TOP SHEET

# FUTURE INSTITUTE OF TECHNOLOGY

CONTINUOUS ASSESSMENT (CA) – ODD / EVEN SEMESTER 202___

THEORY PAPER (CA – 1 / 2 / 3 / 4)

(Please tick the applicable one)

NAME OF THE STUDENT Anirban Manna

DEPT. C.S.E SEM 4th UNIVERSITY ROLL NO. 34200121045

UNIVERSITY REGN. NO. 213420106110015 OF 2021-22

PAPER NAME Formal Language & Automata Theory

PAPER CODE PCC-CS403 DATE OF TEST 2 10/03/2023

FULL MARKS 25 SIGNATURE OF THE STUDENT Anirban Manna

S. Chakraborty

SIGNATURE OF THE TAGGED TEACHER

# Basic Notations & Representation in Automata Theory


## Abstract :

The theory of automata is a theoretical branch of computer science and mathematics. It is developing a machine which processes input towards the output. The study of abstract machines and the computation problems that can be solved using some basic notations or terminologies called "Automatons". Automatons are abstract models of machines that perform computations on input by moving through a series of states or configurations, at each stages of the computation. Here we will know about those basic notations and their representation.


## 1) Introduction:

Automata theory (also known as Theory of Computation) is a theoretical branch of Computer Sc. and Mathematics, which mainly deals with the logic of computation with respect to simple machines, referred to as automata.

It enables scientists to understand how machines compute functions and solve problems. The Automata can be represented with some basic terminologies, which are important and frequency used in the Theory of computation.

## 2) Problem definition :

Automata Theory is an exciting theoretical branch of comuter science. It established its roots during the 20th century, as mathematicians began developing - both theoretically and literally - machines that limited certain features of man, comp leting calculations are more quickly and reliably. So, it is developing a machine which Processing input towards output. There are 2 types of Automata by their output: (i) Finite Automata.
(ii) Infinite Automata.

Finite Automata is the simplest machine to recognize Patterns. The finite automata in an abstract machine that has five elements or tuples. It has a set of states and rules for moving from one state to another depending upon the input symbol. Basically, it is an abstract model of digital computer.

Finite Automata is characterized into two types:

1. DFA (Deterministic Finite Automata): DFA refers to the uniqueness of the compulation. The finite automata are called deterministic finite automata if the machine is read and input string one symbol at a time.

2) NFA: NFA stands for non-deterministic finite automata. It is easy to construct an NFA than DFA for a given regular language. The finite are called NFA when there exist many paths for specific input from the current state to the next state.

We will know about those Automata and also their representations.

## 3. Discussion:

This automata consists of states and transitions. The state is represented by circles, and the Transitions is represented by arrows. Automata is the kind of machine which takes some string as input and this input goes through a finite number of states and may enter in the final state. There are some basic terminologies that are important and frequently used in automata.

- Symbol: Symbols are an entity or individual objects, which can be any letter, alphabet or any picture. It is the smallest building block, which can be any alphabet, letter or picture.

  Example: a, b, c, d
  1, 0
  #, a, b

☐ **Alphabets:** Alphabets is collection of symbol. this finit set of symbols is denoted by $\Sigma$.

Example : $\Sigma = \{0, 1\}$ is an alphabet of binary digit

$\Sigma = \{a, b, c, \ldots z\}$ etc.

☐ **String:** String is collection of symbols over alphabet. The string is denoted by w, and the length of a string is denoted as $|w|$

Example:

$\Sigma = \{a, b\}$

Length of the string $|W| = 2$

∴ string = a, b, aa, bb, ab, ba (W)

☐ **Language -** Language is collection or set of all strings over Alphabet. It is formed over alphabet can be Finite or Infinite both. Represented as 'L'.

Example : Finite Language L = {set of string length = 2}

$L = \{aa, bb, ba, ab\}$

Infinite Language = L = {set of strings end with 'a'}

$L = \{a, ba, bba, baa, aaa, aba \ldots\}$

☐ **Kleene star:** Kleene star is represented with $\Sigma^*$, It is a universal set. It means it can be a infinite set.

$\Gamma \Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \ldots\ldots$

Example: Let's concider a alphabet $\Sigma = \{a, b\}$

$\Sigma^0$ = length of string be 0. Now this is called Null string or empty string.

This is represented as $\{\epsilon\}$

$\Sigma^1$ = Lenght of string must be 1

strings = $\{a, b\}$

$\Sigma^2$ = length of string must be 2

strings = $\{ab, ba, aa, bb\}$

$$\Sigma^* = \{\epsilon, a, b, aa, ba, bb, ab, \ldots\}$$

□ Kleene closure: It represented as '$\Sigma^+$'

$$\therefore \Sigma^+ = \Sigma^* - \{\epsilon\}$$

$$\Sigma^+ = \Sigma^* - \Sigma^0$$

$$\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \ldots\ldots$$

Example: Same as previous but $\epsilon$ will not added

$$\therefore \Sigma^+ = \{a, b, ab, aa, ba, bb, \ldots\ldots\}$$

Finite Automata: Build a model in Finite number states is called Finite automate. It divided into

(i) DFA (ii) NFA

By Default FA will be considered as DFA.

Finite automata can be defined and constructed with 5 Tuples or Tuple with 5 elements, with the help of states.

- State: Here the state is represented as circle with state name. Ex:- (q0)

- Transition - Edge between tow statees and this has directed edge.

  Example: 


- Initial state - state with an edge from No source.

  Example: 


- Final state - It is represented as double circle with state name.

  Example: 

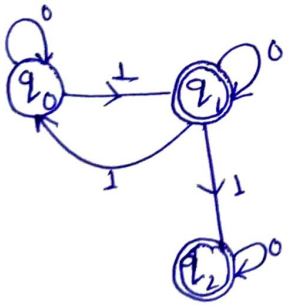

A process will change this one state to another state for every input symbol own the alphabet.

- FA will be having only one Initial state and multiple Final state. And each state have a way of all alphabets.

FA is represented in two ways by using—

(1) Directed graph with states

(2) Transition table.

1) 


2) Transition Table

| | 0 | 1 |
|---|---|---|
| $q_0$ | $q_0$ | $q_1$ |
| $q_1$ | $q_1$ | $q_0$ |
| $q_2$ | ..... | |

Finite Automata - Tuple of 5 elements.

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

Let's know them in short—

$Q$ - Set of all states which are present in Transition Diagram.

$\Sigma$ - Input symbol

$\delta$ - Transition Function (For which i/p the state change from one to another).

$q_0$ - Initial state.

F = Set of Final state.

(So, if $q_0$ is a state and $q_0$ also having the transition for both the symbols 0 and 1)

4) Result Analysis — Finite Automata is considered as DFA means it is also same for DFA but NFA it is diffrent. Only the transition function of NFA is not the same as DFA, Other Tuples are same.

Transition Function (For DFA)

$$\delta(dou) \longrightarrow Q \times \Sigma - Q$$

Transition Function (For NFA).

$$\delta(dou) \longrightarrow Q \times \Sigma - 2^Q$$

5. Bibliography


- Classroom Notes (2)
- https://geekforgeeks.org/
- https://tutorialspoint.com/
- Theory of Computer Science by KLP Mishra, N. Chandrasekaran.