Theory of Computation

It is a branch of computer science that ~~dealt~~ deals with that how efficiently ~~prob~~ ~~a~~ problem can be solved on a model of computation using an algorithm.

Theory of Computation

- Automata Theory and Language
- Computability Theory
- Complexity

1. Automata Theory and Language —

It deals with the definition and ~~problem~~ properties of various mathematical model of computer.

e.g. Finite ~~ar~~ Automal
Context Free Grammar
Turing Machine.

2. Computability Theory —

It deals with what can and cannot be computed by the model.

DATE: / /
PAGE:

(3) Complexity Theory —

It ~~grou~~ makes group with computable proof based on their hardness.

Basic Definitions

1. Symbols :

Symbols are an entity or individual objects which can be any letter, alphabet or any picture.

e.g. a, b, c . . . . z
0, 1, 2 9
+ , − , * , % . . . . special character

2. Alphabets :

Alphabets are a finite set of symbols. It is denoted by Σ.

e.g. $\Sigma = \{0, 1\}$ set of binary alphabets

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

$\Sigma = \{A, B, C, D\}$

$\Sigma = \{0, 1, 2\}$

$\Sigma = \{0, 1 \cdots 5\}$ etc

$\Sigma = \{\#, \beta, \Delta\}$

I/P → think → Yes (accept)
→ No (Reject)

FSM → Finite State Machine
↓
simple model ~~of~~ of computation.
very small memory

CFL → Content Free Language
↓
set of strings

Undecidable → The problem that can not be solve mechanically.

Lecture 2 :
Prerequisites

symbol : Anything like a, b, c, 0, 1, 2 ...

Alphabet :→ Denoted by Σ
→ It is a collection of symbols
e.g. {a, b}, {d, e, f, g}
{0, 1, 2} ....

DATE: / /
PAGE:

String :→ A seq. of symbols

e.g. a, b, 0, 1, aa, bb, ab, 01, ...

Language — set of strings

Eg: $\Sigma = \{0, 1\}$

exmp.1 → L_1 = set of all strings of length 2.

Finite set because of finite no. of elements = $\{00, 01, 10, 11\}$

examp 2 → L_2 = set of all strings of length 3

$= \{000, 001, 010, 011, 100, 101, 111, 110\}$

L_3 = set of all strings that begin with '0'

$= \{0, 00, 01, 000, 001, 010, 011, 0000 \ldots\}$

infinite no. of elements

Infinite set

Powers of Σ : $\Sigma = \{0, 1\}$ → alphabets.

Σ^0 = set of all strings of length '0'

$\therefore \Sigma^0 = \{\epsilon\}$ → cardinalinality $2^0 = 1$

Scanned by TapScanner

DATE: / /
PAGE:

Σ^1 = set of all strings of length

$\therefore \Sigma^1 = \{0, 1\}$ → cardinality 2 $\quad 2^1 = 2$

Σ^2 = Set of all strings of length

$\therefore \Sigma^2 = \{00, 01, 10, 11\}$ → cardinality

Σ^3 = set of all strings of length

$\therefore \Sigma^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$ $\quad 2^3 = 8$ cardinality

⋮

Σ^n = set of all strings of length n

→ cardinality 2^n

<u>Cardinality</u>: Number of elements in a set

→ $\Sigma^n = 2^n$

$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \ldots$

$= \{\epsilon\} \cup \{0, 1\} \cup \{00, 01, 10, 11\} \cup \ldots$

<u>s~~et of pos~~</u>

set of all possible strings of all leng over $\{0, 1\}$

↳ Infinite ~~state~~ ~~set~~ state

$$\Sigma^* = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2 \cup \Sigma_2 \cup \ldots\ldots$$

e.g.

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

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

Kleen Closure / Plus

The set Σ^+ is the infinite set of all possible strings of all possible lengths over Σ ~~except~~ excluding λ.

$$\Sigma^+ = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2 \cup \ldots$$

$$\Sigma^+ = \Sigma^* - \{\lambda\}$$

e.g. If $\Sigma = \{a, b\}$

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

Language

A language is a subset of Σ^* for some alphabet Σ. It can be finite or infinite.

e.g. If the language takes all possible strings of length 2 over $\Sigma = \{a, b\}$, then $L = \{ab, bb, ba, bb\}$

Deterministic & Non-Deterministic (FA) Finite Automaton

→ Deterministic (DFA) Finite Automaton

→ Non " " " (NDFA

classification

DATE: / /
PAGE:

Alphabet

An alphabet is any finite set of symbols.

e.g. $\Sigma = \{a, b, c, d\}$ is an alphabet set where a, b, c and d are alphabets.

String

A string is a finite sequence of symbols taken from Σ.

e.g. 'cacad' is a valid string on the alphabet set where $\Sigma = \{a, b, c, d\}$

Length of the string:

It is the number of symbols present in a string (Denoted by $|S|$)

e.g. if $S =$ 'cabcad', $|S| = 6$
if $|S| = 0$, it is called an empty string (Denoted by λ or

Kleen Star

The set Σ^* is the infinite set of all possible strings of all possible length over Σ including λ.

3. <u>String</u>:

It is a finite collection of symbols from the alphabet. The string is denoted by 'w'.

e.g. abab, 01101 etc.

<u>Finite Automata</u> (FA)

Finite Automata is an abstract computing device. It is a mathematical model of a system with discreate inputs, outputs, states and set of transitions from state to state that occurs on input symbols from alphabet Σ.

It represents:

→ Graphical (Transition Diagram or Transition Table)

→ Tabular (Transition Table)

→ Mathematical (Transition function ~~and~~ or Mapping function)

DATE: / /

PAGE:

Definition of Finite Automata :

A finite automata is a 5 tuples set ; they are

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

where

$Q \rightarrow$ set of all states

$\Sigma \rightarrow$ ~~set~~ ~~of~~ finite set of alphabets.

$\delta \rightarrow$ Transition function

$$Q \times \Sigma \rightarrow Q$$

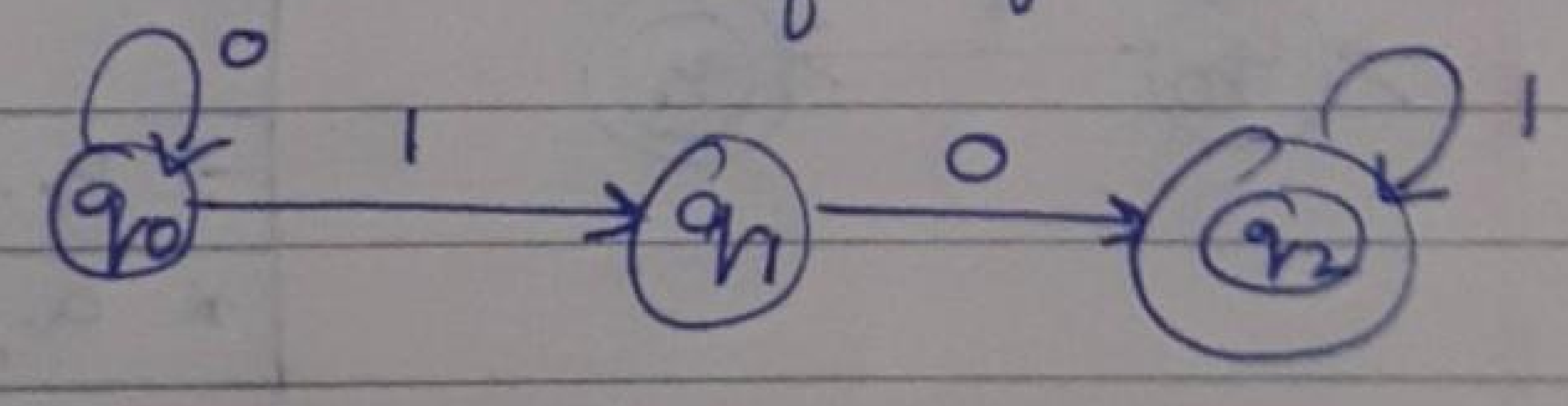
$q_0 \rightarrow$ Initial state $q_0 \in Q$

F $\rightarrow$ Final state $F \subseteq Q$

Transition Diagram :

It is a directed graph associated with the vertices of the graph corresponds to the state of finite automata.

e.g.



$\{0, 1\}$ are inputs

$q_0 \rightarrow$ Initial state

$q_1 \rightarrow$ Intermediate state

$q_2 \rightarrow$ Final state

DATE: / /
PAGE:

Transition Table :

It is basically a tabular representation, of the transition function that takes two arguments (a state and a symbol) and returns a value (the "next state").

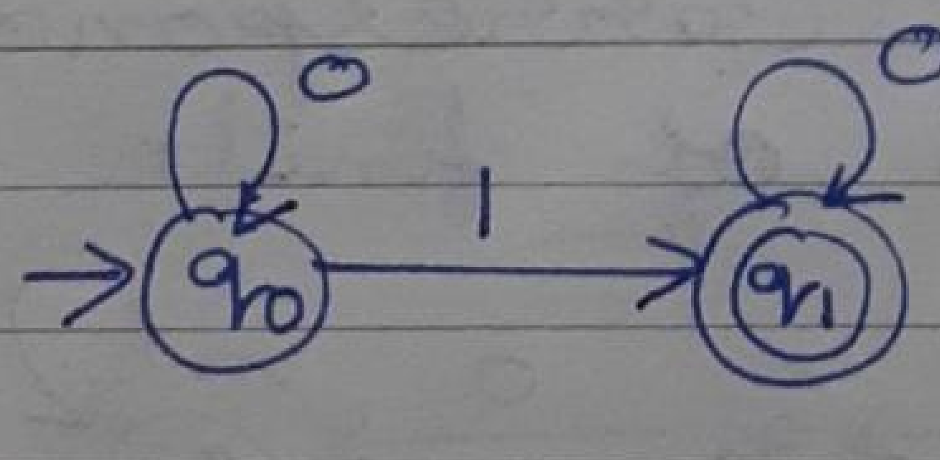
→ Rows corresponds to states

→ Columns corresponds to i/p ~~stat~~ symbols.

→ Entries corresponds to next states.

→ The start state is marked with an arrow. (→)

→ The accept / final state are marked with star (*)



	0	1
→ q_0	q_0	q_1
* q_1	q_1	q_1

DATE: / /
PAGE:

Transition function :

→ The mapping ~~ϕ~~ function or transition function denoted by δ.

→ Two parameter are passed to this transition function.
 i) Current state &
 ii) Input symbol.

→ The transition function returns a state which can be called as next state.

[δ (current-state, current-input-symbol) = next-state]

e.g. $\delta(q_0, a) = q_1$
or
$\delta(q_0, 1) = q_1$

$Q \times \Sigma \rightarrow Q$

Scanned by TapScanner

DATE: / /

PAGE:

Finite state Machines

- Finite Automata
  - FA with output
    - Moore Machine
    - Mealy Machine
  - FA without output
    - DFA
    - NFA
    - ϵ-NFA

DFA – Deterministic Finite Automata

- It is the simplest model of computation
- It has a very limited Memory

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

Q = set of all states

Σ = inputs

q_0 = start state/initial state

F = set of final states

δ = transition function from $Q \times \Sigma \rightarrow$

~~Examples~~

DATE: / /
PAGE:

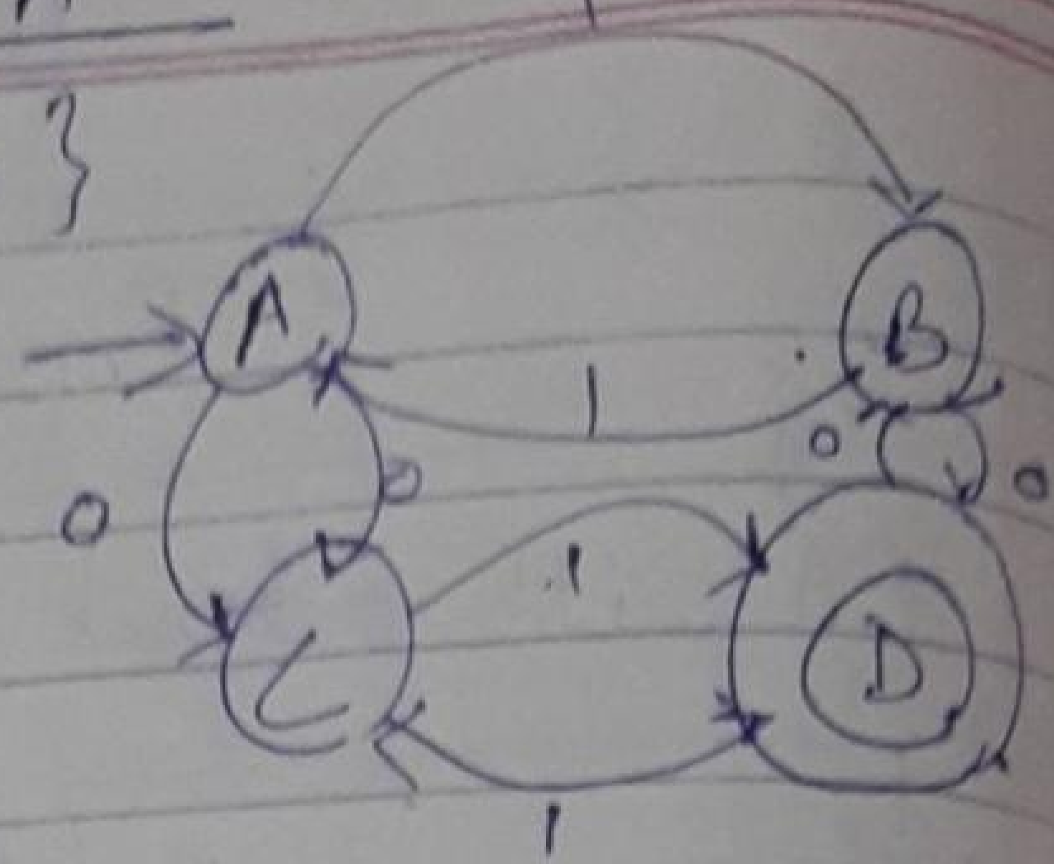
Basic Structure of DFA

$Q = \{A, B, C, D\}$

$\Sigma = \{0, 1\}$

$q_0 = A$

$F = D$

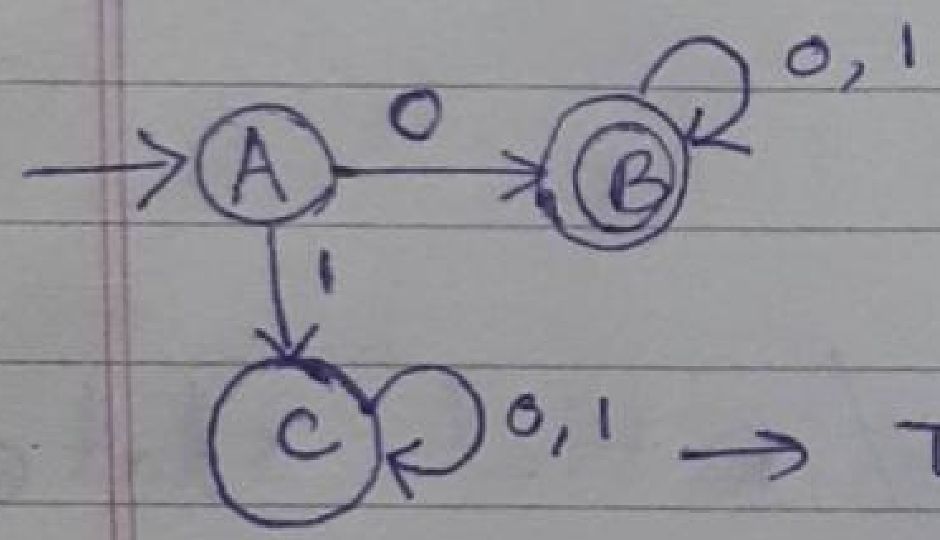


	0	1
A	C	B
B	D	A
C	A	D
D	B	C

Example 1

L_1 = set of all strings that start with 0

= $\{0, 00, 01, 000, 010, 011, 0000, \dots\}$



→ This state cannot reached to the final state.

Dead state
or
Trap state

Eg 001 → ~~Reject~~ Accepted string

Initial state $\rightarrow A \xrightarrow{0} B \xrightarrow{0} B \xrightarrow{1} B$

Final state

Eg 101 $A \xrightarrow{1} C \xrightarrow{0} C \xrightarrow{1} C$

↓ Rejected string

Not final state

DFA (Deterministic Finite Automata)

→ The FA are called DFA if the machine is read an i/p string one symbol at a time.

→ Deterministic refers to the uniqueness of the computation.

→ In DFA, there is only one path for specific i/p from the current state to the next state.

→ DFA does not accept the null move, i.e. DFA cannot change state without any i/p character.