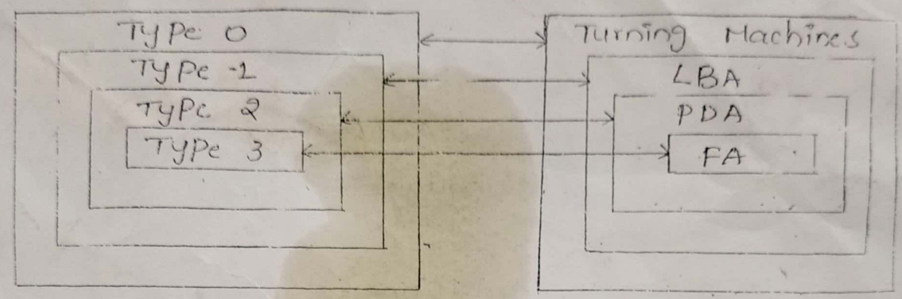
Unit I

Chomsky hierarchy of Grammars :-

Chomsky is a Scientist. He capi categorizes grammers into 4 types :



*FA - Finite Automata

*PDA - Push Down Automata.

*LBA - Linear Bound Automata.

Type 0 :- Type 0 is also called unrestricted Grammar (or) recursively enumarable Grammar. The name denotes it is unrestricted. means there are no restrictions imposed on LHS and RHS of any production.

Production :- It is in The form of LHS → RHS. whenever The LHS is present in The RHS of any production, Then its respected RHS is substituted.

Example :- $Sa \rightarrow aa$
$Aab \rightarrow absa$
$Aab \rightarrow abaa$.

Date 29/11/17

Formal Languages And Automata Theory.

Unit - I :- Fundamentals :-

Strings, Alphabet, language, operations, finite state Machine, definitions, finite Automaton model, Acceptance of strings and languages, deterministic finite Automaton and non-deterministic finite Automaton, Transition diagrams and language recognizers.

Finite Automata :-

NFA with E transitions - significance, Acceptance of languages.

Conversions And Equivalence :-

Equivalence between NFA with and without E transitions, NFA to DFA conversion, Minimization of FSM, Equivalence between two FSM's, Finite Automata with output - moore and Mealy Machines.

Eg: consider The Languages

1) $L_1 = \{a, ab, abb, abbb, \ldots\}$

2) $L_2 = \{\epsilon, ab, aabb, aaabbb, \ldots\}$

union:- $L_1 \cup L_2 = \{\epsilon, a, abb, abbb \ldots,$
$aabb, aaabbb \ldots\}$.

Intersection:- $\{ab\}$

Difference:- $\{\epsilon, a, abb, abbb, \ldots,$
$aabb, aaabbbb \ldots\}$

concatenation:-

* $L_1 = \{good, ~~girl~~, bad\}$
  $L_2 = \{girl, boy\}$
  $L_1 \cdot L_2 = \{$good girl, good boy, bad girl, bad boy

→ $L_2 \cdot L_1 = \{$girl good, girl bad, boy good, boy bad

Note1:- $\left.\begin{array}{l}\epsilon \cdot A = A \\ \epsilon \cdot B = B\end{array}\right\}$ in concatenation.

Note2:-
$\{\epsilon\} \cup \underbrace{\{a, aa, aaa \ldots}_{\Sigma^+}\} = \{\underbrace{\epsilon, a, aa, aaa \ldots}_{\Sigma^*}$

$$\Sigma^* = \{\epsilon\} \cup \Sigma^+.$$

6/12/17 Eg 2:- $L_1 = \{a, ab, abb\}$
$L_2 = \{ab, abab\}$

* $L_1 \cup L_2 = \{a, ab, abb, abab\}$
* $L_1 \cap L_2 = \{ab\}$
* $L_1 - L_2 = \{a, abb, abab\}$.
* $L_1 \cdot L_2 = \{aab, aabab, abab, ababab, abbab,$
$abbabab\}$.

Eg3:- $L_1 = \{\epsilon, a\}$
$L_2 = \{a, ab, abb\}$.

* $L_1 \cup L_2 = \{\epsilon, a, ab, abb\}$
* $L_1 \cap L_2 = \{a\}$
* $L_1 - L_2 = \{\epsilon, ab, abb\}$.
* $L_1 \cdot L_2 = \{a, ab, abb, aa, aab, aabb\}$.

Operations on strings:-

* concatenation :-
* transpose
* palindrome.

concatenation:-

* Let S_1 and S_2 are two strings. Then concatenation of two strings is $S_1 \cdot S_2$

$S_1 =$ good $\quad S_2 =$ boy

$S_1 \cdot S_2 =$ good boy.

transpose:-

* Let S is a string Then The transpose of The string 'S' is denoted as S^T. It is also called reverse of a string.

i.e, $(aS)^T = S^T a$.

Eg:- $(abb)^T =$ bba.

palindrome:-

* The reverse of a string is equal to The original string.

$$S^T = = S$$

Eg:- $S =$ liril
$S^T =$ liril

S equal to S^T. so we conclude S is palindrome.

48MP AI QUAD CAMERA
Shot by VikramKrishnaYadav

The Theory of Automata :- An Automation is defined as a system where energy, materials and information are transform transmitted and used for performing som functions without direct participation of human.

* In computer Science, The term Automata means discrete Automata and is defined as shown below.

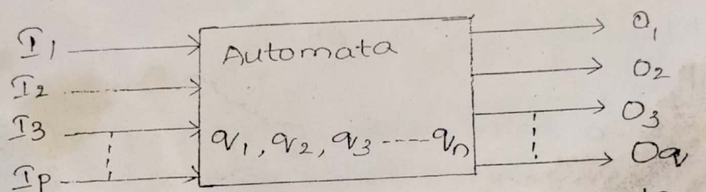


fig:- Block diagram of Automata.

* Characteristics :-

* Its characteristics are :-

* Input :- At each of discrete instance of time t_1, t_2 ------ t_n, The input values I_1, I_2, I_3, ------ I_p each of which can take a finit number of fixed values from input Alpha bet Σ.

* Output :- O_1, O_2, O_3 ----- O_q are The output of the model each of which can take finit number of fixed values.

State :- At Any instant of time, The Automata can be in one of The states, Q_1, Q_2, Q_3 ------ Q_n.

Scanned with CamScanner

State Relation:- The next state of an Automata at any instant of time is determined by the present state and the present input.

output Relation:- The output is Related to either state only to both The input and state.

Description of Finite Automata:-

Analytically, A finite Automata can be represented by a 5-tuple

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

where,

Q is a finite & non-empty set of states.

Σ is a finite non-empty set of inputs.

δ is a mapping function (or) transition function.

$$\boxed{\delta : Q \times \Sigma \rightarrow Q}$$

i.e, δ mapps $Q \times \Sigma$ into Q.

$q_0 \in Q$, is the initial state.

F subset Q ($F \subseteq Q$) is a set of final state.

* There is only one initial state and no. of final states.

Block diagram of Finite Automata:-

input symbols

Input tape:- It is a linear tape having some number of cells. Each input symbol is placed in each cell.

Finite control:- It decides the next states on receiving particular input for input tape.

Reading Head:- It examines (or) scans the current input symbol if the input is processed Then it moves towards the next input.

End Markers:- End markers denote the begining and ending of the input string.

7/12/17 **Types of Finite Automata:-**

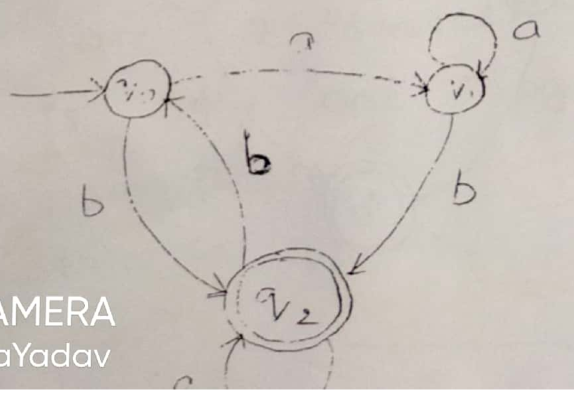
Finite Automata

- Deterministic Finite Automata
- Non-Deterministic Finite Automata.

Deterministic Finite Automata (DFA):-

A Finite Automata is called Deterministic finite Automata if There is only one path for a specific input from current state (or) present state to next state.

Eg:-



* q_0 is the present state, on applying of input 'a' Then it moves to The next state q_1.

* similarly on applying of input b Then it moves to The next state q_2

* q_1 is The present state, on applying of input 'a' Then it moves to The next state q_1.

* Similarly on applying of input b Then it moves to The next state q_2.

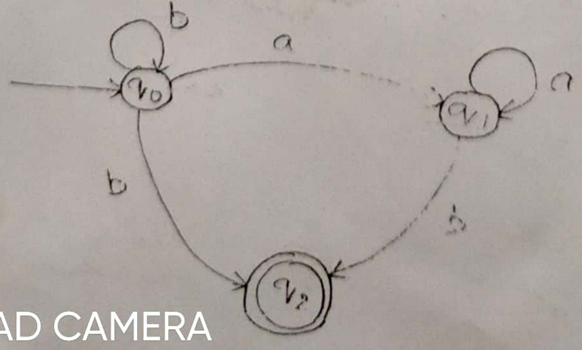
* q_2 is The present state, on applying of input 'a' Then it moves to The next state q_2.

* Similarly on applying of input b Then it moves to The next state q_0.

Non-Deterministic Finite Automata (NFA):-

The concept of NFA is exactly reverse to DFA.

* The finite Automata is called NFA when There exists many paths for a specific input from current state to next state.



* for state q_0 on applying of input a it moves to q_1 but on applying of input b q_0 remains in The same stat (or) moves to The state q_2. Here The situation is Non-determinism.

* Another fundamental difference between DFA and NFA is their transition functio

* $\boxed{\begin{array}{ll} \text{for DFA} & \delta : Q \times \Sigma \rightarrow Q \\ \text{for NFA} & \delta : Q \times \Sigma \rightarrow 2^Q \end{array}}$

Representation of Finite Automata:-

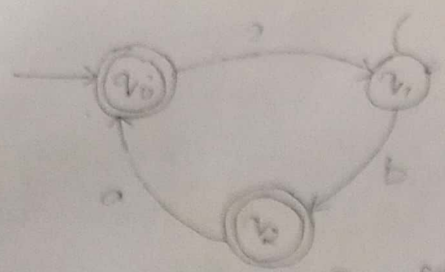
Finite Automata can be Represented in two ways:-

* Transition diagram
* Transition Table

Transition diagram:- A transition diagram is a graph, consists of

* a set of states as circles.
* Initial State q_0 with circle and arrow
* Final states as double circle.
* q_0 as initial state and final state, represent it as

Eg:-



* There are 3 states $\{q_0, q_1, q_2\}$
* There are 2 inputs $\{a, b\}$.
* initial state is q_0.
* final states are q_0, q_2.
* q_1 is the Normal state.

Transition Table :- The transition table is a tabular representation where rows correspon-ds to states and columns corresponds to inputs.

* Initial state is given by($\longrightarrow q_0$) and final state by ◯.

Eg:-

Σ/Q	a	b
$\rightarrow q_0$	q_1	q_2
q_1	q_0	q_1
q_2	q_1	q_2

inputs — next states — present states

* From the table,
* There are 3 states $\{q_0, q_1, q_2\}$.
* There are 2 inputs $\{a, b\}$
* initial state is q_0
* final state is q_2
* normal state is q_1.

2)

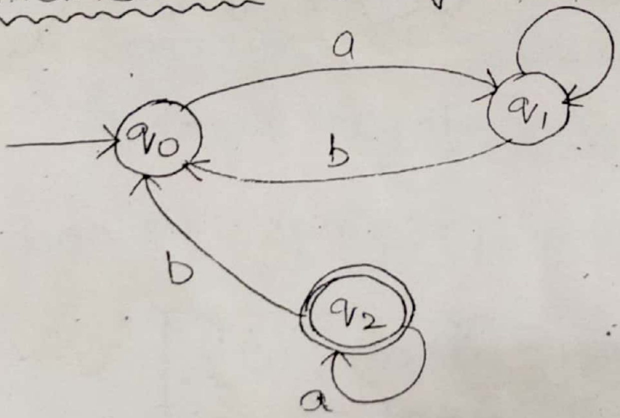
Σ/Q	a	b	ε
→ q_0	–	q_1	q_2
q_1	q_1	q_2	q_3
q_2	q_3	q_0	q_2
q_3	q_1	q_1	q_0.

* '–' (empty) means No existence of present state and present input combination.

* There are 4 states $\{q_0, q_1, q_2, q_3\}$.
* There are 3 inputs $\{\epsilon, a, b\}$.
* Initial state is q_0
* final state is q_0
* normal states are q_1, q_2, q_3.

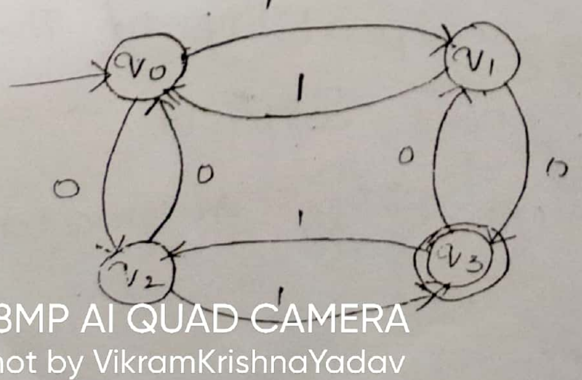
Equivalence between Transition diagram and Transition table :-

* Transition diagram and transition table



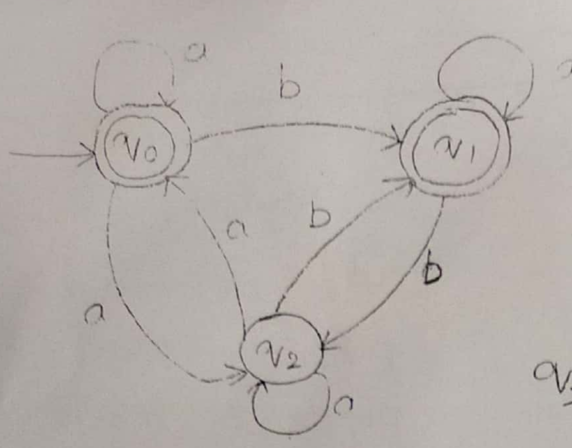
Σ/Q	a	b
→ q_0	q_1	–
q_1	q_1	q_0
q_2	q_2	q_0

*



Σ/Q	0	1
→ q_0	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3

*



Σ/Q	a	b
$\rightarrow q_0$	$\{q_0, q_2\}$	q_1
q_1	q_1	q_2
q_2	$\{q_0, q_2\}$	q_1

Language Acceptance :-

A string 'x' is accepted by a finite Automata $M = (Q, \Sigma, \delta, q_0, F)$ if $\boxed{\delta(q_0, x) = q,\ q \in F}$, represents the acceptability of strings.

Note :- A final state is also called an Accepting State.

Properties of transition Function :-

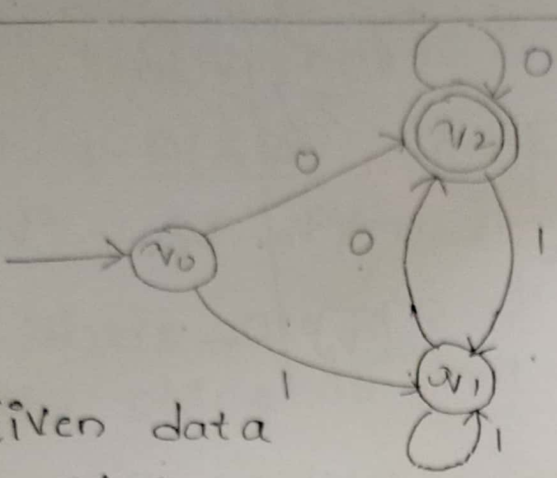
Property-1 :-

* $\boxed{\delta(q, \epsilon) = q}$ In a Finite Automata means the state of the system can be changed only by an input symbol.

* **Property-2 :-** For all strings 'w' and input symbol 'a',

$$\boxed{\begin{aligned} &* \ \delta(q, aw) = \delta(\delta(q, a), w). \\ &* \ \delta(q, wa) = \delta(\delta(q, w), a). \end{aligned}}$$

Examples :- check the acceptability of the string that always ends with 00. based on the given Finite Automata.



Given data

Sol:

Σ / Q	0	1
$\rightarrow q_0$	q_2	q_1
q_1	q_2	q_1
q_2	q_2	q_1

consider $w = 101100$.

from property 2,

$\delta(q, aw) = \delta(\delta(q,a), w)$.

$\delta(q_0, \underset{a}{1}\underset{w}{01100}) \vdash \delta(\delta(q_0, 1), 01100)$

$\vdash \delta(q_1, \underset{a}{0}\underset{w}{1100})$

$\vdash \delta(\delta(q_1, 0), 1100)$

$\vdash \delta(q_2, \underset{a}{1}\underset{w}{100})$

$\vdash \delta(\delta(q_2, 1), 100) \vdash \delta(q_1, \underset{a}{1}\underset{w}{00})$

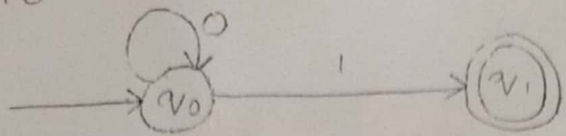
$\vdash \delta(\delta(q_1, 1), 00) \vdash \delta(q_1, \underset{a}{0}\underset{w}{0})$

$\vdash \delta(\delta(q_1, 0), 0) \vdash \delta(q_2, 0)$

$\vdash q_2 \in F$.

the string is accepted.

Q) which of the following strings 0001, 01001, 0000110 are accepted by following DFA



Σ/Q	0	1
$\rightarrow q_0$	q_0	q_1
q_1	–	–

$x = 0001$

$\delta(q_0, x) = q,\ q \in F$

$\delta(q_0, 0001)$

apply property 2.

$\delta(q, aw) = \delta(\delta(q,a), w)$ $(\because \delta(q_0,0)=q_0)$

$(\because \delta(q_0,1)=q_1)$

$\delta(q_0, 0001)$ $(\because \delta(q_1,0) = -)$

$\vdash \delta(\delta(q_0,0), 001)$ $(\because \delta(q_1,1) = -)$

$\vdash \delta(q_0, 001)$

$\vdash \delta(\delta(q_0,0), 01)$

$\vdash \delta(q_0, 01)$

$\vdash \delta(\delta(q_0,0), 1)$

$\vdash \delta(q_0, 1)$

$\vdash q_1 \in F.$

$= q_1 \in F.$

$\therefore$ The string is accepted.

$x = 01001$

$\delta(q_0, x) = q,\ q \in F$

$\delta(q_0, 01001)$

apply property 2.

$\delta(q, aw) = \delta(\delta(q,a), w)$

$\delta(q_0, 01001)$

$\vdash \delta(\delta(q_0,0), 1001)$

$\vdash \delta(q_0, 1001)$

$\vdash \delta(\delta(q_0,1), 001)$

$A_2' \rightarrow A_2A_2A_2' \mid A_2A_2$

Now consider $A_2' \longrightarrow A_2A_2A_2'$ are

$\vdash \delta(q_1, 001)$

$\vdash \delta(\delta(q_1, 0), 01)$

$\vdash \delta(-), 01)$

There are no transitions for $(q_1, 0)$. so

The string is not accepted.

$x = 0000110$.

$\delta(q_0, 0000110)$

$\vdash \delta(\delta(q_0, 0), 000110)$

$\vdash \delta(q_0, 000110)$

$\vdash \delta(\delta(q_0, 0) 00110)$

$\vdash \delta(q_0, 00110)$

$\vdash \delta(\delta(q_0, 0), 0110)$

$\vdash \delta(q_0, 0110)$

$\vdash \delta(\delta(q_0, 0), 110)$

$\vdash \delta(q_0, 110)$

$\vdash \delta(q_0, 1), 10)$

$\vdash \delta(q_1, 10)$

$\vdash \delta(\delta(q_1, 1), 0)$

$\vdash \delta(-), 0)$

There are no transitions for $(q_1, 1)$. so

The string is not accepted.

$=q_0)$

$=q_1)$

$= -)$

$= -)$