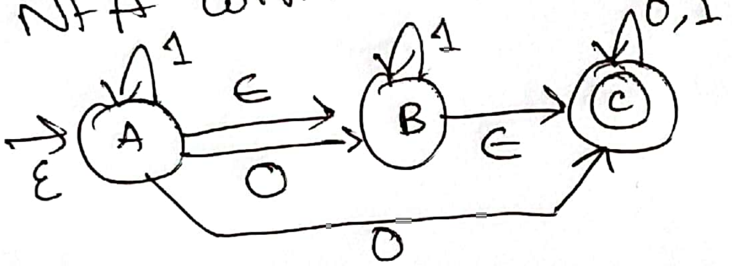
Subject :.. Date :

DFA & NFA definition in $\boxed{\text{All}}$ copy

NFA with epsilon transition: The automaton may allow to change state without reading the input symbol.
The Epsilon closure for a given state X is a set of states which can be reached from state X with only (NULL) on ϵ moves including the state X itself. Example:-

Ⓐ The NFA with ϵ move -



State	0	1	epsilon
A	B,C	A	B
B	–	B	C
C	C	C	–

The, ϵ-closure,
ϵ closure (A) : {A, B, C}
ϵ closure (B) : {B, C}
ϵ closure (C) : {C}

Moore & Mealy description $\boxed{\text{ALL}}$ copy ~~to~~ [illegible]

*Applications of FA

① It plays an important role in Compiler design

② In switching theory and design and analysis of digital circuits, automata theory is applied.

③ Design & Analysis of complex S/w & h/w system

④ To proove correctness of program automata is used.

⑤ To design finite state machines such as Moore & Mealy.

⑥ It is base for formal languages & these formal languages are useful of programming languages

2018 2 ⓒ :: Every formal language is accepted by FA, if its Firstly DFA. Then if it's acceptable by dfa. It's ultimately accepted by nfa, Thus all FA accepts Regular language

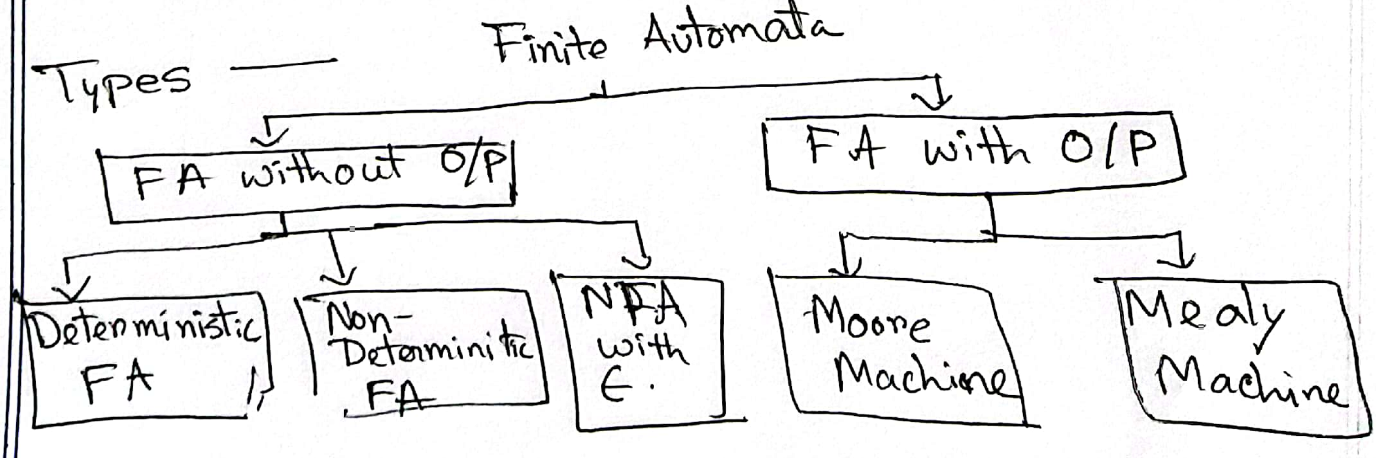
2018 1 (b) Define & Classify finite Automata

Finite automata is an 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 input symbols from alphabet Σ.

Its representation,
→ Graphical (Transition Diagrams or Transition table)
→ Tabular (Transition table)
→ Mathematical (Transition function or Mapping function)

Formal definition of FA.
A finite automata is 5-tuples; they are -
$M(Q, \Sigma, \delta, q_0, F)$.
Q: finite set of states.
Σ: finite set of alphabets
δ: $Q \times \Sigma \rightarrow Q$ transition function
$q_0 \in Q$: start state or initial state
$F \subseteq Q$: Set of accept states / final states.

Types —



Theory of Computation

Subject : 2018 Date :

1(a) Define Alphabet, string, language. Discuss basic operations of language. 3

Alphabet: An alphabet is a finite set of symbols, such as $\{a, b\}$ or $\{0,1\}$ or $\{A, B, C, \ldots, Z\}$

We will usually use the Greek letter Σ to denote the alphabet. example; $\Sigma = \{0, 1\}$ is an alphabet of binary digits

String: A String is a finite sequence of symbol s selected from some alphabet. It is generally denoted as w. For ~~example for alphabet $\Sigma = \{0,1\}$ w = 010101 is a string.~~

~~A string over Σ is a finite sequence of symbol in Σ~~

~~For a string x, $|x|$ stands for the length (the no. of symbols) of x.~~ and the length of a string denoted as $|w|$.

~~Note~~

Σ^* is the set of all possible string (often power set (need not be unique have or we can say multiset) of string)

so this implies that language is a subset of Σ^*

⇒ Empty string is the string with zero occurance of symbol represented as ε.

Subject : .... Date :

Number of Strings (of length 2) generated over alphabets $\{a, b\}$ - aa, ab, ba, bb

Length of String $|w| = 2$, Number of strings $= 4$.

Language: A language over Σ (alphabet) is a subset of Σ^* (the set of all strings over Σ).
This means that language L is subset of Σ^*.
An example is English language, where the collection of legal English words is a set of strings over the alphabet that consists of all the letters.
The language formed over 'Σ' can be finite or Infinite.

Example of Finite language.

$\Sigma = \{a\, x, y\}$

$L1 = \{$set of string of 2$\}$

$L1 = \{xy, xx, yx, yy\}$

Example of Infinite language:

$L1 = \{$set of all strings starting with 'b'$\}$

$L1 = \{ba, baa, babb, bbb, ..\}$

Subject : .... Date :

The basic operations on strings are ① concatenation, ② union, ③ Star / Kleene Closure.

① Union: Let A & B be set of two regular languages then $A \cup B = \{x \mid x \in A \text{ or } x \in B\}$ is also regular.

2. Concatination: $A.B = \{x.y \mid x \in A \ \& \ y \in B\}$

3. Star / Kleene Closure: $A^* = \{x_1, x_2, x_3 \ldots x_k \mid k \geq 0 \ \& \ x_i \in A\}$

$A = \{a, b, \epsilon\}$, $B = \{x, y, z\}$ (i) $A \cup B = \{a, b, x, y, z, \epsilon\}$

(2)(i) $A \cdot B = \{ax, ay, az, bx, by, bz, x, y, z\}$ [ε এর সাথে যা আসে তাই বসবে]

(ii) $BA = \{xa, ya, za, xb, yb, zb, x, y, z\}$

$A^* = \{a^0, a^1, a^2, \ldots, b^0, b^1, b^2, \ldots, ab, ba, aab, \ldots\}$

$a^0 \rightarrow \epsilon$, $b^0 \rightarrow \epsilon$

$B^* = \{\epsilon, x, y, z, xx, xy, xz, yx, yy, yz, zx, zz, xyz, \ldots\}$

Subject :.. Date :

$A = \{good, bad\}$, and $B = \{boy, girl\}$

$A \cup B = \{good, bad, boy, girl\}$

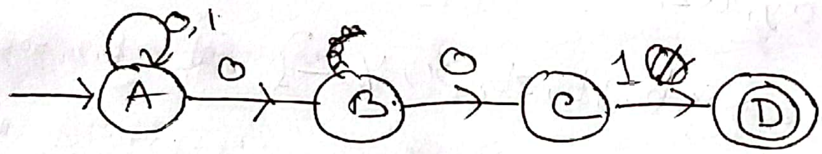
$A \cdot B = \{good boy, good girl, bad boy, bad girl\}$

$A^* = \{\varepsilon, good, bad, goodgood, badbad, goodbad, badgood, goodgoodgood, \ldots\}$

2018 2(b) Construct an NFA that accepts a binary language ending with 001. Convert it to a DFA using subset construction method.

Solution: Given, $\Sigma = \{0, 1\}$

The $L(M) = \{$ ~~Bina~~ w|w ends with 001$\}$

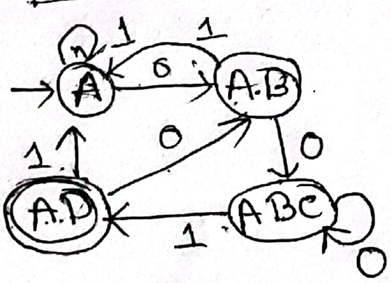


The state transition table of the NFA.

	0	1
→A	{A,B}	{A}
B	{C}	{ }
C	{ }	*{D}
*D	{ }	{ }

The STT for DFA using STT of NFA.

	0	1
→[A]	[AB]	[A]
[AB]	[ABC]	[A]
[ABC]	[ABC]	*[AD]
*[AD]	[AB]	[A]



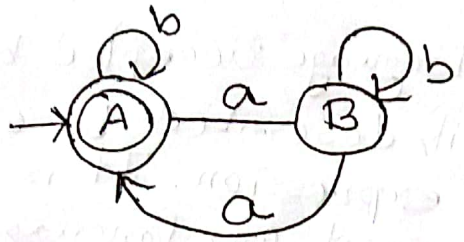
DFA.

Subject :.. Date :

2(a) Construct mDFA accepting the following languages:

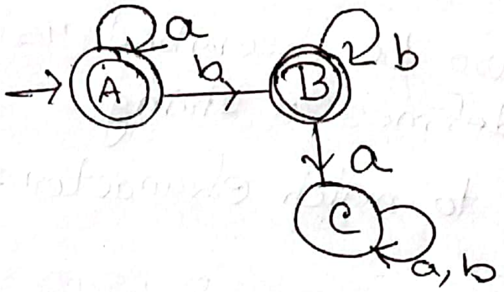
i) $\{w \in \{a, b\}^* \mid w$ has even number of a's$\}$

$L(M) = \{ \varepsilon, baab, aba, aabaa, abab, \ldots \}$



ii) $\{a^n b^m \mid m, n \geq 0\}$ [এটায় sequence matter করে, a এর পরে must b আসবে।

$L(M) = \{ \varepsilon, a, b, ab, aab, abb, aabb, \ldots \}$



[b এর পরে a accepted না]

bax

ab ✓

aaa..b ✓

b--ax

babax

~~b এর~~

3(a) Define regular expression. Write a regular for a language containing the strings starting & ending with same symbol over alphabet $\{a, b\}$

Regular ~~expression~~ language: If Σ is an alphabet, the set R of regular languages over Σ is defined as follows.

(1). The language $\emptyset$ is an element of R, and for every $a \in \Sigma$, ~~the~~ language $\{a\}$ is ~~$\emptyset$~~ in R

Subject : .... Date :

(2) For any two language L_1 and L_2 in R, the three languages $L_1 \cup L_2$, $L_1 L_2$ and L_1^* are elements of R.

~~The lang the~~ :

Regular expression: The language accepted by finite automata can be easily described by simple expression called Regular expression. It is the most efficient way to represent any language

The languages accepted by some regular expression are reffed to as Regular string.

A regular expression can also be described as a sequence of pattern that defines a string.

Regular expression are used to match characters combinations in string.

Example:

In a regular expression, x^* means zero or more occurrence of x. It can generate

$\{\epsilon, x, xx, xxx, xxxx, \ldots\}$

In a regular expression x^+ means one or more occurrence of x. It can generate

$\{x, xx, xxx, \ldots\}$

Subject : .... Date :

The various operations on regular language are -

Union: If L & M are two regular language then their union, L U M is also a union.

1. $L \cup M = \{ s | s$ is in L or s is in M$\}$

Intersection: If L and M are two regular language then their intersection is also an intersection.

1. $L \cap M = \{ st | s$ is in L and t is in M$\}$

Kleen Closure: If L is a regular language then its Kleen closure $L1^*$ will also be a regular language.

1. L^* = Zero or more occurence of language L.

Example 1: The regular expression of language accepting all combination of a's over the set $\Sigma = \{a\}$.

Soln: All combination of "a" means a may be zero, single, double, and so on. If a is appearing zero times, that is null string.

R. E, $R = a^*$

Example-2: ... all ... except null string of a's over the set $\Sigma = \{a\}$

Soln: Regular expression for language

$L = \{a, aa, aaa, ---\}$

This set indicates that there is no null string.

$R = a^+$

3: All string containing any number of a's and b's.

r. e = $(a+b)^*$

$L = \{\varepsilon, a, aa, bb, b, ab, ba, --\}$

Scanned with CamScanner

Subject : .... Date :

Write regular expression for a language containing the string starting and end ending with same Symbol over alphabe $\{a,b\}$.

$a \cup b \cup b\Sigma^*b \cup a\Sigma^*a$

a
b
a---a
b---b

$R = Q + RP$
$= QP^*$

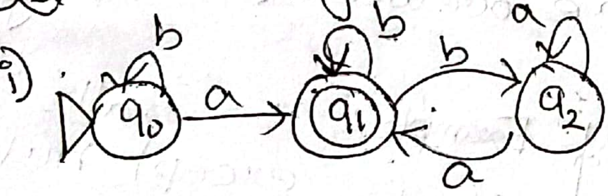
$L = \{w \mid w \text{ contains at least two 1's}\}$ $\Sigma = \{0,1\}$

$\Sigma^* 1 \Sigma^* 1 \Sigma^*$

$L = \{11, \ldots$

3(a) Derive regular expression from the following FAs:

i)



$q_2 = q_1 b a^*$

$q_1 = q_0 a + q_1 b + q_2 a$ —— (i)

$q_2 = q_1 b + q_2 a$ —— (ii)

$q_0 = q_0 b + \epsilon$ —— (iii)

$q_0 = \epsilon + b q_0$
$= \epsilon b^*$
$= b^* \rightarrow$ (vi)

~~From (i) & (ii)~~

(ii) $\Rightarrow q_2 = (q_0 a + q_1 b + q_2 a) b + q_2 a$

$= q_0 ab + q_0 bb + q_2 ab + q_2 a$

$\underbrace{q_2}_{R} = \underbrace{q_0 ab + q_0 bb}_{Q} + \underbrace{(ab + a)}_{P} \underbrace{q_2}_{R}$

$= (q_0 ab + q_0 bb)(ab + a)^*$

$q_2 = q_0 (ab + bb)(ab + a)^*$ —— (iv)

$q_1 = b^* a + q_1 b + q_2 a$
$= b^* a + q_1 b + q_1 b a^* a$
$= b^* a + (b + ba^*a) q_1$
$= (b^* a)((b + ba^*))^*$

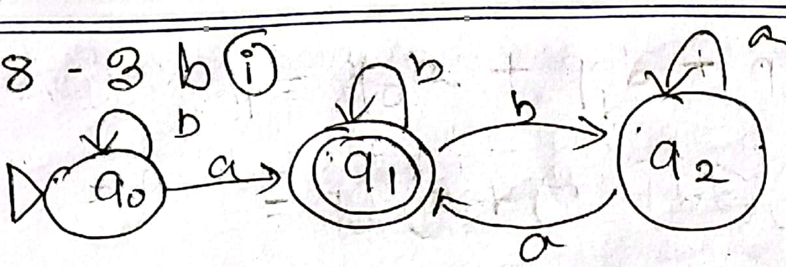
$R = Q + RP$

$= QP^*$

Subject : ....

Date :

2018 - 3 b (i)



Here,

$q_0 = \epsilon + q_0 b$ —— (i)

$q_1 = q_0 a + q_1 b + q_2 a$ —— (ii)

$q_2 = q_1 b + q_2 a$ —— (iii)

From (i) => $q_0 = \epsilon + q_0 b$ [Arden's Theorem]

$= \epsilon b^*$

$q_0 = b^*$ ⟶ (iv)

From (iii) => $q_2 = q_1 b + a q_2$

$= q_1 b a^*$ [Arden's Theorem]

$\therefore q_2 = q_1 b a^*$ —— (v)

Putting values from (iv) & (v) in (ii) we get,

$q_1 = b^* a + q_1 b + q_1 b a^* a$

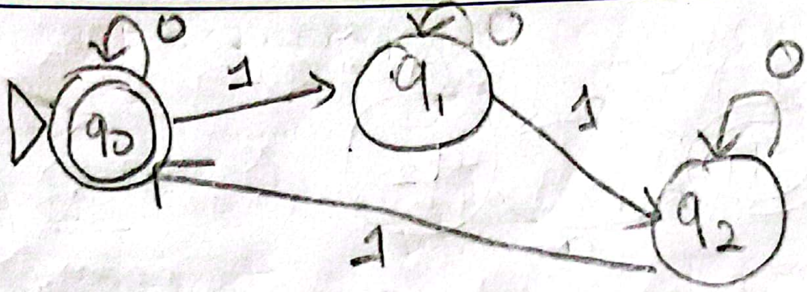
$= b^* a + (b + b a^* a) q_1$

$q_1 = b^* a (b + b a^* a)^*$

Ans.

Subject : .... Date :

3 (b) (ii)



$q_0 = \epsilon + q_0 0 + q_2 1 \longrightarrow$ (i)

$q_1 = q_1 0 + q_0 1 \longrightarrow$ (ii)

$q_2 = q_2 0 + q_1 1 \longrightarrow$ (iii)

(iii) $\Rightarrow q_2 = q_1 1 + q_2 0$

$= q_1 1\, 0^* \longrightarrow$ (iv)

(ii) $\Rightarrow q_1 = q_0 1 + q_1 0$

$= q_0 1 0^* \longrightarrow$ (v)

(i) $\Rightarrow q_0 = \epsilon + q_2 1 + q_0 0$

$= \epsilon + q_1 1 0^* 1 + q_0 0$ [from (iv)]

$= \epsilon + q_0 1 0^* 1 0^* 1 + q_0 0$ [from (v)]

$= \epsilon + (1 0^* 1 0^* 1 + 0) q_0$

$q_0 = \epsilon (1 0^* 1 0^* 1 + 0)^*$

$= (1 0^* 1 0^* 1 + 0)^*$

Digital Products

Scanned with CamScanner

{left is non-terminal}

Subject : .... Date :

3(c) What is the regular language represented by the following left-linear grammar? 2.75

$A \to Aa \mid Ab \mid Ba$, $B \to Cb$, $C \to \varepsilon$

$A \to Aa$
$\Rightarrow Aba$ [$A \to Ab$]
$\Rightarrow Baba$ [$A \to Ba$]
$\Rightarrow Cbaba$ [$B \to Cb$]
$\Rightarrow \varepsilon baba$ [$C \to \varepsilon$]
$\Rightarrow baba$

$A \to Ab$
$\Rightarrow Abb$ [$A \to Ab$]
$\Rightarrow Abbb$ [$A \to Ab$]
$\Rightarrow Babbb$ [$A \to Ba$]
$\Rightarrow Cbabbb$ [$B \to Cb$]
$\Rightarrow \varepsilon babbb$ [$C \to \varepsilon$]
$\Rightarrow babbb$

$A \to Ba$
$\Rightarrow Cba$ [$B \to Cb$]
$\Rightarrow \varepsilon ba$ [$C \to \varepsilon$]
$\Rightarrow ba$

$A \to Aa$
$\Rightarrow Aaa$ [$A \to Aa$]
$\Rightarrow Aaaa$ [$A \to Aa$]
$\Rightarrow Baaaa$ [$A \to Ba$]
$\Rightarrow Cbaaaa$ [$B \to Cb$]
$\Rightarrow \varepsilon baaaa$ [$C \to \varepsilon$]
$\Rightarrow baaaa$

$A \to Aa$
$\Rightarrow Aba$
$\Rightarrow Abba$
$\Rightarrow Babba$
$\Rightarrow Cbabba$
$\Rightarrow babba$

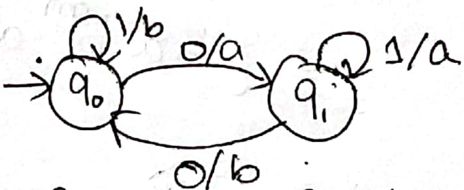
$L(G) = \{ba, baba, babbb, baaaa, \text{------}$

$= \{bab^m a^n \mid m, n \geq 1\}$

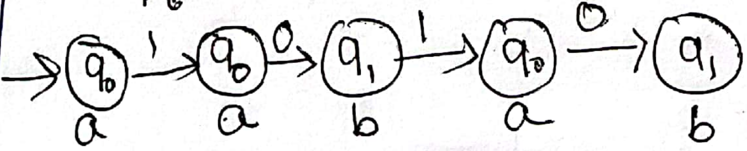
bab^*a^*

Subject : .... Date :

4(a) Contrast between Moore machine & Mealy machine

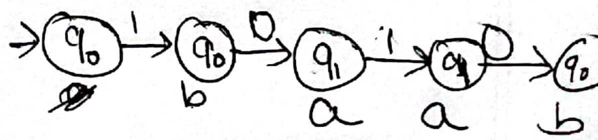
Moore Machine	Difference	Mealy Machine
Output depends only on current state.	1. Definition.	Output depends on the present state as well as on the present input.
More states than Mealy machine	2. States	Fewer states than Moore machine.
Synchronous output and state generation	3. Output generation	Asynchronous output generation.
The design of Moore model is easy	4. Design	The design of Mealy model is complex
Output is a set of state	5. Output type	Output is a set of transition
To design, more hardware is necessary	6. Hardware	It's easier to design with less hardware.
Can refer a counter as a Moore machine	7 Counter	Can not refer a counter as Mealy machine
	8. Example.	

i/p seq 1010 – o/p seq aabab



If sequence 1010

o/p -> baab



Subject :.. Date :

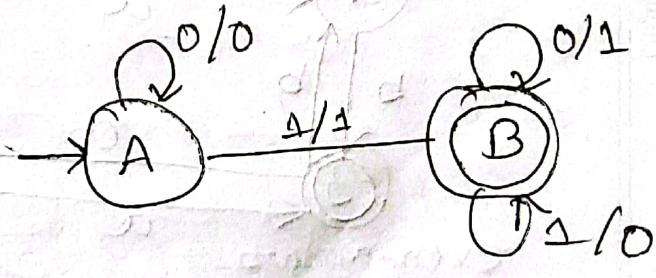
4 (b) Construct a Mealy machine that takes binary numbers as input and produces 2's complement of that number as output. Assume that the string is read ~~LSM~~ LSB to MSB and end carry is discarded.

Ans: 2's Complement logic:- First calculate 1's complement of binary number, Convert 1 to 0 and 0 to 1 and then add 1 to it. For example: if binary number is 1011 then 1's complement is 0100 and it's 2's complement 0101

$$\begin{array}{r} 100 \\ 1's\ \ 011 \\ +1 \\ \hline 2's\ \ 100 \end{array}$$

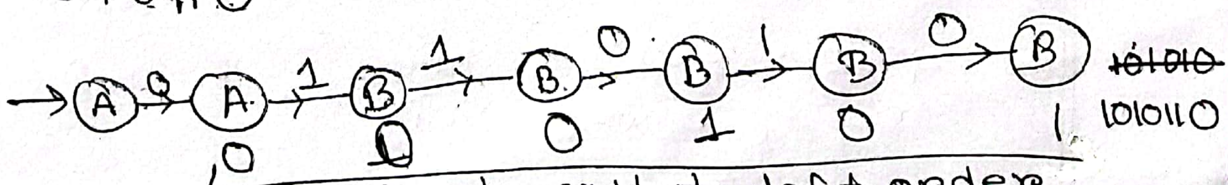
Design Mealy machine:

1. Take initial state A.
2. If ~~there are~~ n number of zeros at initial state, it will remain at initial state.
3. Whenever 1st input 1 is found then it gives output 1 and goes to state B.
4. In state B, if ~~ip~~ input is zero, output will be 1. And if input is 1 then output will be 0.
5. And then set state B as final state.



The approach:
1. Start from right to left
2. Ignore all 0's
3. When 1 comes ignore it and then take 1's complement of every digit.

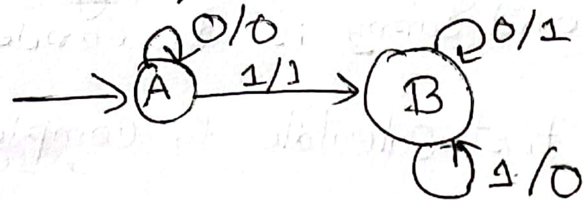
example: 010110 Read right to left.

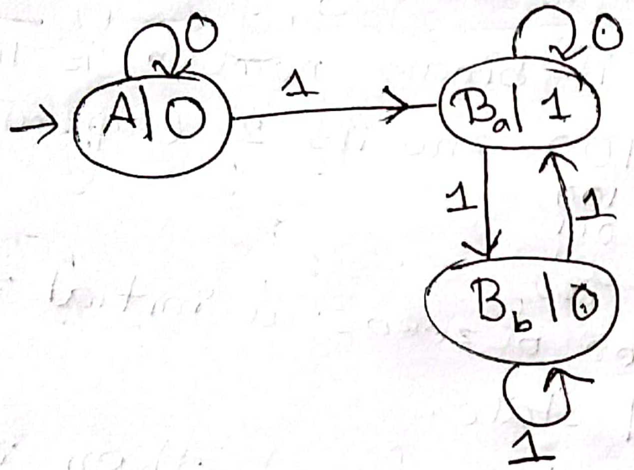


Subject : .... Date :

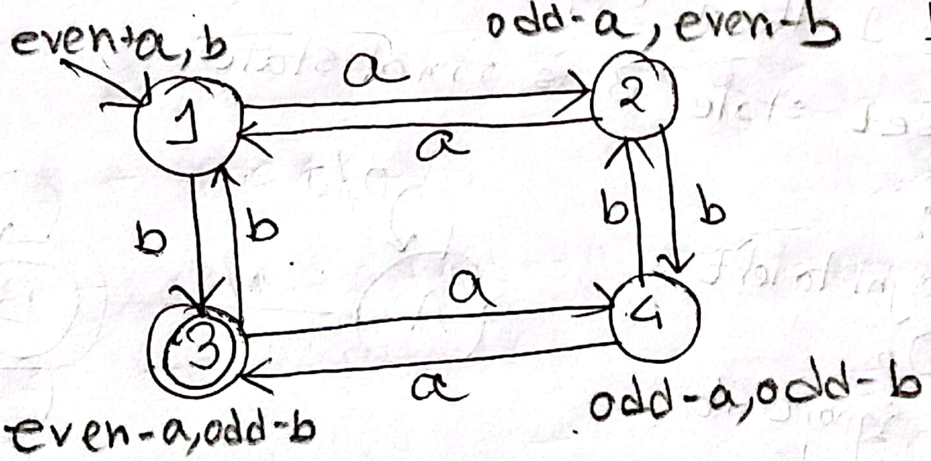
4 (c) Convert above Mealy machine of 4(b) into Moore machine.

Soln: The Mealy machine for 2's Complement.





2019 2(a) i) {w ∈ {a,b}* | w consist of even no. of a's & odd no. of b's} aab, abaaa, babba



$3\,|\,57$ $19 \to 0$ $3\,|\,6 \to 1$ $3\,|\,2 \to 0$ $3\,|\,2$ $0 \to 1$ $3\,|\,1$ $0 \to 1$ $3\,|\,5$ $3\,|\,1 \to 2$ $0 \to 1$

Subject : .... Date :

ⓑ $\{w \in \{0,1,2\}^* \mid w$ is ternary number whose decimal equivalent is divisible by $2\}$.

$0|1|2$

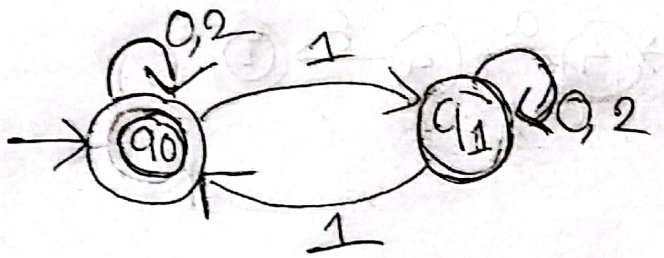
$3^4\ 3^3\ 3^2\ 3^1\ 3^0$

$81\ 27\ 9\ 3\ 1$

$2\ 0\ 1\ 0$

$L = \{0, 2, 4, 6, 8, 10, 12, 14, 16, \ldots\}$

$= \{0, 2, 11, 20, 22, 101, 110, 112, 121, \ldots\}$ $9+2$



	0	1	2
q_0	q_0	q_1	q_0
q_1	q_1	q_0	q_1

ⓒ $\{a^n b^m c^l \mid l, m, n \geq 0\}$.

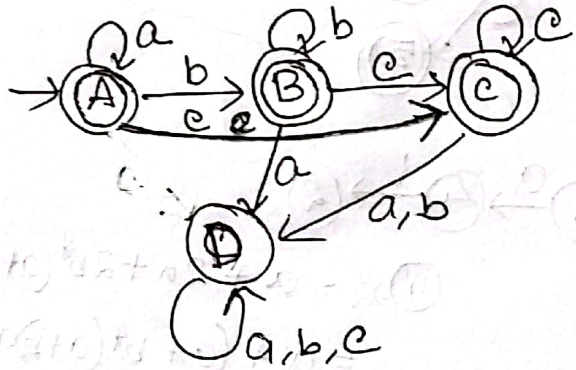
$L = \{a, b, c, ab, ac, bc, abc, abbc, aabc, aab, \ldots\}$

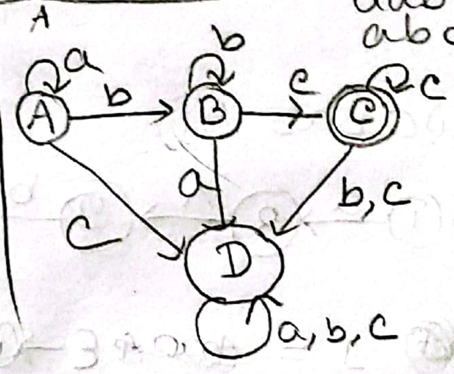
$l, m, n \geq 1$

abc

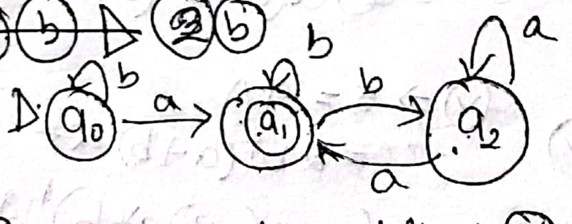
aabbccc

abcc





~~2 ⓑ~~ 2 ⓑ



(i) $q_0 = \epsilon + q_0 b = b^* \to$ (iv)

(ii) $q_1 = q_0 a + q_2 a + q_1 b$.

(iii) $q_2 = q_1 b + q_2 a = q_1 b a^*$ — (v)

$q_1 = b^* a + q_1 b a^* + q_1 b$

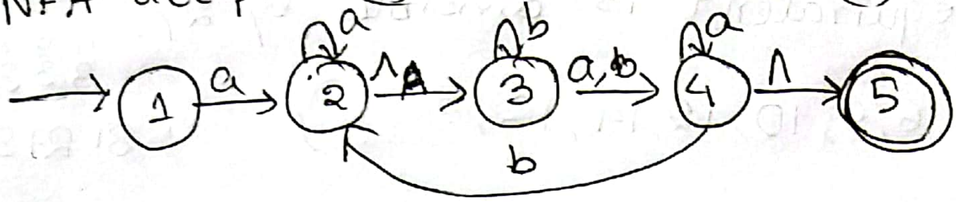
$= b^* a + (ba^* + b) q_1$

$= b^* a (ba^* + b)^*$

~~$bbab$~~

Subject :.. Date :

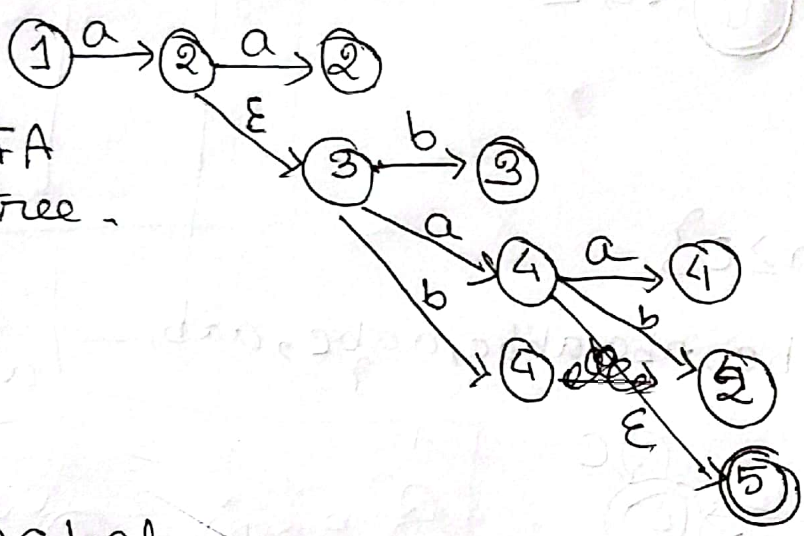
3 (a) Figure 3.1 ~~sto~~ shows ~~F~~G for NFA. For strings - check if NFA accepts - (i) aba, (ii) abab (iii) aaabbb.



(i) aba

$\rightarrow (1) \xrightarrow{a} (2) \xrightarrow{\varepsilon} (3) \xrightarrow{b} (4) \xrightarrow{a} (4) \xrightarrow{\varepsilon} ((5))$

NFA tree.



(ii) abab

$\rightarrow (1) \xrightarrow{a} (2) \xrightarrow{\wedge} (3) \xrightarrow{b} (4) \xrightarrow{a} (4) \xrightarrow{b} (2)$.

(i) $1 = \cancel{000A}\ \varepsilon \rightarrow$ (i)

$2 = 1a + 2a + 4b \rightarrow$ (ii)

$3 = 2\wedge + 3b \rightarrow$ (iii)

$4 = 4a + 3a + 3b \rightarrow$ (iv)

$5 = 4\wedge$ ———— (v)

(iii) $\Rightarrow 3 = 2\wedge + 3b = 2\wedge b^* = 2b^* \rightarrow$ (vi)

(iv) $\Rightarrow 4 = 2b^*a + 2b^*b + 4a$

$= 2b^*(a+b)a^* \longrightarrow$ (vii)

(ii) $\Rightarrow 2 = a + 2a + 2b^*(a+b)a^*b$

$= a + (a + b^*(a+b)a^*b)\, 2$

$= a(a + b^*(a+b)a^*b)^*$

(v) $\Rightarrow 5 = 4\wedge$

$= 2b^*(a+b)a^*$

$= a(a + b^*(a+b)a^*b)^* b^*(a+b)a^*$

~~aaaba~~

a b a

Scanned with CamScanner

2017 3. (a) Define Regular expression.

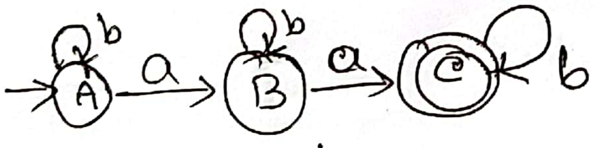
- A expression ~~whi- if & only if~~ which can be recognized by a finite state meching is regular expression.
- Memory of FSM is very limited.
- It cannot store or count string.

~~$a^n b^n$~~ $\{a^n b^n \mid n \geq 1\}$ is not regular language expression. because we dont have the ability to count the no. of times a alphabet comes in a string.

e.g. aaabbb, here $a^3 b^3$ but this cannot be recognized in regular expression as RE cannot count the string content.

3. (b) i) $\{w \in \{a,b\}^* \mid w$ has exactly two a's$\}$

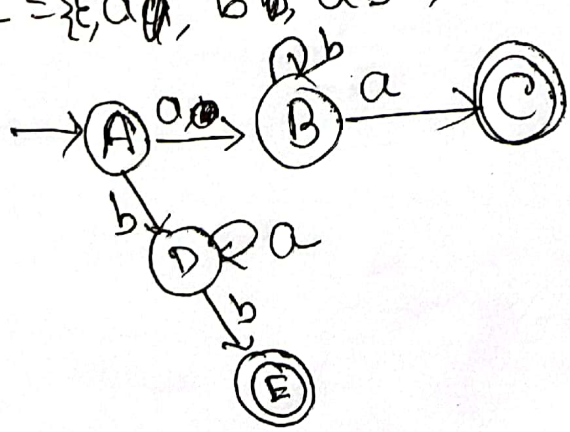
L = { aa, aab, baa, aabb, aabbb, aba, abba, ---- }



$b^* a b^* a b^*$

(ii) $\{w \in \{a,b\}^* \mid w$ starts & ends with sam symbol $\}$

L = {ε, a, b, aba, bab, abba, baab, ---- }



$\varepsilon \mid a \mid b \mid a(a+b)^*a \mid b(a+b)^*b$

$a \cup b \cup a\Sigma^* a \cup b \Sigma^* b$

$(a(a|b)^*a) \mid (b(a|b)^*b)$

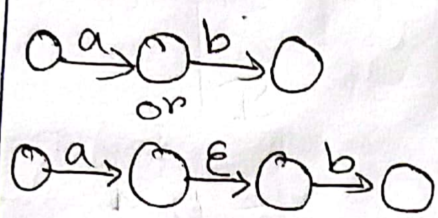
2017 Subject :.. Date :

3 (c) Given R.E $x(x|y)^*|z$. Convert it to DFA via ε-NFA.

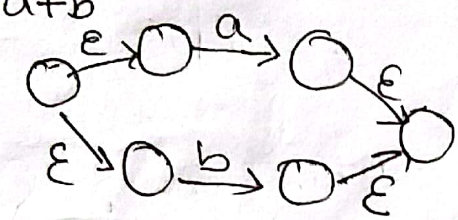
$x(x|y)^*|z$

Re to ε-NFA.

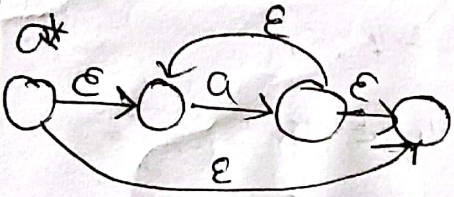
ab



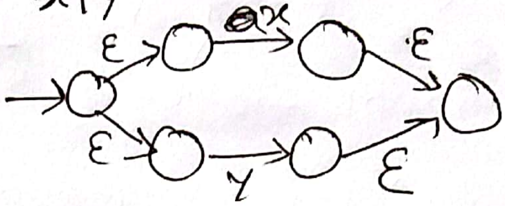
a+b



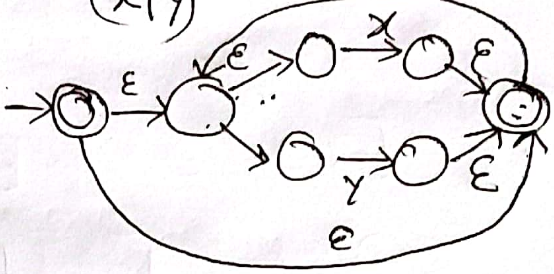
a*



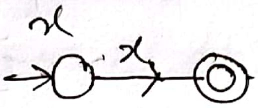
x|y.



$(x|y)^*$

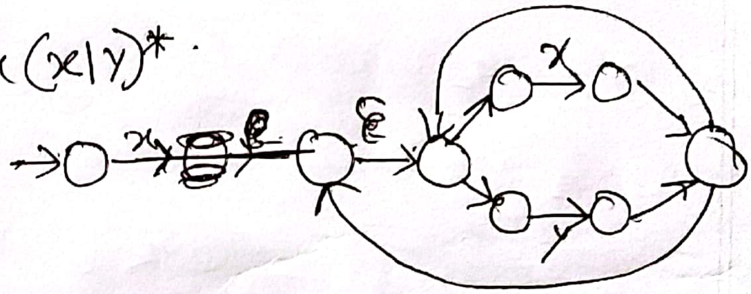


x

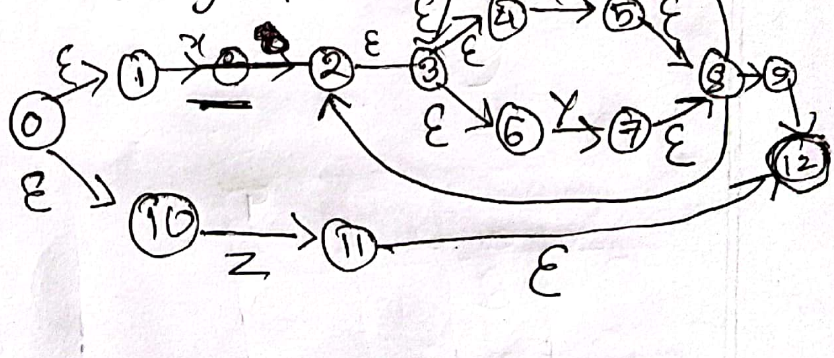


z

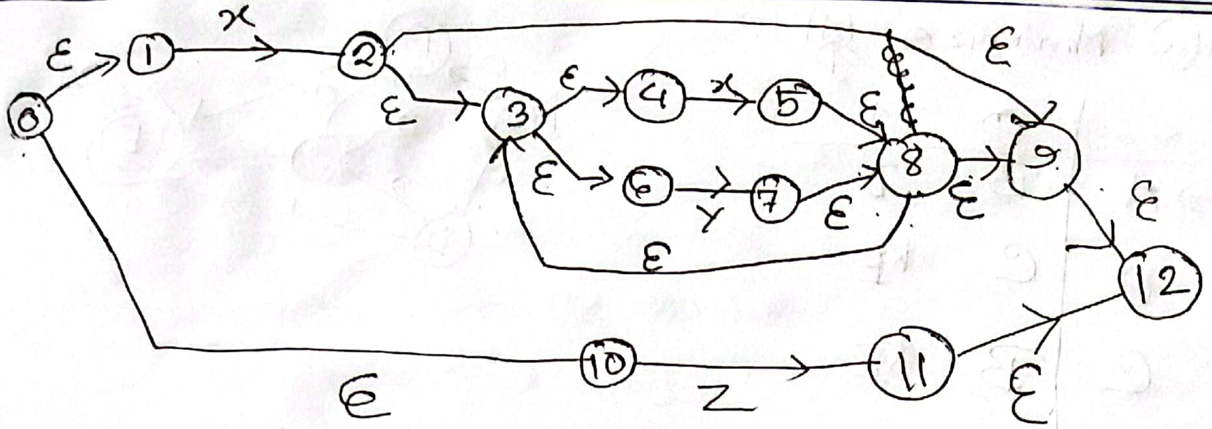
$x(x|y)^*$.



$x(x|y)^*|z$



Subject :.. Date :



	x	y	z	ε
0	–	–	–	{1,10}
1	{2}	–	–	–
2	–	–	–	{3,9}
3	–	–	–	{4,6}
4	{5}	–	–	–
5	–	–	–	{8}
6	–	{7}	–	–
7	–	–	–	{8}
8	–	–	–	{3,9}
9	–	–	–	{12}
10	–	–	{11}	–
11	–	–	–	{12}
12	–	–	–	–

ε-closure(0) = {0, 1, 10}

ε-closure(1) = {1}

ε-closure(2) = {2, 3, 4, 6, 9, 12}

ε-closure(3) = {4, 6}

ε-closure(4) = {4}

ε-closure(5) = {5, 8, 9, 12}

ε-closure(6) = {6}

ε-closure(7) = {8, 9, 12}

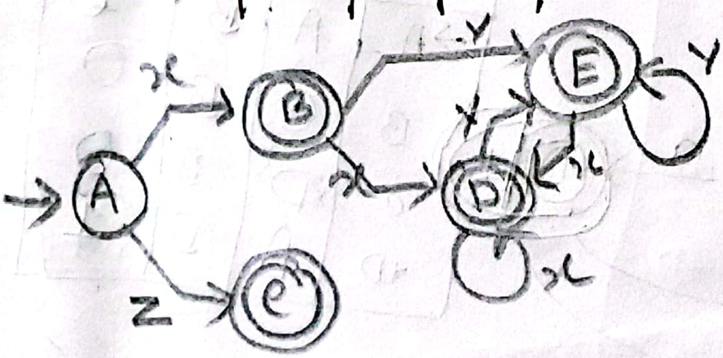
ε-closure(8) = {8, 9, 12}

ε-closure(9) = {9, 12}

ε-closure(10) = {10}

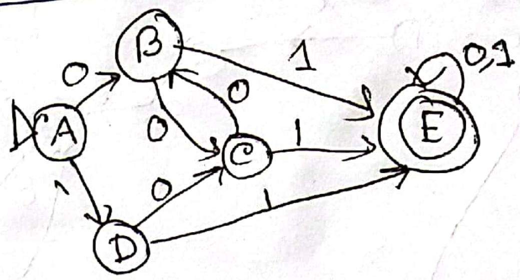
ε-closure(11) = {11, 12}

ε-closure(12) = {12}



Subject : .... Date :

4(a) Minimize ~~N~~DFA:



A $\xrightarrow{0}$ B
B C

A $\xrightarrow{1}$ D
B E

A $\xrightarrow{0}$ B
C B

A $\xrightarrow{1}$ D
C E

B $\xrightarrow{0}$ C
C B

B $\xrightarrow{1}$ E
C E

B D $\xrightarrow{0}$ C
D C

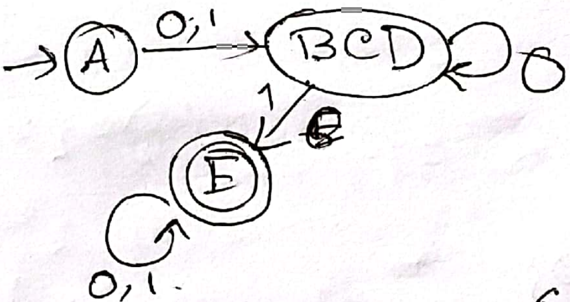
B $\xrightarrow{1}$ E
D E

	0	1
→A	B	D
B	C	*E
C	B	*E
D	C	*E
*E	*E	*E

0 equivalent state: [E] [A B C D]

1 equivalent state: [E] [A] [BCD]

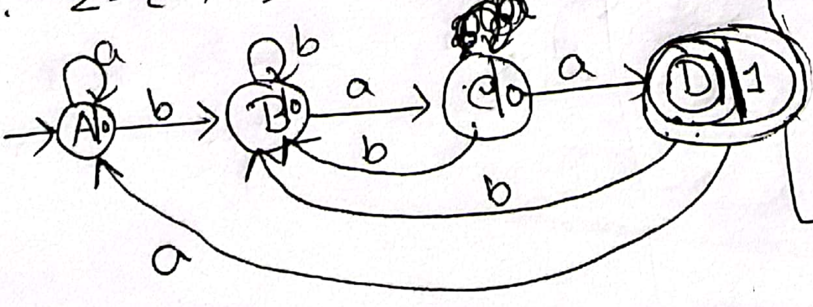
2 equivalent state: [E] [A] [BCD]



4(b) Moore. $\Sigma = \{a,b\}$, L = {all strings over {a,b} as input and prints '1' as output for every occurrence of 'baa' as substring, otherwise print '0'.

check - abaaababaa.

Soln: $\Sigma = \{a,b\}$, $\Delta = \{0,1\}$



Present	Next a	Next b	O/P
→A	A	B	0
B	C	B	0
C	*D	B	0
*D	A	B	1

Subject :... Date :

String - abaaababaa

$$A \xrightarrow{a} A \xrightarrow{b} B \xrightarrow{a} C \xrightarrow{a} \underset{1}{D} \xrightarrow{a} A \xrightarrow{b} B \xrightarrow{a} C \xrightarrow{b} B \xrightarrow{a} C \xrightarrow{a} D$$

4 (e) Let N be set of natural numbers T the set of non-empty subsets of N, and P the set of partitition of N into two nonempty subset. Suppose, $f : T \to P$ is defined by formula $f(A) = \{\{A, N-A\}$ (in other words, for a non-empty subset A of N, f(A) is the partition of N consisting of two subsets A and (N-A). If f is bisection from T to P? Justify your answer.

Subject : .... Date :

2016 1.(a)(ii) Transition diagram:

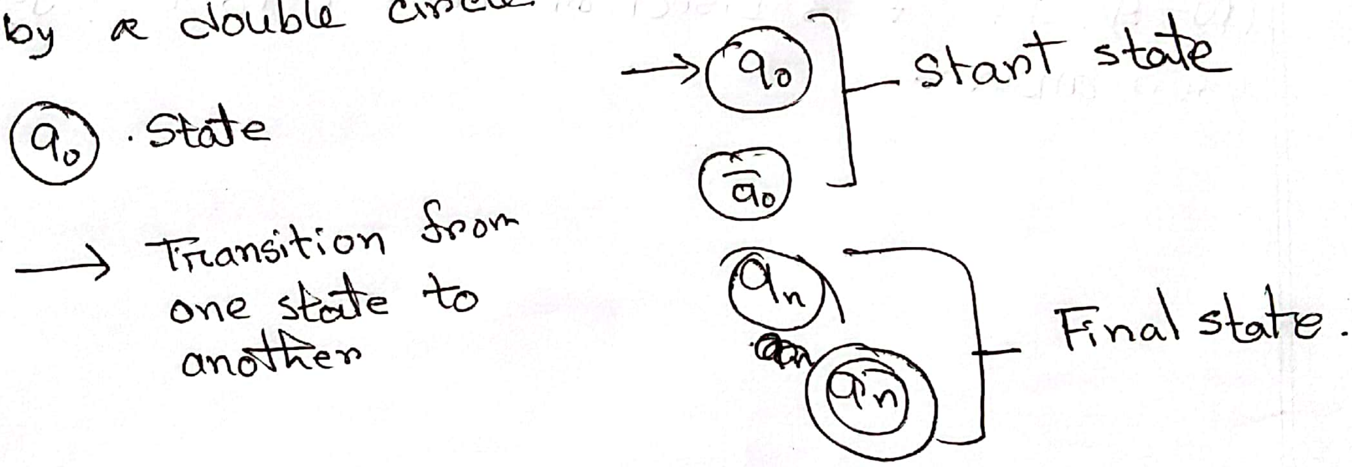
TD or state diagram is a directed graph which can be constructed as follow:-

① There is a node for each state in Q, which is represented by the circle.

② There is a directed edge from node q to node p labeled a if $\delta(q, a) = p$.

③ In the ~~state~~ start state, there is an arrow with no source.

④ Accepting states or final states are indicating by a double circle.



1(c) Prove by Mathematical Induction $1+2+3+\ldots+n = \frac{n(n+1)}{2}$

Let,

$P(n) = 1+2+3+\ldots+n$

Now $P(1) = 1 = \frac{1(1+1)}{2} = 1$

So, $P(1)$ is true.

Let us assume that $P(k)$ is true that is $1+2+\ldots+k = \frac{k(k+1)}{2}$ —①

We shall prove that $P(k+1)$ is true

Now, $P(k+1) = (1+2+3+\ldots+k)+(k+1)$

$= \frac{k(k+1)}{2} + (k+1)$

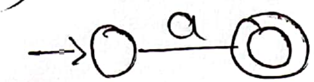
$= \frac{k(k+1) + 2(k+1)}{2}$

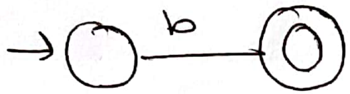
$= \frac{(k+1)(k+2)}{2}$

$= \frac{(k+1)(k+1+1)}{2}$

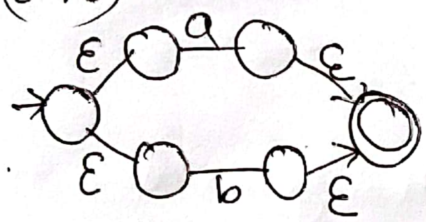
So, $p(k+1)$ is true when $P(k)$ is true.
Now by principle of mathematical induction we've curave $p(n)$ is true $\forall n \in N$.

2(a)(i) $(a+b)^*$

a 

b 

$(a+b)$



$(a+b)^*$