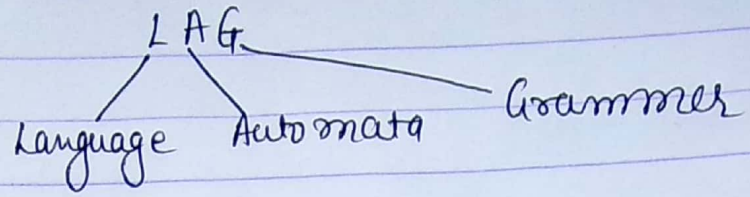
27/01/2024

Automata Theory

TOC – Mathematical Model



Language :-

① Symbol. ⇒ {a, b, c, 0, 1, 2, 3}

To understand a language you have to identify symbols unit.

② Alphabet Σ (a, b) ← finite set of symbols

③ String (words) collection/sequence of alphabets

L = {a, b, aa, ab, ba, bb}.

Length of string = 2
{aa, ab, ba, bb}

length = 3
{aaa, aab, aba, abb, baa, bab, bba, bbb}

④ Language ← collection of strings.

Q ⇒ Σ(a, b)

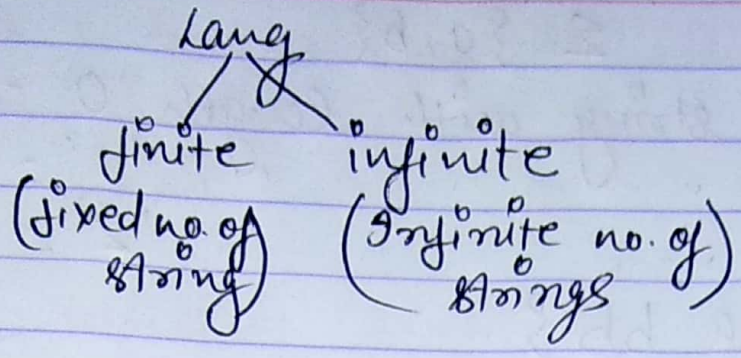
L_1 = strings of length 3 ← finite

L_2 = Strings starts from a and end with b

L_2 = {aa, aaa, aaaa, aba, abba ...}

L_3 = String with length 0

= ϵ



L

L1 = String length = 2

L1 = {aa, ba, ab, bb}

L2 = atleast one a

{a, aa, aaa :---

ab, aab, abba, abbba}

Q ⇒ find if 'abba' is part of L2.

Automata ⇒ model or machine

⇒ whether a string is part of language

FA

PDA

TM

Power of Σ $\Sigma = \{a, b\}$

Σ^0 = set of all string with length '0' = {ϵ} (NULL string)

Σ^1 = " " " " " " '1' = {a, b}

Σ^2 = " " " " " " '2' =

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

Σ^* ⇒ **Kleene closure**

⇒ Set of all strings of all length possible on a & b possible

⇒ $(a+b)^*$

⇒ {a, b, aa, bb, ----} Infinite language

Σ^+ ⇒ positive closure

= set of all strings possible but not Σ^0

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

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

Grammer

Grammer = A grammer 'G' is dfined as Quadruple

$G = \{V, T, P, S\}$

Grammer whether perticular sentence is part of the language

My name is Snigdha.

✓ Are you a student?

✗ a student are you?

$G = \{V, T, P, S\}$

- V — Variable (Capital letter)
- T — Terminal (Small letters)
- P — product^n rule
- S — Start symbol

eg1 $S \rightarrow aSb/\epsilon$

$L = \{\epsilon, asb, aaSbb, aaaSbbb \ldots\}$

$L = \{\epsilon, ab, aabb, aaabbb \ldots\}$

$L = \{a$ followed by equal no. of $b\}$

$= a^n b^n \quad n \geq 0$

= Is 'abab' part of this language L?

eg2

$S \rightarrow SS$

$S \rightarrow asb$

$S \rightarrow bsa$

$S \rightarrow \epsilon$

$L = \{\epsilon, asb, babsab, asbbsa \ldots$

$= \{\epsilon, ab, abab, abba \ldots\}$

$L =$ no. of a equal to no. of b in a string}

Page
Date

Chapter No 1 :- FINITE AUTOMATA.

Q1. What is Automata?

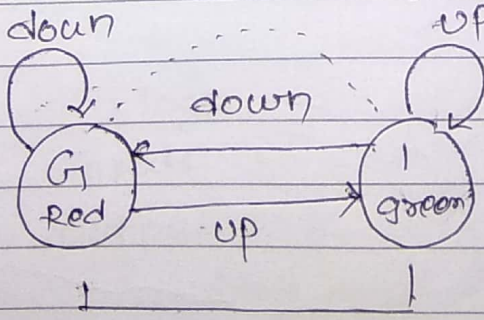
Ans. An automation is an abstract model of a digital computer.

1. An automation has a mechanism to read the input from a input tape.
2. A machine with intermediate stages which on recival of inputs choose a particular path from the initial stage to the final stage to produce a valid output, such a machine having finite number of intermediate stages is called finite automata or finite state machine.

* Basic concepts.

1. Alphabet :- set of input symbols eg Σ $\{0, 1\}$.
2. String :- sequential arrangement of ~~alphabet~~ input symbols.
3. Empty string :- ϵ or null
4. language :- collection of strings.
5. String length :-

* State Transition Digram



finite Automata.

Note :- Use theory in Each sum.

Page
Date

* Finite state Machine. (Theory)

1. FSM consists of finite set of states 'S' which alert on reciving the input set I to produce output set O

2. FSM can be mathamatically represented as follows.

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

where,

Q :- set of states

Σ :- Alphabetic set of input symbols

δ :- Transition function

$\delta - Q \times \Sigma \rightarrow Q$ -used to find next state.

q_0 :- Initial state.

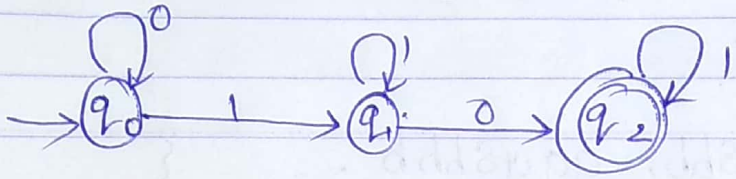
f - set of final states.

Finite automation (FA)

FA is a abstract computing device. It is a mathematical model of a system with discrete inputs, outputs states and set of transitions from state to state that occurs on ~~out~~ input symbols from alphabet Σ.

Its representations :-
- Graphical (Transition Diagram)
- Tabular (Trasition Table)
- Mathematical (Transition function)

① Transition Diag :-
It is a directed graph associated with vertices of the graph corresponds to the state of finite automata

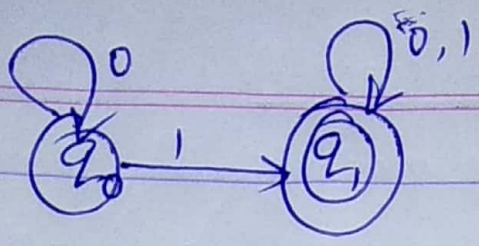


$\{0, 1\}$ – inputs
q_0 – initial state
q_1 – intermediate
q_2 – final.

② Transition Table — Tabular form

	Σ	Σ	Σ	Σ
q →				
q →				
q →				

$$\delta = Q \times \Sigma \rightarrow Q$$



.	0	1
q_0	q_0	q_1
q_1	q_1	q_1

Transition function. (δ)

- Two parameters are passed this function:-
  1. current state
  2. Input symbols
- The transition function returns a state which can be called as next state

δ(current state, current i/p) = next state

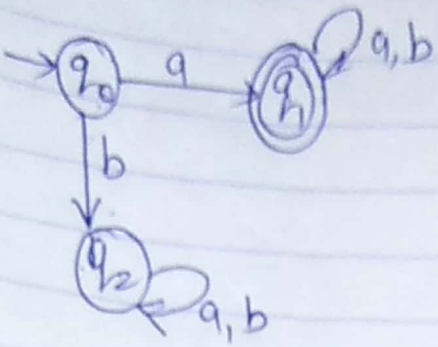
eg $\delta(q_0, 0) = q_1$

or

$\delta(q_0, 1) = q_1$

L = String starting with a for $\Sigma^*(a, b)$

$L = \{a, aa, abb, aaa, abbb \ldots, ba\}$



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

abb

$\delta(q_0, a) = q_1$

$\delta(q_1, b) = q_1$

$\delta(q_1, b) = (q_1)$ ← final state

String abb belongs to L

bab

$\delta(q_0, b) = q_2$

$\delta(q_2, a) = q_2$

$\delta(q_2, b) = (q_2)$ ← non final state

bab does not belong to L

Questions (Practice)

1. String containing 'a'
2. string ends with 'a'
3. String starting with 'a' & ends with 'b'
4. " must start with substring w
   1. w = ba 2. w = abb
5. every string must end with substring w
   1. w = bb 2. ab 3. bab
6. Design a DFA over $\Sigma = \{a, b\}$ such that every string accepted must contain a substring w
   1. w = ad 2. w = ba 3. w = abb
7. Design a DFA over $\Sigma = (a, b)$ such that every string
   1. must start & end with same symbol
   2. must start & end with diff. symbol
   3. must start & end with w = ab

(8) Design DFA over $\Sigma = (a, b)$ such that every string accepted must

(a) $|w| = 2$

(b) $|w| \geq 2$

(c) $|w| \leq 2$

(d) $|w|_a = 2$

(e) $|w|_a \geq 2$

(f) $|w|_a \leq 2$

To design FSM/FA we always following steps:-

1. Logic
2. Formal Notations
3. Transition Function
4. Testing the automata.

Q1 Design a FSM to check if the given number (decimal) is divisible by 3 or

⇒ ~~FSM consists of~~ Definition of FSM

1. Logic : $I = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$O = \{Y, N\}$

$S = \{q_0, q_1, q_2\}$.

(n mod 3) $q_0 \rightarrow n \bmod 3 = 0$

$q_1 \rightarrow n \bmod 3 = 1$

$q_2 \rightarrow n \bmod 3 = 2$

$0/3 = 3/3 = 6/3 = 0$ remainder $\rightarrow q_0$

$1/3 = 4/3 = 7/3 = 1$ " $\rightarrow q_1$

$2/3 = 5/3 = 8/3 = 2$ " $\rightarrow q_2$

Q = Set of all states = $\{q_0, q_1, q_2\}$

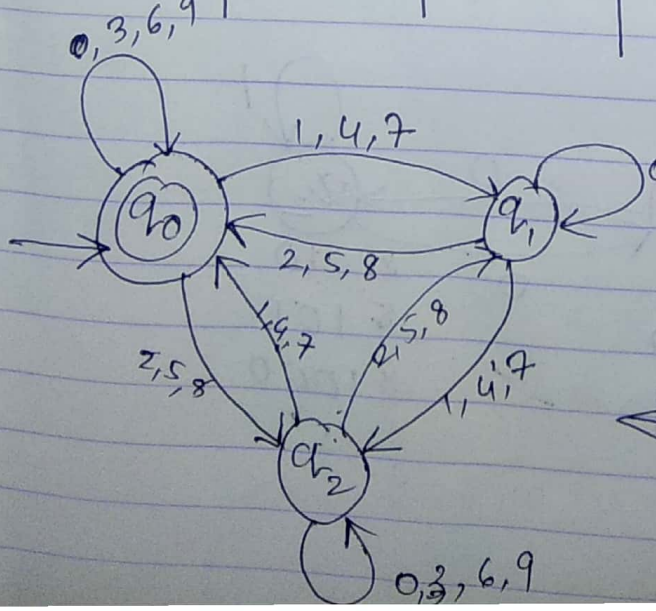
Σ = Set of all I/P symbols = $\{0,1,2,3,4,5,6,7,8,9\}$

δ = Delta = step 3 = Transition function (table/diag)

q_0 = start state = q_0

F = Set of all final states = $\{q_0\}$.

Q \ Σ	0, 3, 6, 9	1, 4, 7	2, 5, 8
rem 0 $*(q_0)$	q_0	q_1	q_2
rem 1 q_1	q_1	q_2	q_0
rem 2 q_2	q_2	q_0	q_1



Transition Diagram

$(q_0, 369)$

$\vdash (q_0, 69)$

$\vdash (q_0, 9)$

$\vdash (q_0, 0)$

$\vdash (q_0, \epsilon) = y$

$(q_0, 247)$

$\vdash (q_2, 47)$

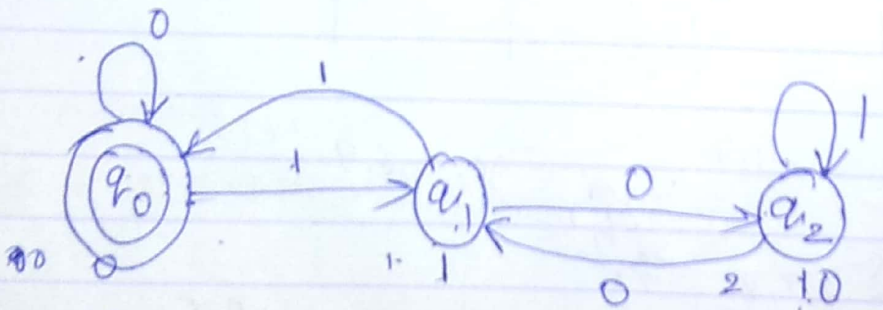
$\vdash (q_0, 7)$

$\vdash (q_1, \epsilon) = n$

0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10

q_0

10 100



0 1 10

3 11 4 100 5 101

6 110 7 111 8 1000

DFA = $\delta(q_0, a) = q_1$ but

NFA = $\delta(q_0, a) = \{q_1, q_2\}$

NFA

Acceptance of NFA :- A string ω is said to be accepted by a NFA if there exist at least one transition path on which we start at initial state and ends at final state.

$$\delta^*(q_0, \omega) = F$$