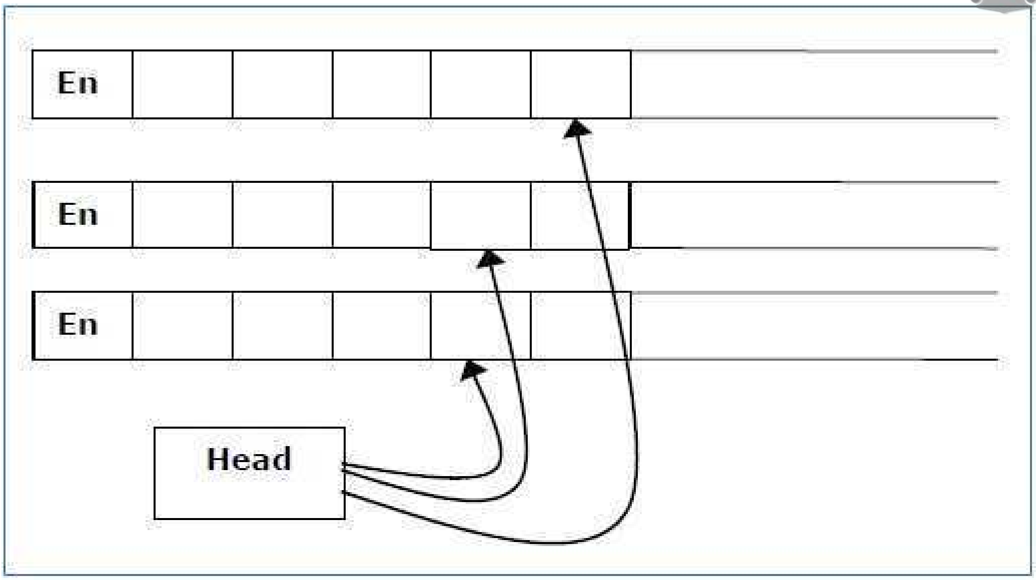
The transitions for Stage 2 are specified by the following instructions:

| | | |
|---|---|---|
| $q_1a \rightarrow$ cannot happen | $q_1b \rightarrow q_{reject}$ | $q_1c \rightarrow q^cBL$ |
| $q_1B \rightarrow q'_1BL$ | $q'_1a \rightarrow q_{reject}$ | $q'_1b \rightarrow$ cannot happen |
| $q'_1c \rightarrow$ cannot happen | $q'_1B \rightarrow q_{accept}$ | $q^ca \rightarrow$ cannot happen |
| $q^cb \rightarrow q^bcL$ | $q^cc \rightarrow q^ccL$ | $q^cB \rightarrow q_{reject}$ |
| $q^ba \rightarrow$ cannot happen | $q^bb \rightarrow q^bbL$ | $q^bc \rightarrow$ cannot happen |
| $q^bB \rightarrow q_2bL$ | $q_2a \rightarrow q_2aL$ | $q_2b \rightarrow$ cannot happen |
| $q_2c \rightarrow$ cannot happen | $q_22 \rightarrow q_0BR$ | |

**5.11 Variants of TM : Multi-tape TM**

Multi-tape Turing Machines have multiple tapes where each tape is accessed with a separate head. Each head can move independently of the other heads. Initially the input is on tape 1 and others are blank. At first, the first tape is occupied by the input and the other tapes are kept blank. Next, the machine reads consecutive symbols under its heads and the TM prints a symbol on each tape and moves its heads.




A Multi-tape Turing machine can be formally described as a 7-tuple

M = (Q,Σ, Γ, δ, q0, B, F )

where −

Q is a finite set of states

Σ is set of input alphabet

Γ is the tape alphabet

B is the blank symbol

δ is a relation on states and symbols where

$\delta: Q \times X^k \rightarrow Q \times (X \times \{Left\_shift, Right\_shift, No\_shift\})^k$

where there is **k** number of tapes

$q_0$ is the initial state

F is the set of final states

**Note** − Every Multi-tape Turing machine has an equivalent single-tape Turing machine.

**5.12 Variants of TM : NDTM**

In a Non-Deterministic Turing Machine, for every state and symbol, there are a group of actions the TM can have. So, here the transitions are not deterministic. The computation of a non-deterministic Turing Machine is a tree of configurations that can be reached from the start configuration.

An input is accepted if there is at least one node of the tree which is an accept configuration, otherwise it is not accepted. If all branches of the computational tree halt on all inputs, the non-deterministic Turing Machine is called a **Decider** and if for some input, all branches are rejected, the input is also rejected.

A non-deterministic Turing machine can be formally defined as a 7-tuple (Q, ∑, Γ, δ, $q_0$, B, F) where −

**Q** is a finite set of states

Γ is the tape alphabet

∑ is the input alphabet

**δ** is a transition function;

$\delta : Q \times X \rightarrow P(Q \times X \times \{Left\_shift, Right\_shift\})$.

$\mathbf{q_0}$ is the initial state

**B** is the blank symbol

**F** is the set of final states

**5.13 Universal Turing Machines**