Subject Code:ACSE0404

NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY ,GREATER NOIDA

(An Autonomous Institute)

Affiliated to Dr. A.P.J. Abdul Kalam Technical University, Uttar Pradesh, Lucknow

Course- B.Tech Branch - AI/AIML/IOT/CS/DS

Semester- 4th Sessional Examination – 1st Year (2021-2022)

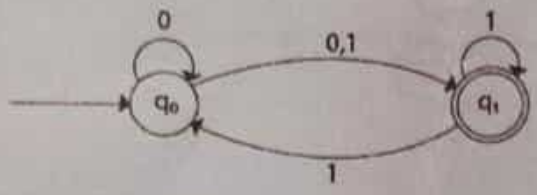
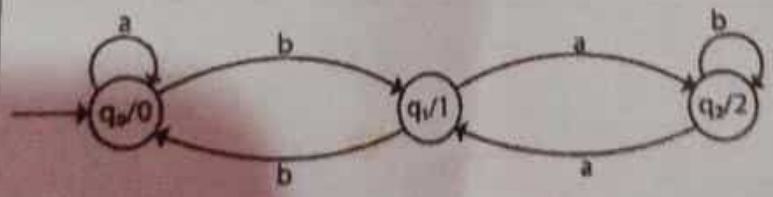
Subject Name: Theory of Formal Automata and Languages

Time: 1.15Hours Set-1 Max. Marks:30

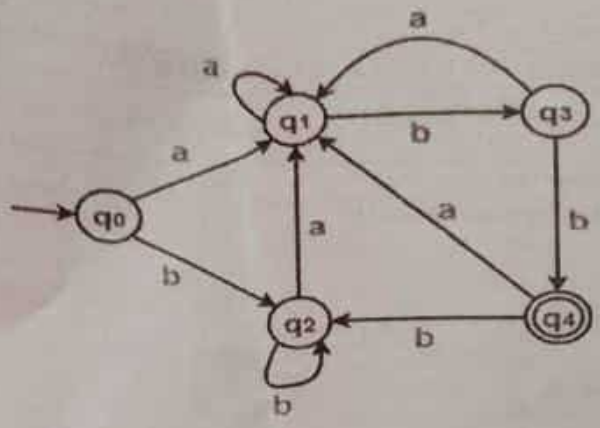
General Instructions:

- This Question paper consists of 03 pages & 05 questions. It comprises of three Sections, A, B, and C. You are to attempt all the sections.
- **Section A** -Question No- 1 is objective type questions carrying 1 mark each, Question No- 2 is very short answer type carrying 2 mark each. You are expected to answer them as directed.
- **Section B** - Question No-3 is Short answer type questions carrying 5 marks each. You need to attempt any two out of three questions given.
- **Section C** -Question No. 4 & 5 are Long answer type (within unit choice) questions carrying 6marks each. You need to attempt any one part *a* or *b*.
- Students are instructed to cross the blank sheets before handing over the answer sheet to the invigilator.

		SECTION - A	[8Marks]	
1.	Attempt all parts-		(4×1=4)	CO1
	a.	If minimal NFA has 'n' states then for same language minimal DFA in worst case can have how many states? A.n B.2n C.2^n D.n^2	(1)	
	b.	In DFA the transition function δ is given by a) $\delta: Q \times \Sigma \rightarrow Q$ b) $\delta: Q \times \Sigma \rightarrow 2^Q$ c) $\delta: Q \times q0 \rightarrow Q$ d) $\delta: Q \times q0 \rightarrow F$	(1)	
	c.	In Moore machine if we give string of length 'n' as an input, then output string will be of length: A. n B. 2n C. n+1 D. n-1	(1)	

	d.	Let $\Sigma = \{a,b,c\}$. How many strings are there in Σ^3 a) 27 b) 8 c) 243 d) 64	(1)	
2.	**Attempt all parts**		(2×2=4)	CO1
	a.	Differentiate between Σ^* and Σ^+.	(2)	
	b.	Explain String Acceptance and Language Acceptance in case of DFA.	(2)	
		SECTION – B	[10marks]	
3.	**Answer any two of the following-**		[2×5=10]	CO1
	a.	Convert following NFA to equivalent DFA:  	(5)	
	b.	Explain how complementing a language is different from complementing a NFA. Draw the transition diagram for the transition table given below: Present state | Next state on input 0 | Next state on input 1 →A | A | B B | B,C | B *C | B | {}	(5)	
	c.	Define symbol, alphabet, string and language. Design a DFA that reads string defined over $\Sigma=\{a,b\}$ and accepts only those strings in which every 'a' is followed by 'b'.	(5)	
		SECTION – C		
4	**Answer any one of the following-**		[2×6=12]	CO1
	a.	**Question-** Change the Moore machine into Mealy machine  	(6)	
	b.	**Question-**Give the DFA accepting the following languages over the alphabet {0, 1} such that:	(6)	

	(i) The set of all strings which starts with ' 0'. (ii) The set of all strings such that the binary number is divisible by 3.		
	Answer any one of the following-		
a.	Minimize the automata given below:	**(6)**	



b.	(i)Differentiate between epsilon-NFA and NFA. What are applications of Finite Automata? (ii)Draw the Mealy Machine that converts binary string into its 2's complement form . Assume string is read from least significant bit (LSB) to most significant bit (MSB).	**(6)**	

Date Page

THEORY OF AUTOMATA & FORMAL LANGUAGE (TAFL)

Theory of computation is a branch that deals what Problems can be solved on a model on computation using an algo. and how efficiently they can be solved.

L → language
A → Autometa (mathematical model)
G → Grammar.

Language :-

Symbol :- a to z, 0 to 9
Alphabet :- Σ : $\Sigma(a, b)$, $\Sigma(1,2)$, etc (finite Set of symbols)
String :- Sequence of Alphabets.
language :- collection of String (Finite / Infinite)

$\Sigma^* = (\lambda, \wedge, \epsilon)$ Empty

Name of languages :

1. RL (Regular lang.) $a^n \mid n \geq$

Date _____ Page _____

2. CFL (context Free lang) $a^n b^n | n \geq 1$
3. CSL (context Sensitive lang.) $a^n b^n c^n | n \geq 1$
4. REL (Recursive Enumerable lang)

Automata: mathematical Model

it is defined as whether a string a part of lang. or not

- FA (finite automata)
- PDA (Push down automata)
- LBA (Linear Bounded Automata)
- TM (Turing Machine)

Grammar: Rules

4 tuples → V – variable (A–Z)
↳ T – Termination (a–z)
↳ P – Production Rule
↳ S – Start Symbol 'S'

Q
$S \to SS$
$S \to aSb$
$S \to bSa$
$S \to \epsilon$

(i) bababa ?

{ bSa
ba basba
baba babSaba
bababa

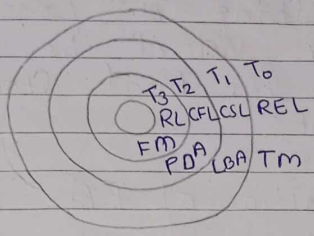
{ ∈, aSb
ab aaSbb abSab

$a^2 b^2$

Date _____ Page _____

CHOMSKY MODEL :-

T- Type.



Kleen closure

$$\Sigma^n = \Sigma^* = \Sigma^0 + \Sigma^1 + \Sigma^2 \dots + \Sigma^n$$

$\Sigma^0 = t$

$\Sigma^1 = a, b$

$\Sigma^2 = aa, ab, ba, bb$

$\Sigma^3 = ----$

Positive closure

$$\Sigma^+ = \Sigma^1 + \Sigma^2 + \dots \Sigma^n$$

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

Finite Automata :- Finite automata is the simplest machine to recognize pattern. It takes the string of symbol as input and changes its state accordingly. When the desired symbol is found then the transition occurs.

FSM = FA
Finite State Machine = Finite Automata

Date _____ Page _____

- Finite automata have two states, Accept State or Reject State. When the input string is processed successfully and the automata reached its final state. then it will accept.

Formal definition of FA :- A finite automata is a collection of 5-tuple $(Q, \Sigma, \delta, q_0, F)$

Q = finite set of states
Σ = finite set of the input symbol.
q_0 = initial state
F = Final state
δ = Transition function

Application of Finite automata :-

1. String Matching
2. lexical Analysis
3. Design & Analysis of digital circuits.

Types of Automata :- There are two types of automata.

1. DFA (deterministic finite automata)
2. NFA (Non-deterministic finite automata)

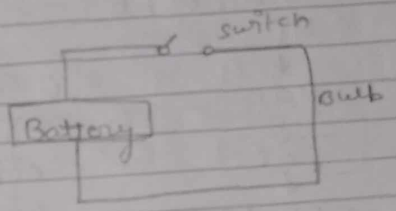
Date Page

Finite Automata

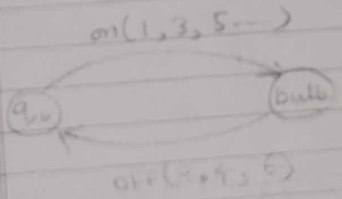
- Deterministic Finite Automata DFA
- Non-deterministic Finite Automata NFA

States Example

i/p :- Switch
o/p :- Bulb
Action :- Flip switch
State :- on, off



on - odd
off - even



Initial
Final
Stages

transition 1
one than

Date

Page

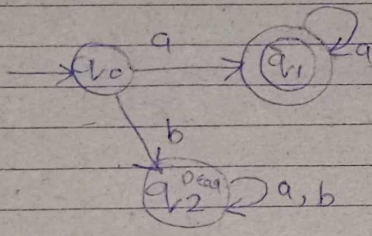
DFA - 5 Tuple

DFA (Deterministic finite automata):-

There are five tuples in DFA.
$(Q, \Sigma, \delta, q_0, F)$

Q - Finite set of states
Σ - Finite set of Input State
$\delta \rightarrow$ transition ($Q \times \Sigma \rightarrow Q$)
q_0 - ~~start~~ initial state
$F \rightarrow$ Final State.

Ex:- Strings starting with a {a, aa, aaa, ab, abb}

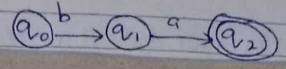


transition function:- $\delta: Q \times \Sigma \rightarrow \delta$
transition table

	a	b
q_0	q_1	q_2
q_1	q_1	q_1
q_2	q_2	q_2

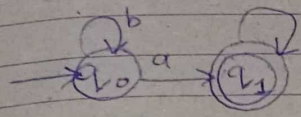
Date

Page

$w = ba$ 

Q1 construct a DFA which accept a language of all strings. containing 'a'
$\Sigma = (a, b)$

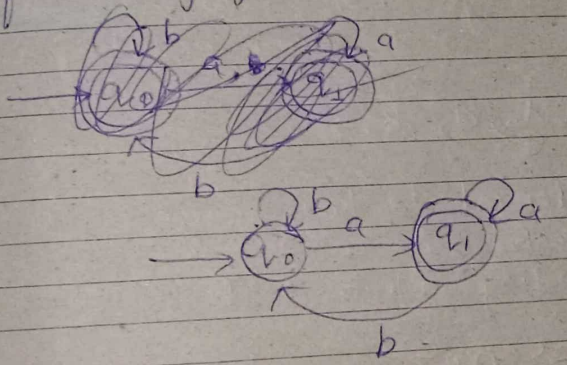
$L = \{ a, aa, aaa, ba, ab, abba, abab$ -- infinite string.



q_0 से b नहीं bhejge kuki agar b jayega to singal b bhi accept ho jayega.

Q End with a

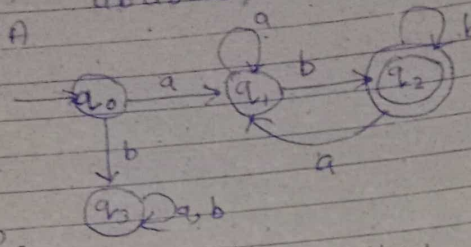
$L = \{ a, aa, aaa, ba, baa, bba, baba$ -- infinite laguage.



Q Construct a DFA ~~with~~ which accept a language of all strings starting with a and ending with b.

$L = \{ab, aabb, aab, abb, abab, ababab, aaaabbbb\}$

DFA



aaaabbbbab

Q Construct a DFA which accept a language of all strings not starting with a or not ending with b

$L = \{\lambda, \epsilon, a, b, ba\}$

Complement le kar karege.

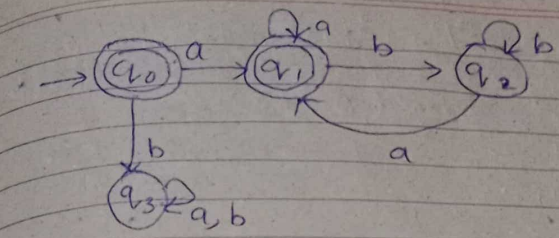
Let - not starting with a → A

" ending with b - B

$(A \cup B)^c = A^c \cap B^c$

starting with a and ending with b

Create dfa same as starting with a and ending with b and change all final state into non final and all non final state into final state.

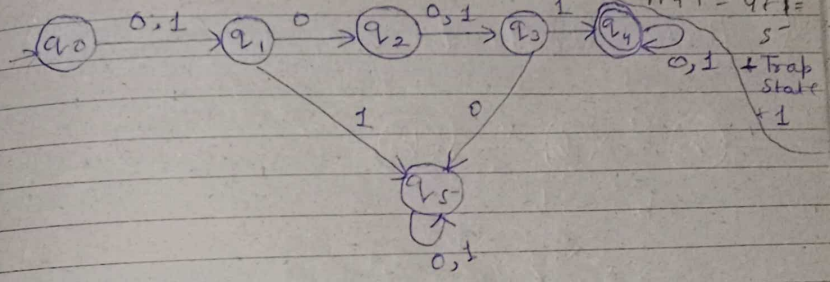


Q Design a DFA which accepts all string over {0, 1} in which second symbol is 0 and fourth symbol 1.

0/1 0 0/1 1

minimal length = 4

minimum states = n+1 = 4+1 = 5 + Trap State = 6



Q construct a DFA with accept a language of all binary string divisible by three over {0, 1}

Kisi bhi no. ko 3 se divide karne par remainder hamesa 0, 1, 2 hi ayega.

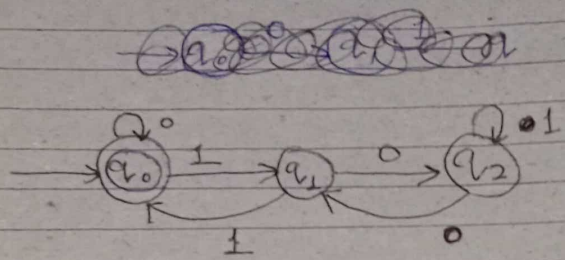
q_0 q_1 q_3

Q construct a DFA which accepts a language of all binary strings divisible by three over $\Sigma\{0,1\}$

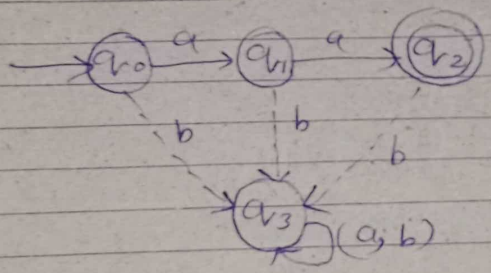
remainder 0, 1, 2

q_0, q_1, q_2

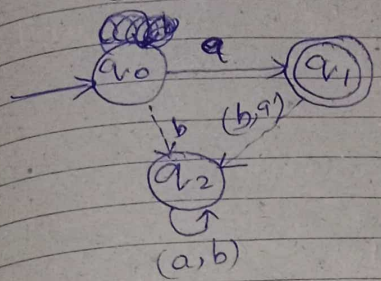
0 0 0
0 0 1
0 1 0
0 1 1



Q There should be aa when $\Sigma(a, b)$

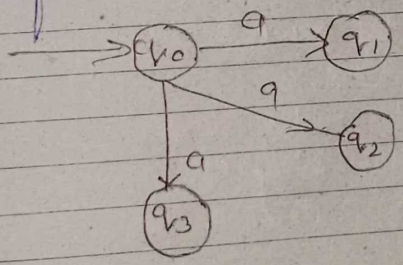


Q create a DFA accepting a
$\Sigma(a,b)$: a



NFA (Nondeterministic finite automata):-

NFA stands for non-deterministic finite automata. it is easy to construct an NFA than DFA for a given regular language.
IN NFA. the machine can move from a present state to a combination of state for an Input as the no. of States are finite.



Formal definition of NFA :- NFA Also has five States same as DFA but with different transition fuction.

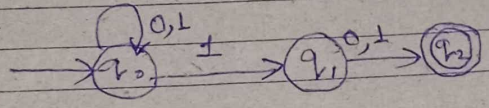
$$\delta : Q \times \Sigma \rightarrow 2^Q$$

Where :-

Q : finite Set of States
Σ : finite Set of the Input Symbol.
q_0 : initial State
F : Final State
δ : transition function.

Q NFA of all binary string in which 2nd last bit is 1

$\Sigma(0,1)$
min String {10, 11} (2nd last bit digit) min states find formula $(n+1)$

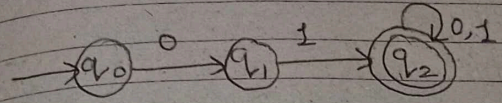


min state find
$n+1$ ⇒ min string +1
⇒ 2+1=3
3 (min State)

Laglage : $(0+1)^* 1 (0+1)$

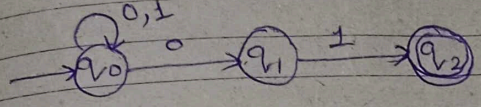
Q Design a NFA over $\Sigma\{0,1\}$, such that it accepts every string starting with 0,1.

$L = \{01, 010, 011, 0111 \ldots\}$



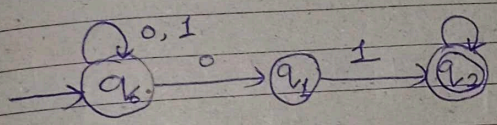
Q end with 01

$L = \{01, 001, 101 \ldots\}$



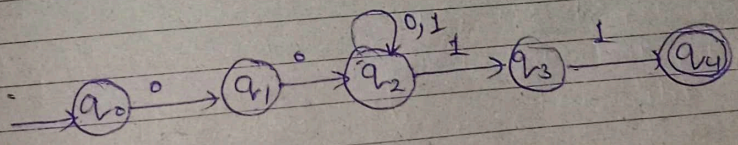
Q Contain 01.

$L = \{01, 0011, 001, 101 \ldots\}$



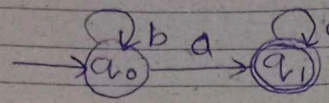
Q Design a NFA with Start with 00 & end with 11

$L = \{0011, 00011, 00111 \ldots\}$



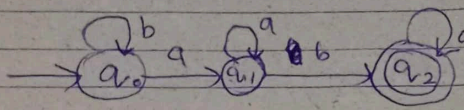
Q containing a → NFA

L = { a, aa, ab, bab, baa ... }



Q containing ab.

L = { ab, abb, aab, aabb ... }



n+1 ≠ 2+1
minState = 3

Difference between NFA & DFA :-

DFA	NFA
1) Dead configuration is not allowed	1. Dead configuration is allowed.
2). Multiple choices are not available corresponding to an Input.	2.) Multiple choices are available corresponding to an input.
3) ∈ move is not allowed.	3. ∈ move is allowed.
4) designing and understanding is difficult.	4. Designing and understanding is easy.

q_2	L_2	q_1
q_0		q_1

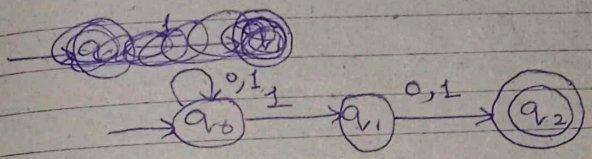
5) All DFA are NFA
6) DFA required more space
7) Dead state maybe required.
8) $\delta: Q \times \Sigma \to Q$

5. Not all NFA are dFA.
6) NFA required less space than DFA.
7) Dead State is not required.
8) $\delta: Q \times \Sigma \to 2^Q$

Conversion of NFA to DFA.

Q construct of all binary string in which 2nd last bit is 1. $\Sigma(0,1)$.

10 } minimum
11 } string.



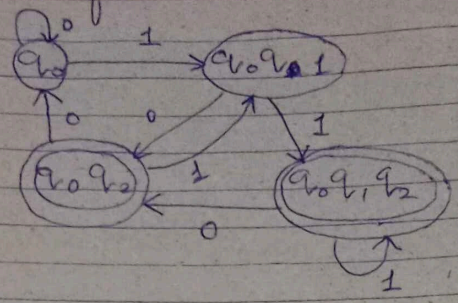
transition table:-

	0	1
q_0	q_0	$q_0 q_1$
q_1	q_2	q_2
q_2	–	–

convert into (DFA) - make transition table.

	0	1
q_0	q_0	$q_0 q_1$
$q_0 q_1$	$q_0 q_2$	$q_0 q_1 q_2$
$q_0 q_2$	q_0	$q_0 q_1$
$q_0 q_1 q_2$	$q_0 q_2$	$q_0 q_1 q_2$

make DFA from the transition table:-

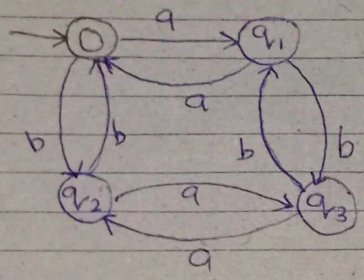


DFA

(means done condition true.

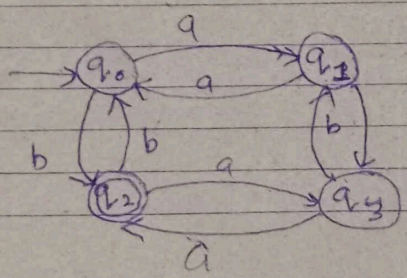
Q even no of a and even no of b

L = {ϵ, aa, bb, abab, aabb ---- }



Q even a and odd no of b.

L = { b, aab, aaaab ---- }



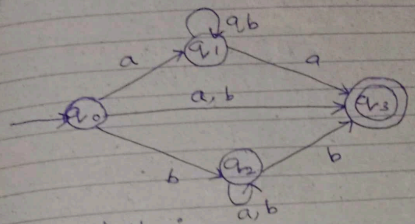
odd a odd b (q_3) final

even a, add b → (q_2) final

odd a, even b → (q_1) final.

Q Convertion from NFA to DFA
→ Start and end with same Symbol.

$L = \{a, b, aa, bb, aba, bab \dots\}$

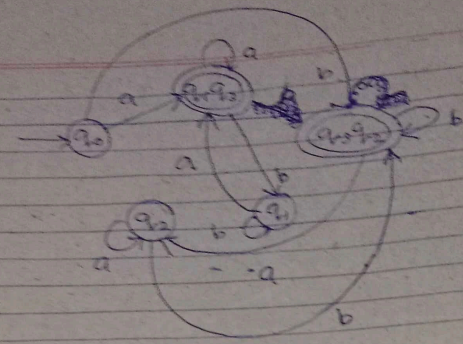


transition table :-

	a	b
→ q_0	q_1, q_3	$q_3 q_2$
q_1	$q_1 q_3$	q_1
q_2	q_2	$q_2 q_3$
final q_3	—	—

make DFA transition table.

	a	b
q_0	q_1, q_3	$q_3 q_2$
$q_1 q_3$	$q_1 q_3$	q_1
$q_3 q_2$	q_2	$q_2 q_3$
q_1	q_1, q_3	q_1
q_2	q_2	$q_2 q_3$

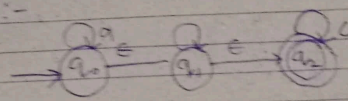


Epsilon NFA :- (ϵ-NFA) :-

Q :- Finite set of ~~by~~ states.
Σ : Finite Set of input symbol.
q_0 : initial state
F : Set of Final States
δ : transition function

$$\delta : Q \times (\Sigma \cup \epsilon) \rightarrow 2^{Q}$$

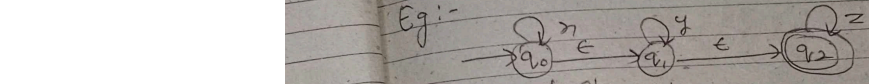
Eg :-



1) The automata may be allow to change its state without reading the input symbol.
2) Null(ϵ) symbol doesn't belong to any alphabet (Σ)

Epsilon Closure :- epsilon closur (q_0) set of states reachable from q on epsilon transition alone.

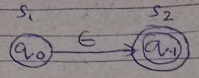
Eg:-



Find ϵ-Closure :-

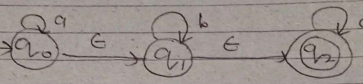
ϵ-clo. q_0 = $\{q_0, q_1, q_2\}$
" q_1 = $\{q_1, q_2\}$
" q_2 = $\{q_2\}$.

Conversion E-NFA to NFA
(with ϵ-NFA to without ϵ NFA)



1) find all the edges starting from S_2 without
2) Duplicate all edges to S_1 changing edge labels.
3) if S_1 is initial state, make S_2 initial.
4) if S_2 is final state, make S_1 final.

Ex:-



Firstly find Eclosure:-

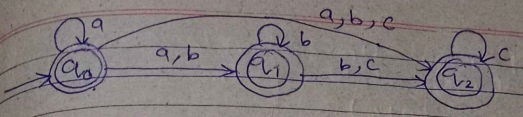
$q_0 = \{q_0, q_1, q_2\}$
$q_1 = \{q_1, q_2\}$
$q_2 = \{q_2\}$

NFA Transition table:-

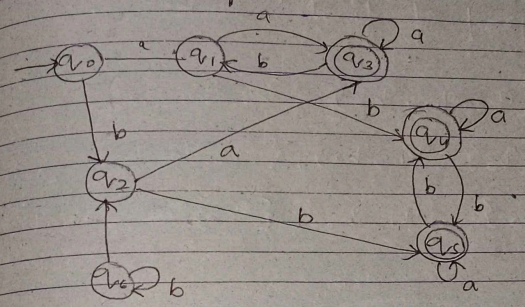
	a	b	c
q_0	$q_0 q_1 q_2$	$\{q_1 q_2\}$	q_2
q_1	—	$q_1 q_2$	q_2
q_2	—	—	q_2

q_2 aur final then q_2 liske sath bhi ayega vo hamara final state hogi (creat check from ϵ closure)

then create NFA.



Minimization of DFA :-



transition table :-

	a	b
q_0	q_1	q_2
q_1	q_3	q_4
q_2	q_3	q_5
$*q_3$	q_3	q_1
$*q_4$	q_4	q_5
$*q_5$	q_5	q_4
q_6	q_2	q_6

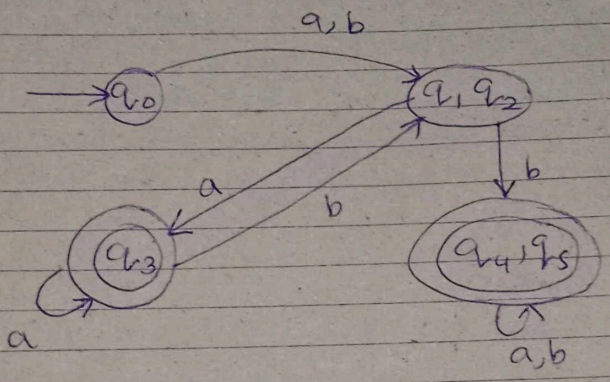
Step 1 - Remove Unreachable State (Unreachable from Initial State).

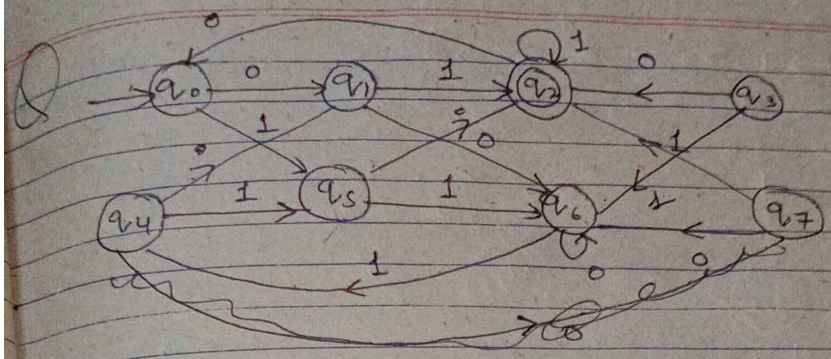
Step 2:- Divide States in two classes.
C_1 : all state except final state.
C_2 : Group of final States.

$\pi_0 = \{q_0, q_1, q_2\}\ \{q_3, q_4, q_5\}$

$\pi_1 = \{q_0\}\ \{q_1, q_2\}, \{q_3, q_4, q_5\}$

$\pi_2 = \{q_0\}\ \{q_1, q_2\}, \{q_3\}\ \{q_4, q_5\}$





Transition table.

	0	1
q_0	q_1	q_5-
q_1	q_6	q_2*
*q_2	q_0	q_2*
q_3	q_2*	q_6
q_4	q_6	q_5-
q_5-	q_2*	q_6
q_6	q_6	q_4
q_7	q_6	q_2*

q_3 not reachable. State:-

0 equivalent

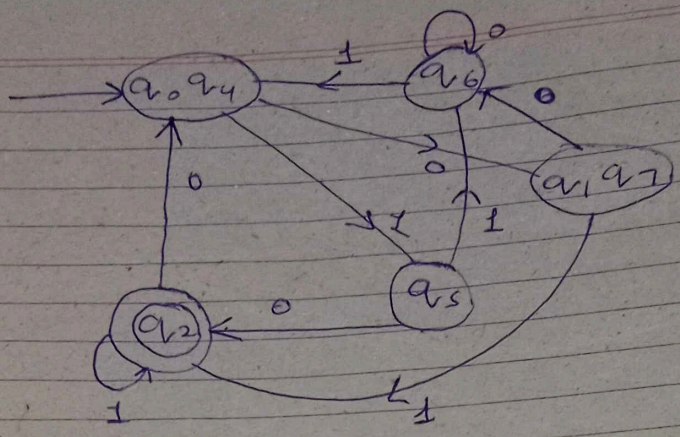
$\{q_0, q_1, q_4, q_5, q_6, q_7\}\ \{q_2\}$

1 equivalent

$\{q_0, q_4, q_6\}\ \{q_1, q_5, q_7\}\ \{q_2\}$

$\{q_0, q_4, q_6\}\ \{q_1, q_7\}\ \{q_5\}\ \{q_2\}$

$[q_0, q_4]\ [q_6]\ [q_1, q_7]\ [q_5]\ [q_2]$

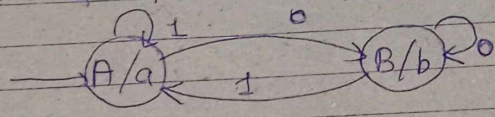


Finite automata with outputs :-

1) Mealy Machine
2) Moore Machine.

1) Moore Machine :- Tuples → 6
$(Q, \Sigma, q_0, \Delta, \lambda, \delta)$

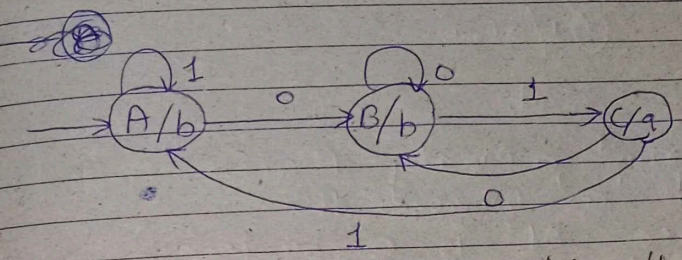
Q → finite set of states
Σ → finite set of I/P symbol.
q_0 → initial state
Δ → output symbol
λ → output function : $Q^* \rightarrow \Delta$
δ → transition function : $Q \times \Sigma \rightarrow Q$



Q construct a moore machine that prints 'a' whenever the sequence 01 is encountered in any input binary string.

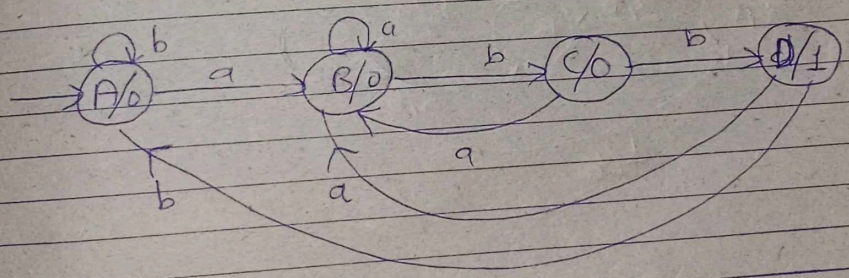
$\Sigma = \{0, 1\}$

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



Q Construct a Moore Machine that counts the occurences of the Sequence 'abb' in any input String $\{a, b\}$

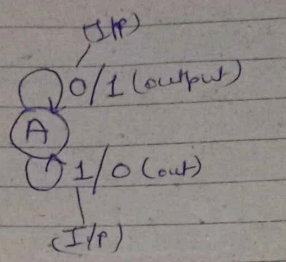
Let $\Sigma = (a, b)$, $\Delta = \{0, 1\}$



Construction of Mealy Machine

Q Construct a Mealy Machine that Produces the 1's complement of any binary input string.

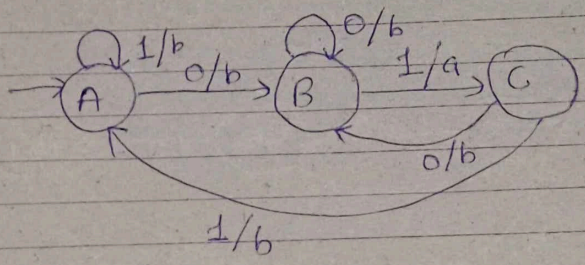
0 → 1 [complement]
1 → 0



Q Construct a Mealy Machine that prints 'a' whenever the sequence '01' is encountered in any input binary string.

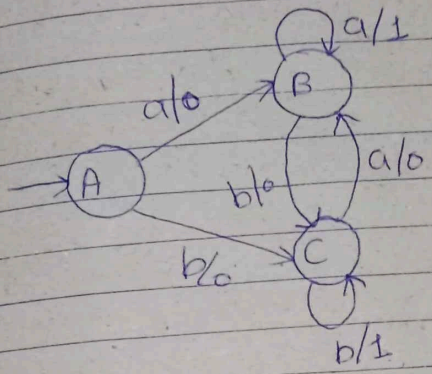
$\Sigma = \{0, 1\}$

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



Q Design a Mealy Machine accepting the lang consisting of strings from Σ^*, whe $\Sigma = \{a, b\}$ and the strings should end either aa or bb

aa — 1
bb — 1



a, b

Date

Page

Unit - 2.

Type 1:

→ context Sensitive grammar

→ Linear bounded automata :-

→ CSL

Type 0

→ Unrestricted grammar

Turing Machine

→ Recursively enumerate Set.

Type 2

context free grammer

CFL

PDA

Type 3

Regular grammar

RL

Finite automata.

Regular Expression :-

↓

Regular language

(Grammar which generate Regular language is called regular grammar)

generator

↓

Regular grammar

↓ RLA (Right linear grammar)

↓ LLG (Left linear grammar)

acceptor (Given a string it will say la is there o not)

↓

FA

↓ with output → Mealy, Moore

↓ without output → DFA, NFA, ε-NFA

Date Page

Regular expression :-

→ It is a representation by a language which are exactly accepted by finite automata.

→ Basically it contains three operations (i) Union(+), (ii) concatenation(.) (iii) kleen closure. (*).

$\emptyset = \{\}$
$\epsilon = \{\epsilon\}$
$a = \{a\}$

a-z = Terminal
A-z = Variables.

$a^* = \{\epsilon, a, aa, ---\}$
$a^+ = aa^*, a^*a$
$(a+b)^* = \{\epsilon, a, b, aa, ab, aba\}$
↓ we can create any no. combination of a and b.

length exactly two.

$L = \{aa, ab, ba, bb\}$
$aa + ab + ba + bb$
$a(a+b) + b(a+b)$
$R.E = (a+b)(a+b)$

Date _____ Page _____

Q Length atleast two

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

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

Q Length atmost two.

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

$\epsilon + a + b + aa + ab + bb + ba$

$(\epsilon + a + b)(\epsilon + a + b)$

Set of all the strings even length.

$L = \{ \epsilon, aa, ab, bb, ba, baba, bbbb \cdots \}$

$\epsilon + aa + ab + bb + ba + baba + bbbb$

$(aa + bb + ba + bb)^*$ or $[(a+b)(a+b)]^*$

Q Set of all the strings odd length.

$L = \{ \epsilon, a, b, aaa, bbb, aab, abb \}$

$(a+b)[(a+b)(a+b)]^*$

Q Set of all strings divisible by 3.

$L = \{ \epsilon, aaa, aab, ~~abb~~ bbb, abb, 0, 3, 6, 9 \}$

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

Q Set of all strings whose length is $\cong 2 \bmod$

Date Page

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

$3n+3$

$(a+b)$

$(a+b)^{3n}\ (a+b)^2$

$((a+b)(a+b)(a+b))^n (a+b)(a+b)$

$((a+b)(a+b)(a+b))^*\{(a+b)(a+b))$

1. Number of a's exactly two; $\Sigma = \{a, b\}$

   $L = \{aa, baa, aba, baba \cdots\}$

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

2. a's atleast two.

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

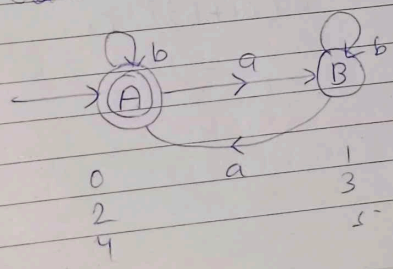
3. no. of a's atmost two ? 0, 1, 2

   $b^* (\epsilon + a)\ b^* (\epsilon + a)\ b^*$

4. Set of all strings when number of a's is even.

   $L = \{a, b, aab, abb, abab \cdots\}$

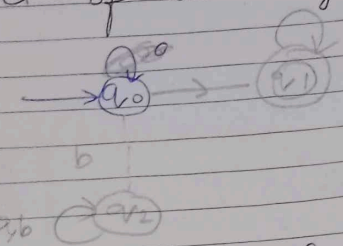
   $b^* a b^* b^* a b^*)^* + b^*$



Date

Page

Set of all String Start with a.



$a(a+b)^*$

Set of all String end with a.

$(a+b)^* a$.

Set of all String containing with a

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

Start and end with Same Symbol.

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

(either this one or this one.)

Start and end with different Symbol.

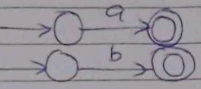
$a(a+b)^* b + b(a+b)^* a$.

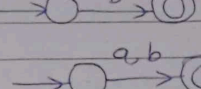
Conv

Date _____ Page _____

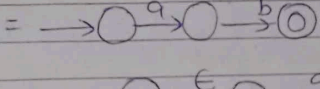
Convert regular expression into finite automata:-

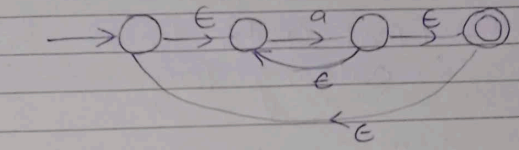
$\phi = \{\}$

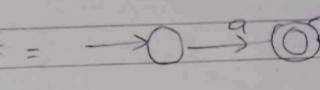
$a =$ 

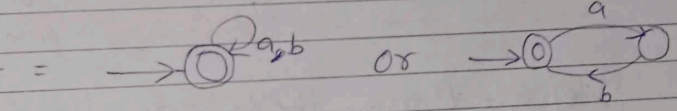
$b =$ 

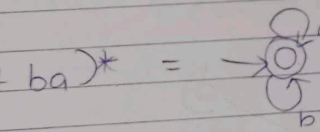
$a+b =$

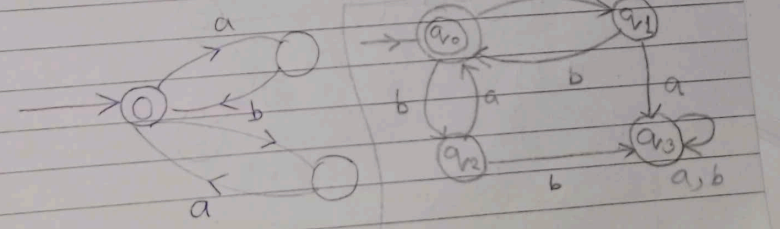
$a.b =$ 

$a^* =$ 

$a.b^* =$ 

$(ab)^* =$ 

$(ab+ba)^* =$ 

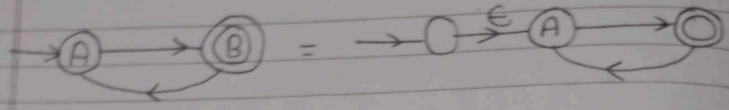


Date Page

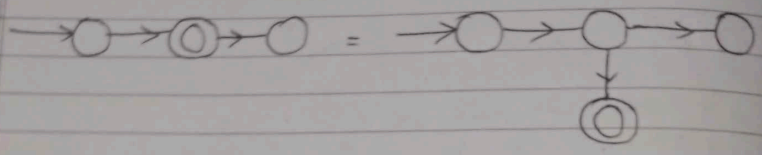
Finite automata to regular expression:-

- Using state elimination method :-

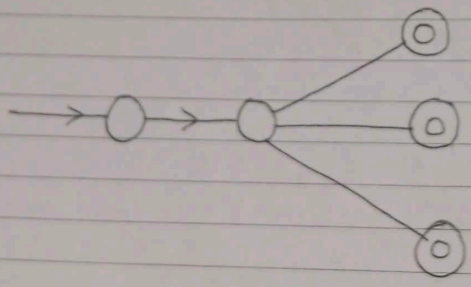
Step 1- Initial state has no incoming edge.



Step 2:- Final state doesn't have any outgoing Edge.

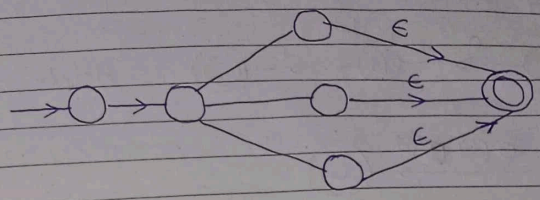


Step 3:- More than one final state.

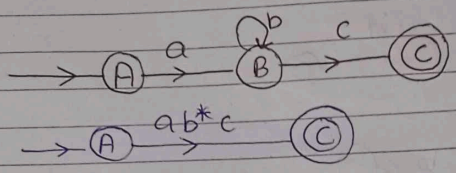


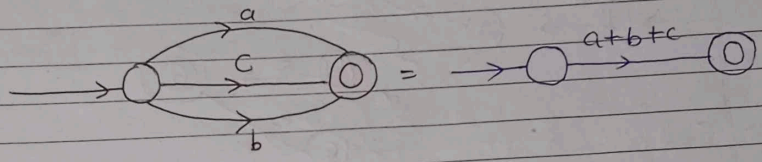
line multiplication | ↑ (addition)

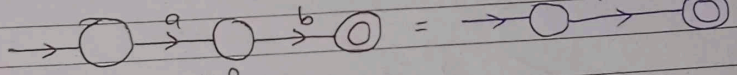
Date _____ Page _____

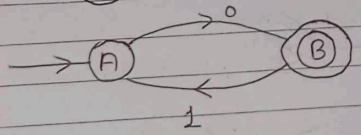


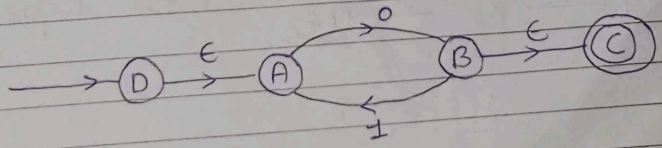
Step 4:- Eliminate all Stages other than initial State & final State.



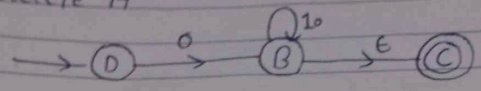
Q1 

Q2 

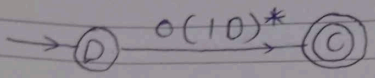
Q3 



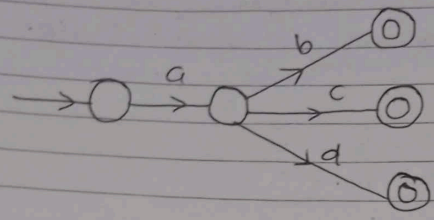
delete A

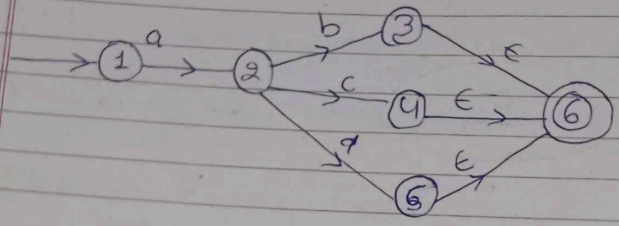


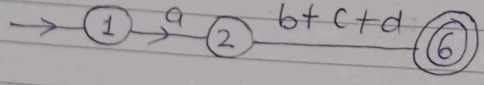
delete B

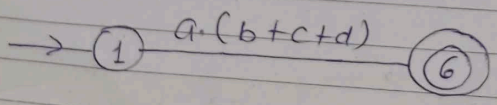


Q2







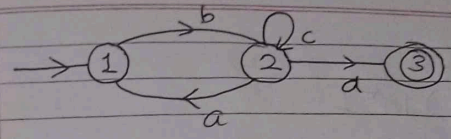


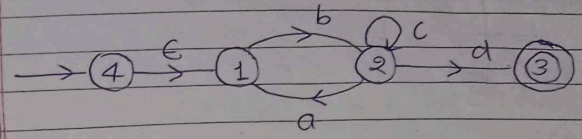
final expression.

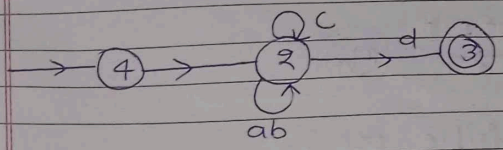
Date

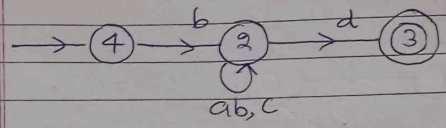
Page

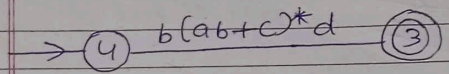
Q3



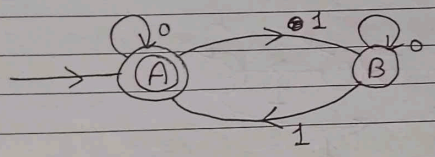








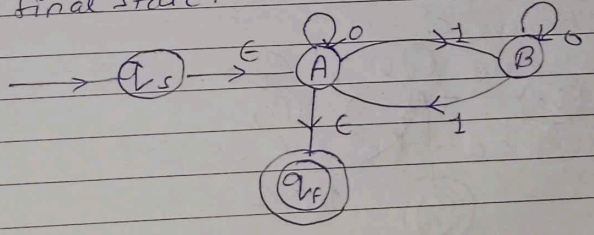
(i)



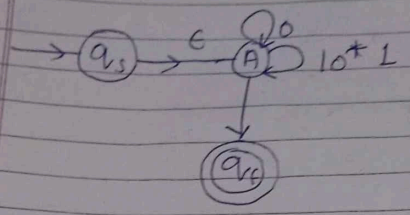
even no. of 1's

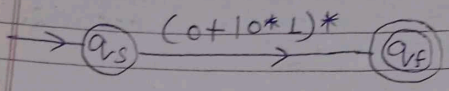
q_s = initial state

q_f = final state.

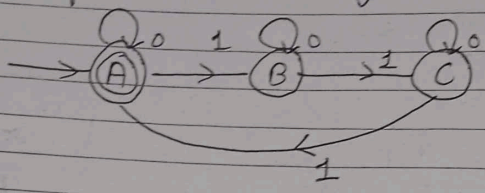


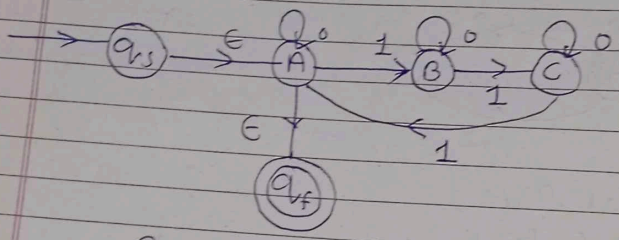
Date Page



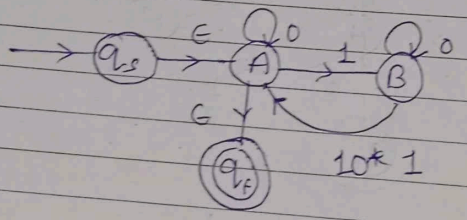


Q No. of 1's divisible by 3.



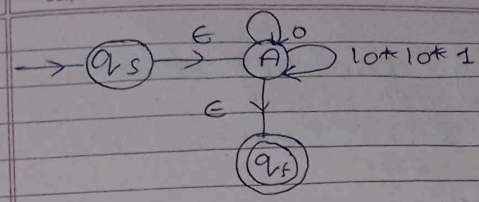


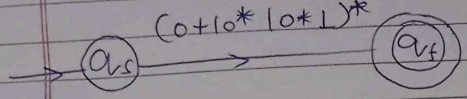
Remove 'C'.



Remove B.

Date _____ Page _____





language Regular or not finite or not

Test Wheather language is RE or not

$L = \{ab, abab, ababab \ldots\}$

Finite = Regular

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

Finite hai to finite automata bana sakta hai

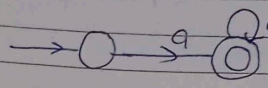
→ Pattern ban raha hai tab RE or not done ho sakta hai.

Otherwise pattern nahi ban raha hai RE(X)

Date Page

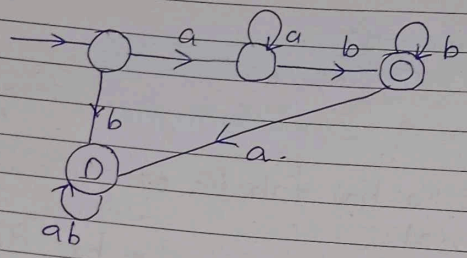
This is say by "pumpping Lemma", that language is not regular if you that not found such pattern but doesn't generate that language is regular (negativity test)

a. $a^n / n \geq 1$



— Regular.

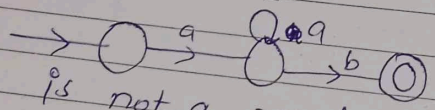
b. $a^n b^m / n, m \geq 1$



Regular.

c. $a^n b^n / n \leq 10^{10^{10^{10^{10}}}}$ Regular

$a^n b^n / n \geq 1$



is not a regular language for this expression.

Date

Page

Q ww^R / $|w| = 2$ (Reverse)

length is given to you so that expression is a regular lang.

Q ww^R / $w \in (a+b)^*$

length is not given to you. This goes to infinite

this is not regular language.

Q ww / $w \in (a,b)^*$

not regular lang.

Q $a^n b^m c^k$ / $n, m, k \geq 1$

$L = \{a, b, c\}$

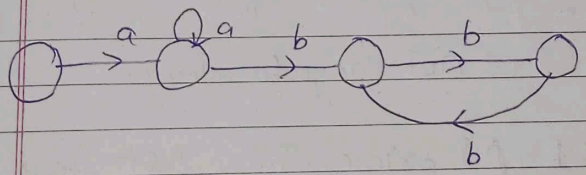
Regular expression

$aa^* bb^* cc^*$

Q $a^i b^{2j}$ / $ij \geq 1$ Regular lang.

Regular expression $\Rightarrow aa^* bb(bb)^*$

DFA :-



Page

Date

① $a^i b^{4j}$ Regular lang.

Regular expression:

~~$aa^*(bbbb)^*$~~

$aa^* bbbb(bbbb)^*$

② a^n / n is even.

③ a^n / n is odd.

④ a^n / n is prime. not regular.

$L = \{2, 3, 5, 7, \ldots\}$

⑤ a^{n^2}, $n \geq 1$ — not regular lang.

⑥ a^{2^n}, $n \geq 1$ not regular lang.

⑦ $a^i b^p$ / $i \geq 1$, p is prime no. = not regular.

Date _____ Page _____

- $\omega\omega\omega^R$ / $\omega \in (a,b)^*$ not regular
- $a^n b^n c^n$ / $n \geq 1$ not regular
- $a^n b^{n+m} c^n$ / $n \geq 1$ not regular.
- $\omega x \omega^R$ / $\omega, x \in (0,1)^+$ Regular lang. finite automata.
- $\omega / n_a(\omega) = n_b(\omega)$ = not regular lang.
- Describe the following sets of regular expression ?
- Set of all string ending with ~~0 Starting with 0~~ 00 and $\Sigma = (0,1)$

$(0+1)^* 00$

- Set of all string ending with 1 staring with 0

$0(0+1)^* 1$

- $\{\epsilon, 11, 111, 111111 - - -\}$

$(11)^*$

- $\{101\} = 101$
- $\{abba\} = abba$.

Page

Date

Q {01, 10} = 01 + 10

Q {∈, ab} = ∈ + ab

Q {abb, a, b, aba} =
abb + a + b + aba

Q {1, 11, 111, 1111 - -} = 11* or 1*

Q 8
i) 011 ⇒ (011 + 1)* (either this or that)
ii) 1, 11

Identities :-

$I_1 = \phi + R = R$
$I_2 = \phi R = R\phi = \phi$
$I_3 = \Lambda R = R\Lambda = R$
$I_4 = \Lambda^* = \Lambda$ and $\phi^* = \Lambda$
$I_5 = R + R = R$
$I_6 = R^* R^* = R^*$
$I_7 = RR^* = R^* R$
$I_8 = (R^*)^* = R$
$I_9 = \Lambda + RR^* = R^* = \Lambda + R^* R$
$I_{10} = (PQ)^* P = P(QP)^*$
$I_{11} = (P+Q)^* = (P^* Q^*)^* = (P^* + Q^*)^*$
$I_{12} = (P+Q)R = PR + QR$
and $R(P+Q) = RP + RQ$

Date Page

Q Prove that the regular ex. are $R = \Lambda + 1^*(011)^*(1^*(011)^*)^*$ also describe the same set of strings:

$R = \Lambda + 1^*(011)^*(1^*(011)^*)^*$

$R = \Lambda + P_1 P_1^*$ using Ig.

$= P_1^*$

$(1^*(011)^*)^* \rightarrow (P_2^* P_3^*)^*$ using Ig11

$P_2 \quad P_3$

$(P_2 + P_3)^*$

$(1 + 011)^*$

Q $(1 + 00^*1) + (1 + 00^*1)(0 + 10^*1)^*$

$(0 + 10^*1) = 0^*1(0 + 10^*1)^*$

12. $(1 + 00^*1)[\Lambda + (0 + 10^*1)^*(0 + 10^*1)]$

$(1 + 00^*1)[\Lambda + R^*R]$

g → $R^* = (0 + 10^*1)^*$

$(1 + 00^*1)[0 + 10^*1]^*$

Date _____ Page _____

* Transection System containing ϵ - moves:-

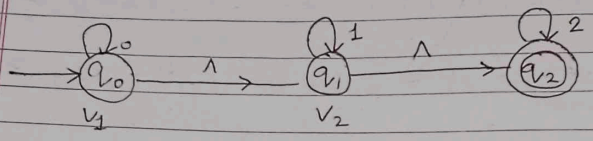
Step 1. Find all the edges starting from V_2.

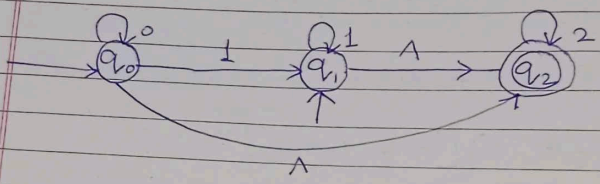
Step 2. Duplicate all edges starting from V_1. without change edge labels.

Step 3. if V_1 is initial make V_2 also initial state.

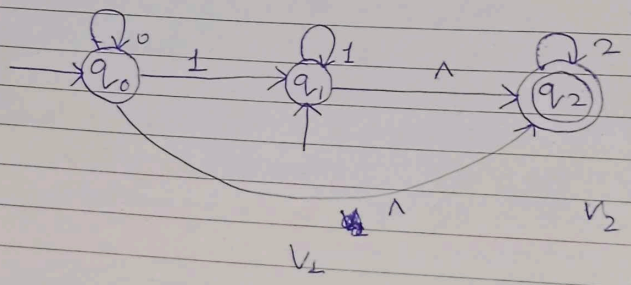
Step 4. if V_2 is final make V_1 also final state.

Q1

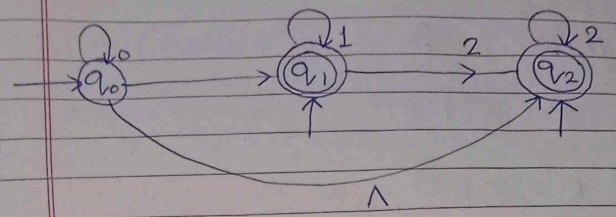


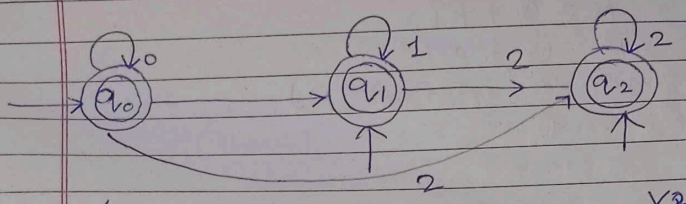


Step 3.

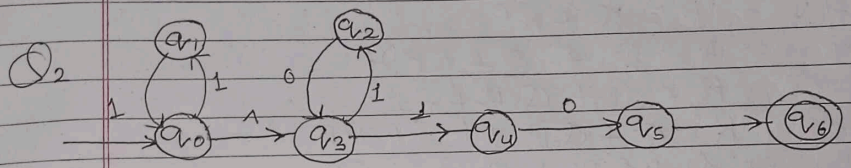


Date _____ Page _____

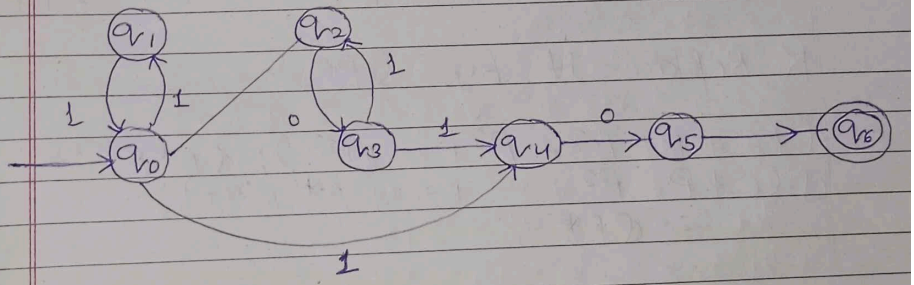




V_1 V_2



V_1 V_2



Date _____ Page _____

Arden's Theorem :-

$$R = Q + RP$$

$$R = QP^*$$

1st Proof-

$$R = Q + RP$$
$$R = Q + QP^* P$$
$$R = Q(\Lambda + P^* P)$$
$$R = QP^*$$

2nd Proof :-

$$R = Q + RP$$
$$R = Q + (Q + RP)P$$
$$R = Q + QP + RP^2$$
$$R = Q + QP + (Q + RP)P^2$$
$$= Q + QP + QP^2 + RP^3$$
$$= Q + QP + QP^2 + (Q + RP)P^3$$
$$= Q + QP + QP^2 + QP^3 + RP^4$$
$$= Q(\epsilon + P + P^2 + P^3 - \cdots + P^n) + RP^{n+1}$$

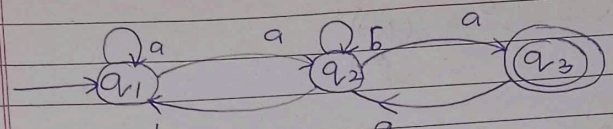
R Replace it by QP^*

$$Q(\epsilon + P + P^2 + P^3 + \cdots + P^n) + QP^* P^{n+1}$$
$$Q(\epsilon + P + P^2 + P^3 + \cdots + P^* P^{n+1})$$
$$= QP^*$$

$$\boxed{R = QP^*}$$

Date _____ Page _____

Q Proof that the String recognize are $(a+a(b+aa)^*b)^* a(b+aa)^* a$



In q_3 we have to write incoming arrow

$q_1 = aq_1 + bq_2 + \Lambda$ ——— (i)
$q_2 = bq_2 + aq_3 + aq_1$ ——— (ii)
$q_3 = aq_2$ ——— (iii)

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

$q_2 = aq_1 + bq_2 + aq_3$
$q_2 = aq_1 + bq_2 + q_2aa$
$q_2 = aq_1 + q_2[b+aa]$
$R = Q + RP$

$q_3 = aq_2$ put in eqn (ii)

$[q_2 = q_1 a(b+aa)^*]$

Put in eqn (i)

$q_1 = aq_1 + q_1a(b+aa)^* b + \Lambda$ (null)
$q_1 = q_1(a + a(b+aa)^* b) + \Lambda$
$R = RP + Q$

$q_1 = \Lambda(a + a(b+aa)^* b)^*$ Remove null
$q_1 = (a + a(b+aa)^* b)^*$

we have put q_1 value in q_2

$q_2 = a((a + a(b+aa)^* b)^*(b+aa)^*$

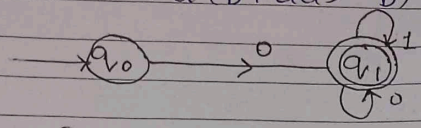
Date _____ Page _____

we put q_2 in q_3

$q_3 = aq_2$

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

(9)



$q_0 = \Lambda$

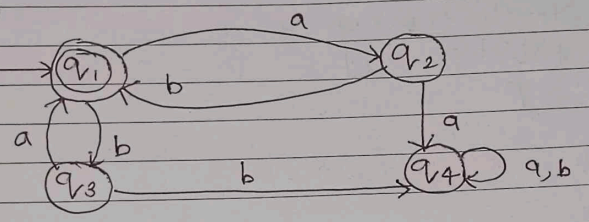
$q_1 = q_0 0 + 1q_1 + 0q_1$

$q_1 = q_0 0 + q_1(1+0)$

$R = Q + RP$

$q_1 = q_0 0 (1+0)^*$

(8)



$q_1 = \epsilon + q_2 b + q_3 a$

$q_2 = q_1 a$

$q_3 = q_1 b$

$q_4 = q_3 b + q_2 a + q_4 a + q_4 b$

putting q_2, q_3 in eqn q_1

$q_1 = \epsilon + q_1 ab + q_1 ba$

$q_1 = \epsilon + q_1 (ab + ba)$

$R = Q + RP$

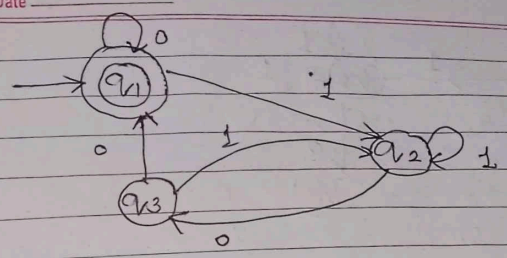
$q_1 = \epsilon + (ab + ba)^*$

$\boxed{q_1 = (ab + ba)^*}$

Date

Page

Q



$q_1 = \epsilon + 0q_3 + 0q_1$

$q_2 = 1q_1 + 1q_2 + 1q_3$

$q_3 = 0q_2$

putting q_1 and q_3 in eqn q_2

$q_2 = 1q_1 + 1q_2 + 1q_3$

$q_2 = 1(0q_1 + 0q_3) + 1q_2 + 1(0q_2)$

$q_2 = 01q_1 + 01q_3 + 1q_2 + 01q_2$

$q_2 = q_1 1 + q_2 1 + q_2 01$

$q_2 = q_1 1 + q_2 (1 + 01)$

$R = Q + RP$

$q_2 = q_1 1 + (1 + 01)^*$

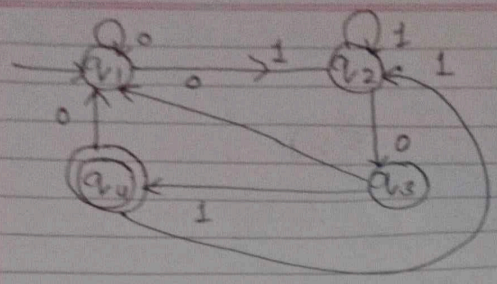
$q_1 = \epsilon + 0q_1 + 00q_2$

$= \epsilon + q_1 0 + q_1 1 (1+01)^* 00$

$q_1 = \epsilon + q_1 (0 + (1+01)^* 00)$

$R = Q + RP$

$= \epsilon + (0 + (1+01)^* 00)$



$q_1 = \epsilon + q_1 0 + q_3 0 + q_4 0$

$q_2 = 1 q_1 + q_2 1 + q_4 1$

$q_3 = 0 q_2$

$q_4 = q_3 1$

putting q_3 in q_4

$q_4 = 0 q_2 1 = q_2 01$

putting q_4 in q_2

$q_2 = 1 q_1 + q_2 1 + q_2 011$

$q_2 = 1 q_1 + q_2 (1 + 011)$

$R = Q + RP$

$q_2 = 1 q_1 (1 + 011)^*$

putting q_2 in q_3

$q_3 = q_1 1 (1 + 011)^* 0$

$q_1 = \epsilon + q_1 0 + 00 q_2 + q_2 010$

$q_1 = q_1 0 + q_1 1 (1 + 01)^* 00 + q_1 1 (1 + 010)^* 010 + \epsilon$

$q_1 = q_1 (0 + 1 (1 + 011)^* 00 + 1 (1 + 011)^* 010) + \epsilon$

$R = RP + Q$

$q_1 = \epsilon (0 + 1 (1 + 011)^* 00 + 1 (1 + 011)^* 010)$

Date

Page

Q $P = PQ^*Q = a^*bQ^*$

Where $= P = b + aa^*b$ & a is any regular expression.

$I_3 = \wedge R = R\wedge = R$

$P\wedge + PQ^*Q$ I_{12}

$P(\wedge + Q^*Q)$

PQ^* putting P's value.

$(b+aa^*b)Q^*$

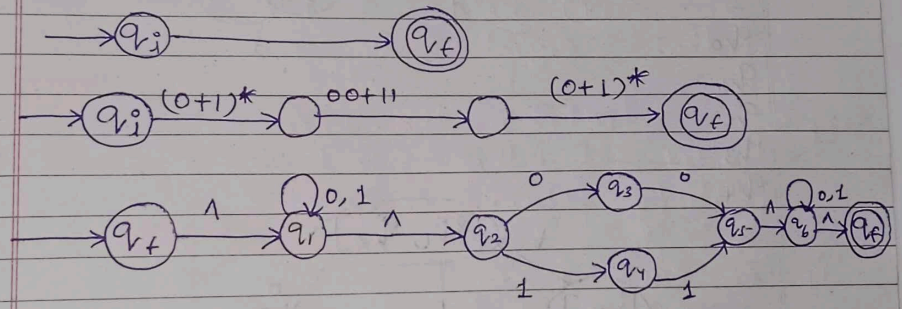
$(\wedge + aa^*bQ^*$ I_{12}

I_9 $(\wedge + aa^*)bQ^*$

a^*bQ^*

$P + PQ^*Q = \boxed{a^*bQ^*}$

Q $(0+1)^*(00+11)(0+1)^*$



Date _____ Page _____

DFA Table

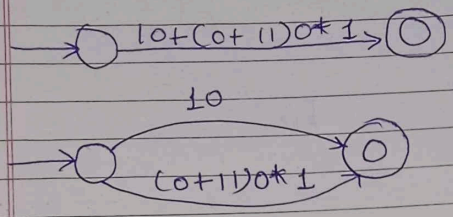
Σ	0	1
q_1	q_1q_4	q_1q_6
q_4	q_8	–
q_6	–	q_8
q_8	q_8	q_8

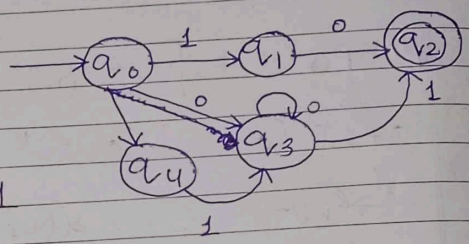
Σ	0	1
$[q_1]$	$[q_1q_4]$	$[q_1q_6]$
$[q_1q_4]$	$[q_1q_4q_8]$	$[q_1q_6]$
$[q_1q_6]$	$[q_1q_4q_8]$	$[q_1q_6q_8]$
$[q_1q_4q_8]$	$[q_1q_4q_8]$	$[q_1q_6q_8]$
$[q_1q_6q_8]$	$[q_1q_4q_8]$	$[q_1q_6q_3]$

*

Q1

$10+(0+11)0^*1$

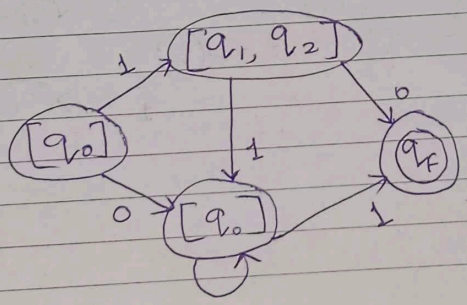




Q2

NFA Table

	0	1
q_0		
q_1		
q_2		
q_3		
q_4		

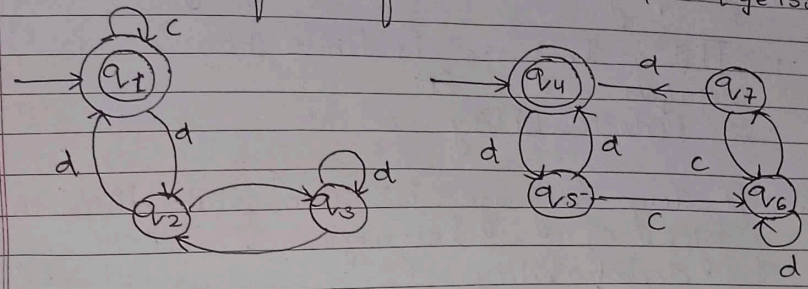


Date

Page

* Equivalence of two finite automata :- Page 158

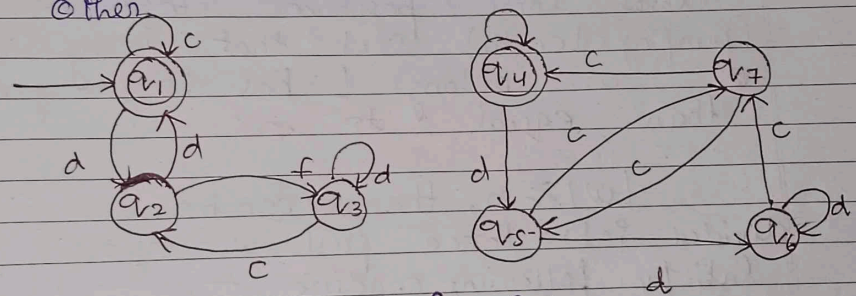
Q1



~~(q_1, q_1)~~	(q_c, q_c)	(q_1, q_d)
(q_1, q_4)	(q_1, q_4)	$(q_2, q_{5'})$
$(q_2, q_{5'})$	(q_3, q_6)	(q_1, q_4)
(q_3, q_6)	$(q_2, q_{5'})$	(q_3, q_4)

These two DFA are equivalent to each other.

Q2



(q_2, q_5)	(q_3, q_7) ,	(q_1, q_6)

They are not equivalent to each other.

Date _____ Page _____

Pumping Lemma → (Negativity test) :-

- If any language is finite than it is regular — FA
  undecidability.
- If any language is non finite than this type of language is
  decidability.

(L) → (PLT) → Pass ⇒ undecidiability.

(L) → (PLT) → fail ⇏ decidable.

- If L is an infinite language than their exists some positive integar n (pumping Length) such that any String w belongs L has length greater than equal to n.

  |w| ≥ n, than w can be divided into three parts w = xyz Satisfy following condition. —

(i) for each i ≥ 0,
  $xy^iz \in L$

(ii) |y| ≠ 0

(iii)

Date

Page

(iii) $|xy| \leq n$

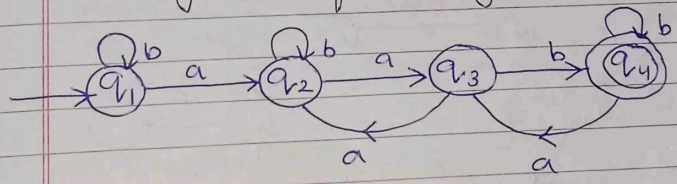
Pigeon - Hole - Problem :-

→ This principle is used in automata theory to prove that certain infinite language are not regular.

→ It states that n Pigeons fly into m^m Pigeon hole and $n \geq m$ then atleast one hole must contain two or more Pigeon.

→ In automata theory, Pigeon are the Strings of a's, the Pigeon hole are the State and the correspondance associated each String with the State and two aways a goes when the String is input.

① 10 Symbol from right end 1.



1 ____ $(0+1)(0+1)(0+1)(0+1)$

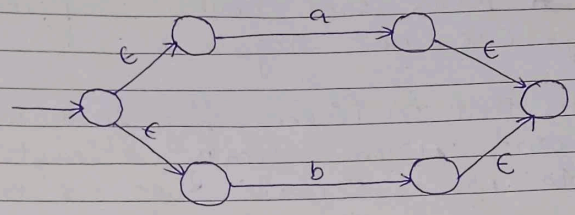
Date _____ Page _____

Q) i) $a^{2n} b^{2m+1}$

$(aa)^* (bb)^* b$

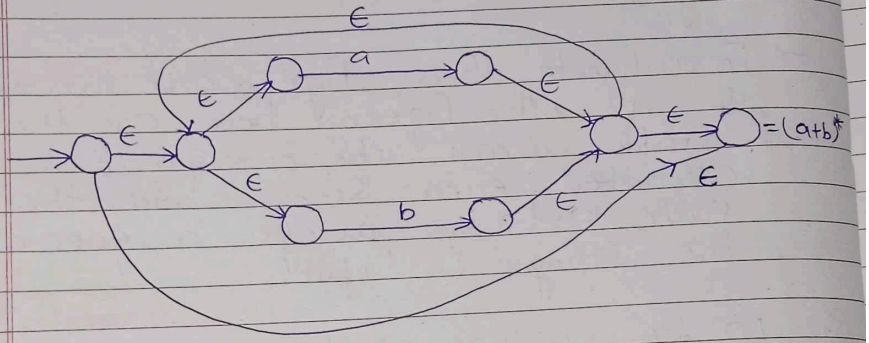
Q) $a \cdot (a+b)^* \; b \cdot b$

Step 1.

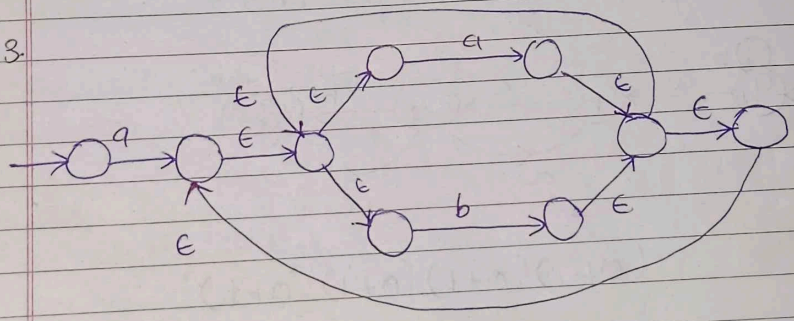


$= (a+b)$ ka auto mata

Step 2.



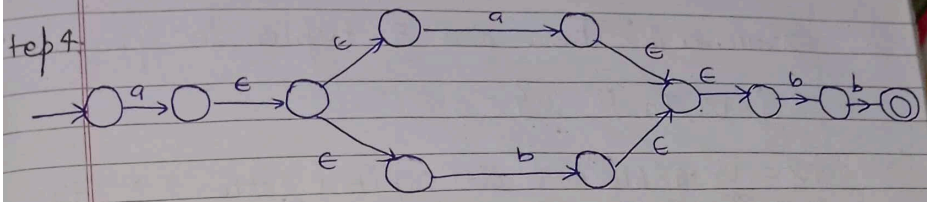
Step 3.



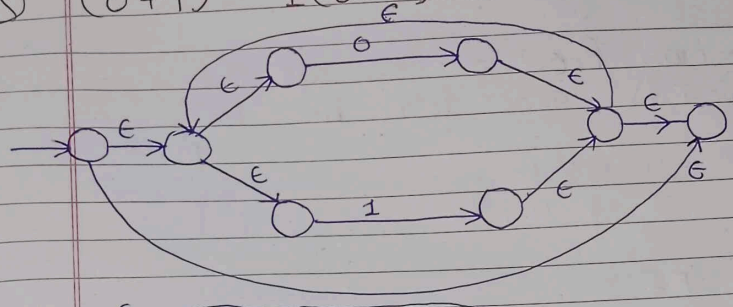
$a \cdot (a+b)^*$

Date Page

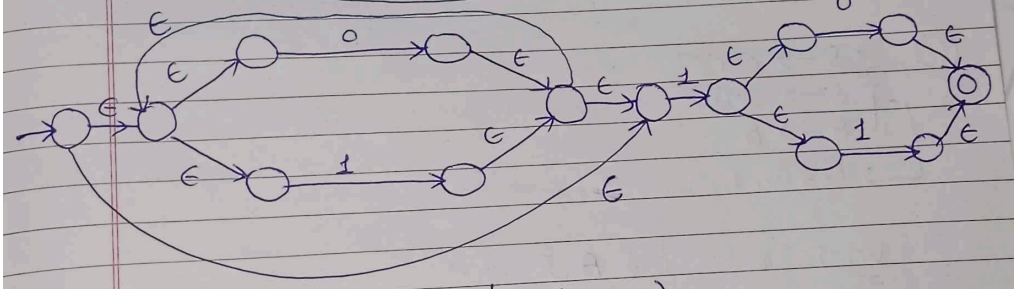
step 4



(3) $(0+1)^* 1(0+1)$



$= (0+1)^*$



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

Date Page

Grammar :- There are 4 tuples in Grammar.

{ V, T, P, S }

V = Variable → (all capital letter) → States.
T = Terminal → (all small letter) → Input Symbol
P = Production
S = Start Symbol

→ Grammar Provide you all the String which is generated by language.

Language
↓ ↓
FA Grammar.

Q
S → aSB
S → aB
B → b
→ Production.

Variable = { S B }
Terminal = { a, b }
Start Symbol = { S }

String = aabb (Left most Derivative)

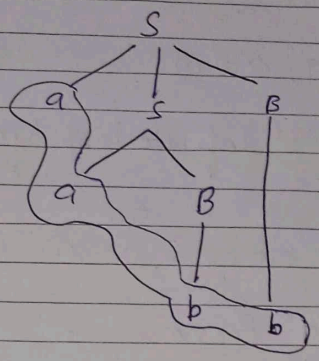
RMD (Right most derivative)

S = aSB
= aSb
= aaBb
= aabb

S = aSB
S = aaBb
= aabb.

Date _____ Page _____

Parse Tree.



aabb.

eied of tree.

Q Set of all String of length 2.

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

$RE = (a+b)(a+b)$

Prove ab

Production = $S \rightarrow AA$
$A \rightarrow a/b$

Variable = { S, A }
Terminal = { a, b }

String ⇒ a b
$S = AA$
$= aA$
⇒ ab

Date _____ Page _____

Q $a^n / n \geq 1$

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

RE = a+ (Infinite loop kye liye + symbol use karte hai without ϵ)

$S \rightarrow as/a$ either S or a

String - aaaa Proove

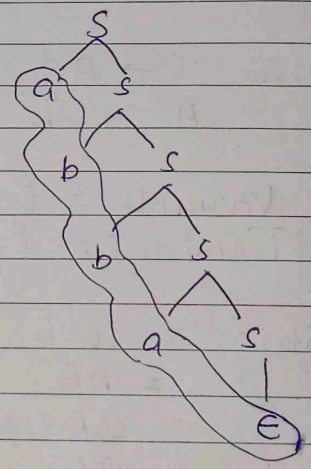
$S = as$
$= aas$
$= aaas$
$S = aaaa$

Q (a+b)*

$S \rightarrow as/bs/\epsilon$

String ⇒ abba

as
abs
abbs
abbas
abbaϵ
abba



Date Page

Q1 Set of all strings of even length.

$[(a+b)(a+b)]^*$

$B \rightarrow a/b$
$A \rightarrow BB$
$S \rightarrow AS/\epsilon$

abba.

Q2 Set of all String of odd length.

$RE = [(a+b)(a+b)]^* (a+b)$

$S \rightarrow AB$
$A \rightarrow BB$
$B \rightarrow a/b$

Q $a^n b^m / n, m \geq 1$

$S \rightarrow AB$
$A \rightarrow aA$
$B \rightarrow bB$

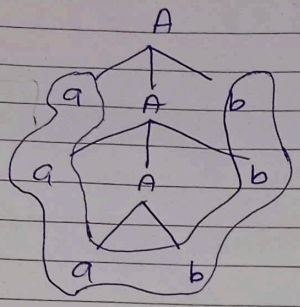
Q $a^n b^n c^m / n, m \geq 1$ this is not regular grammar.

CFG se context free grammar likh sakte.

$S \rightarrow AB$
$A \rightarrow aAb/ab$
$B \rightarrow CB/C$

Date Page

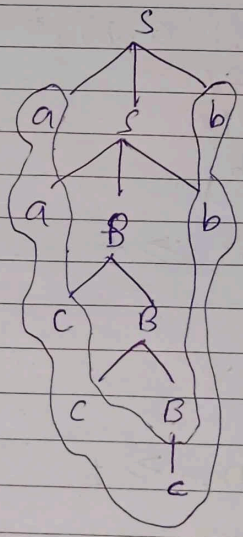
String - aaabbb



Q $a^n c^m b^n$ / $n, m \geq 1$ aaccbb

$S \rightarrow aSb / aBb$

$B \rightarrow CB / C$



Q $a^n b^n c^m d^m$ / $n, m \geq 1$

$S \rightarrow AB$

$A \rightarrow aAb / ab$

$B \rightarrow CBd / Cd$

Date _____ Page _____

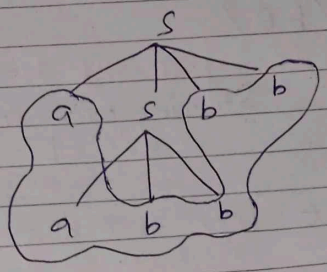
Q $A^n b^n C^n$, $n \geq 1$ not RE.

Turing Machine.

Q $a^n b^{2n}$ / $n \geq 1$ not regular
context free grammar.

$S \rightarrow aSbb / abb$

String - aabbbb



Q $a^n b^m C^m d^n$ / $n, m \geq 1$

$S \rightarrow aSd / aAd$
$A \rightarrow bAC / bC$

Q Set of all string atleast two.

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

$S \rightarrow AAB$
$B \rightarrow aB / bB / \epsilon$
$A \rightarrow a / b$

Date Page

(4) atmost two 0, 1, 2

$(a+b+\epsilon)(a+b+\epsilon)$

$S \to BB$

$B \to a/b/c$

(5) Starting and ending with different Symbol -

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

$S \to aBb / bBa$

$B \to aB / bB / \epsilon$

(6) $a(a+b)^*b$ Staring with a ending with b

$S \to aBb$

$B \to aB / bB / \epsilon$

(7) Starting and ending with same symbol.

$a(a+b)^*a + b(a+b)^*b + \epsilon$

$S \to aBa / bBb / a / b / \epsilon$

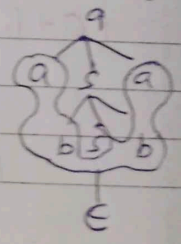
$B \to aB / bB / \epsilon$

(8) Set of pallendrom: (even pallendroom).

$L = ww^R$

$S \to asa / bSb / a / b / \epsilon$

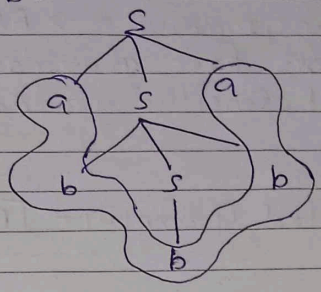
ng abba



Date _____ Page _____

① Set of odd palendrom:

$L = \omega b \omega^R$



② $a^n b^m c^n d^m \; / \; n\, m \geq 1$

③ $a^{m+n} b^m c^m \; / \; n, m \geq 1$

④ $a^n b^{n+m} c^m \; / \; n, m \geq 1$

⑤ $a^n b^m c^{n+m} \; / \; n, m \geq 1$

⑥ $a^{m+n} b^m c^m \; / \; n, m \geq 1$

$L = a^n a^m b^m c^n$

$S \rightarrow aSC \; / \; aAC$

$A \rightarrow aAb \; / \; ab$

Sm'y $\Rightarrow$ aaaabbcc

$S \rightarrow aSC$

$S \rightarrow aaACC$

$S \rightarrow aaaAbCC$

$= aaaabbcc$

⑦ $a^n b^{n+m} c^m \; / \; n, m \geq 1$

$L = a^n b^m b^m c^m$

Date

Page

Types of Grammar :-

Chemistry hierarchy

Type 3 → Regular Grammar → FA
Type 2 → Context Free grammar → PDA
Type 1 → context Sensitive grammar → Linear bounded automata.
Type 0 → Unrestricted Grammar - TM (Turing Machine)

Type 3 :- Regular grammar → FA.

* $\boxed{A \to \alpha B / \beta}$ Right linear grammar.

$B \in$ variable. $\alpha \cup \beta \in T$

* $\boxed{A \to B\alpha / \beta}$ Left linear grammar.

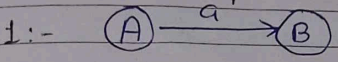
Ex: $A \to aB/a$
$B \to aB/bB/a/b$] Right linear grammar

Ex: $A \to Ba/a$
$B \to Ba/Bb/a/b.$] Left linear grammar.

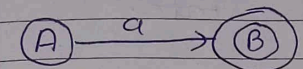
Combination of LFG and RLG is not type 3 Grammar either it is in LLG or RLG.

Date _____ Page _____

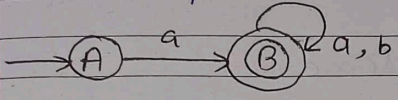
Conversion of FA to Regular grammar.

Rule 1:- 

$A \rightarrow aB$



$A \rightarrow aB/a$

Ex: 

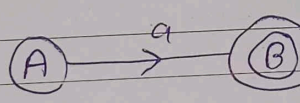
FA $\rightarrow$ RLG $\rightarrow$ LLR
(L) (L) $(L)^R$

$A \rightarrow aB/a$
$B \rightarrow aB/bB/a/b$ Right linear grammar.

$A \rightarrow Ba/a$
$B \rightarrow Ba/Bb/a/b/\epsilon$ $\Rightarrow$ Left linear Grammar

without converting Reverse.

FA $\rightarrow$ FA $\rightarrow$ RLG $\rightarrow$ LLR
(L) $(L)^R$ $(L)^R$ $((L)^R)^R$



$A \rightarrow aB/a$
$A \rightarrow aB/bB/a/b$

Date

Page

Reverse steps :-

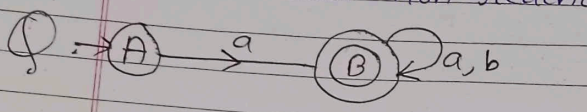
Step 1- Draw FA First.

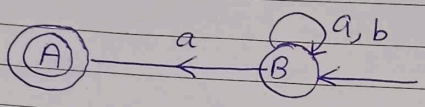
Step 2 Make Initial State as Final State and Final state as Initial State.

Step 3. Reverse the transition.

Step 4:. loops remain Same.

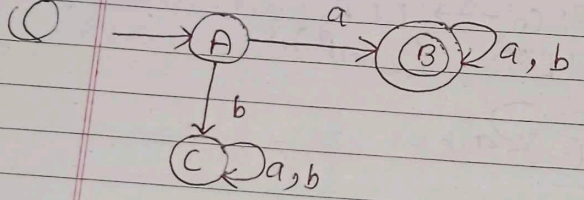
Step 5- Remove all non reachable State.





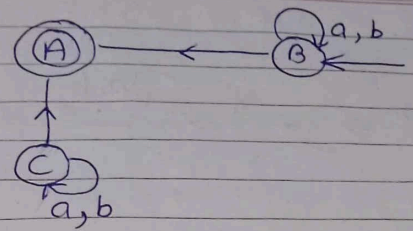
RLG ⇒ $B \rightarrow aA / aB / bB / a$
$A \rightarrow \epsilon$

LLG = $B \rightarrow Aa / Ba / Bb / a$
$A \rightarrow \epsilon$



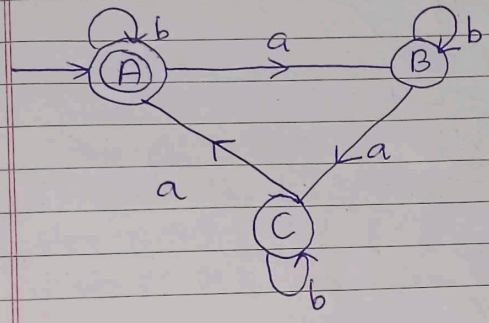
Date

Page



Converting RG to finite automata:

A → aB / bA / b
B → aC / bB
C → aA / bc / a

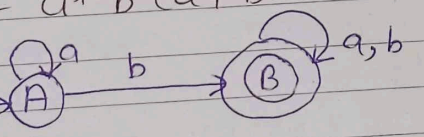


Language → No. of a's divisible by 3.

Q) Construct a regular grammar for this.

P = a* b (a+b)*

FA →

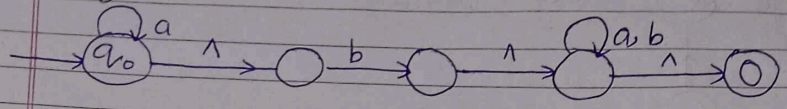


RG →

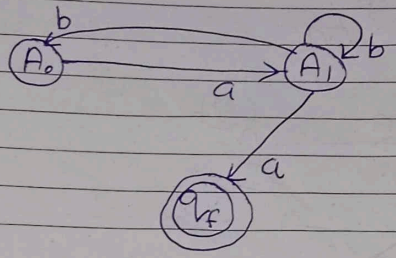
A → aA / bB / b
B → aB / bB / ∈

Date _____ Page _____

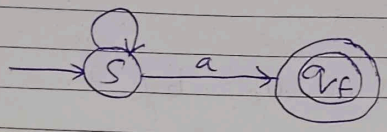
using ϵ



Q $A_0 \Rightarrow aA_1$
$A_1 \Rightarrow bA_1$
$A_1 \rightarrow a$
$A_1 \rightarrow bA_0$



Q $S \rightarrow aS/a$



Palindrom :- $S \rightarrow 0S0/1S1/\epsilon/0/1$

↳ Sentential form.

$S \rightarrow 0S0$

$S \rightarrow 0|S|0$

$S \rightarrow 0|1|0$

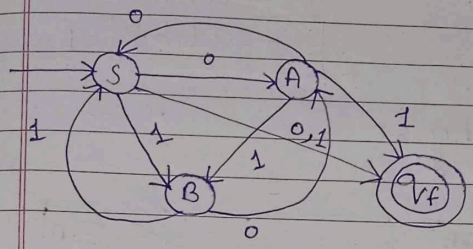
Date

Page

Convert the following RG in FA.

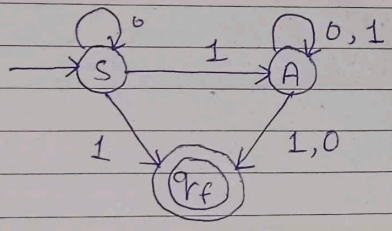
R.G ⇒ $S \to 0A / 1B / 0 / 1$
$A \to 0S / 1B / 1$
$B \to 0A / 1S$

{ $V = S, A, B$, $q_0 = S$, $\Sigma = (0, 1)$, }
$F = q_f$



Q $S \to 0S / 1A / 1$
$A \to 0A / 1A / 1 / 0$

{ $V = S, A, q_f$, $q_0 = S$, $\Sigma (0,1)$; $F = q_f$ }

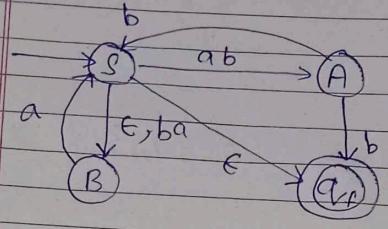


Q $S \to abA$
$S \to B$
$S \to baB$
$S \to \epsilon$
$A \to bS$

Date _____ Page _____

$B \rightarrow aS$
$A \rightarrow b$

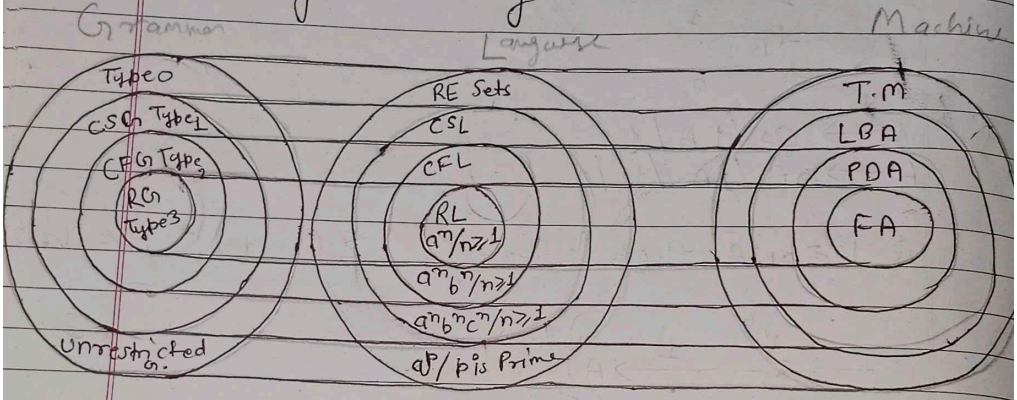
$\{ V = S, A, B, \ q_0 = S, \ \Sigma (0, 1)$
$F = q_f \}$



Prove abba

$S \rightarrow abA$
$S \rightarrow abbbS$
$S \rightarrow abbB$
$S \rightarrow abbaS$
$S \rightarrow abbaϵ$
$S \rightarrow abba$ Proved

Chomsky hierarchy :-



Type 0 Grammar :-

UG ← RE Sets → TM
generates acceptor

(Recursively Enumarable Language.)
REG / Phase Structure grammar.

Date

Page

$\alpha \to \beta \quad , \alpha \in (\Sigma \cup V_n)$

$\beta \in$

Type 1: Context Sensitive Grammar:

$\alpha \to \beta$

$\alpha \in (\Sigma \cup V_m)^* V_n (\Sigma \cup V_n)$

$\beta \in (\Sigma \cup V_n)^+$

length Increasing grammar or non-contracting grammar.

Rule 1: only Start State $\in$
$S \to \in$ is allowed to exception.

Rule 2: if we use Start ($S \to \in$) then we will never use S on the right side.

$\alpha A B \to \alpha S \beta$

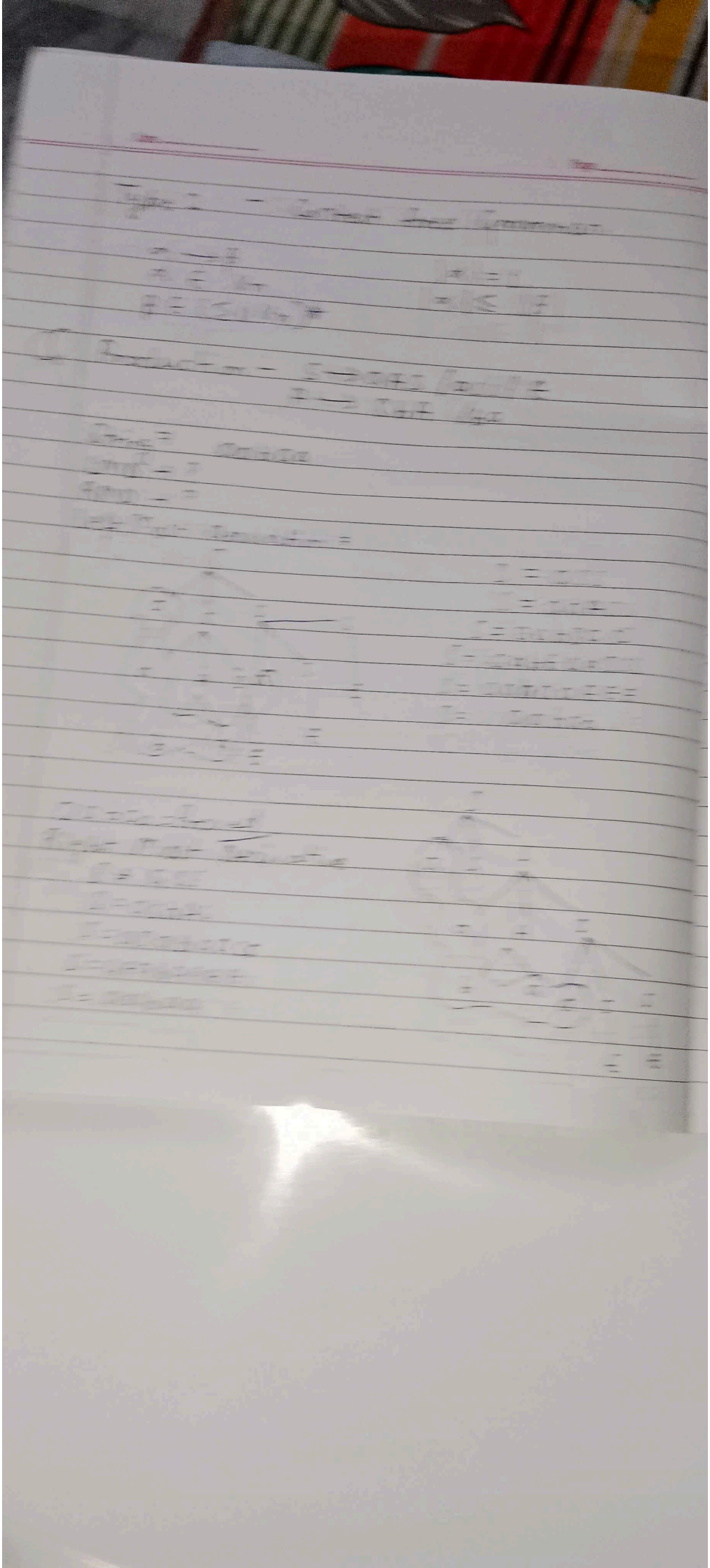
$\alpha, \beta \in (\Sigma \cup V_n)^*$

$A \in V_n$

$S \in (\Sigma \cup V_n)^*$

i) $S \to \in$
is allowed as exception

ii) $S \to aS$
X

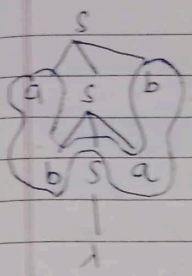


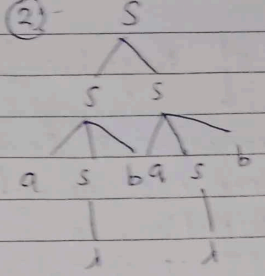
Date _____ Page _____

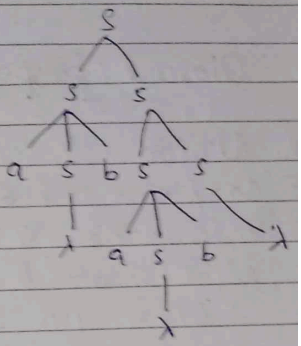
Q $G = (\{S\}, \{a, b\}, S, S \to asb / bsa / ss / \lambda$
is this Grammar ambiguous or not.

String ⇒ abab

NOTE ⇒ for a grammar we make so many Parse tree then this grammar is Ambiguous.

i) 

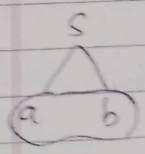
② 

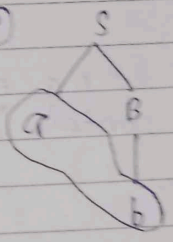
③ 

So many Parse tree so this is called Ambiguous grammar.

Q $G = (\{S\}, \{a, b\}, S,$ $S \to aB / ab$
$A \to aAB / a$
$B \to Abb / b$

String ⇒ ab.

i) 

② 

this is ambiguous grammar.

Q1 $L = \{a^m b^m, m \geq 0, m \geq 3\}$

RLG, LLG

$L = \{aaabbb\}$

RLG

LLG

Closure property of Regular Language:

1. Union 2. concatenation 3. Reversal 4. Difference
5. complement 6. Kleen Closure 7. homomorphism
8. Intersection 9. Substitution 10. Inverse homomorphism
11. right Quotient, 12. EDIT operation. 13. Infinite Union 14. Subset / Subset

Date _____ Page _____

Ex 1.

1. Union :- $L_1 \cup L_2$

$RE_1 + RE_2 = RE$

$RE_1 = a(aa)^*$ (odd no of 'a' excluding null)

$L_1 = \{a, aaa, aaaaa ---\}$

$RE_2 = (aa)^*$ (even no. of a's including null)

$L_2 = \{\epsilon, aa, aaaa ---\}$

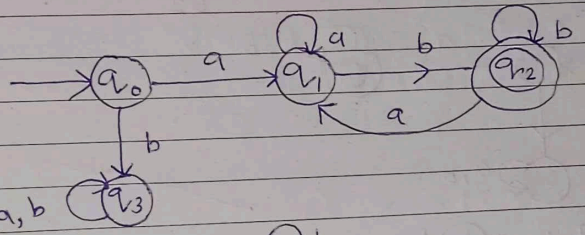
$L_1 \cup L_2 = \{\epsilon, a, aa, aaa ---\} = RE = a^*$.

Ex 2. Starts with a and ends with b.

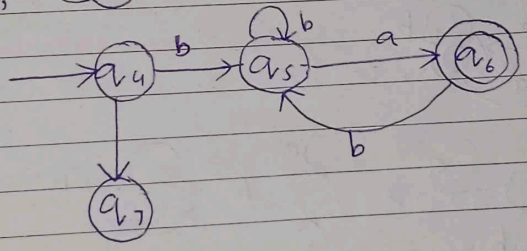
$L_1 = \{a, ab, abb ---\}$

$L_2 = \{b, ba, baa ---\}$

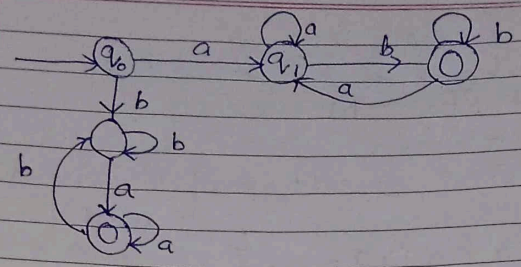
$L_1 \Rightarrow$



$L_2 \Rightarrow$



Date Page

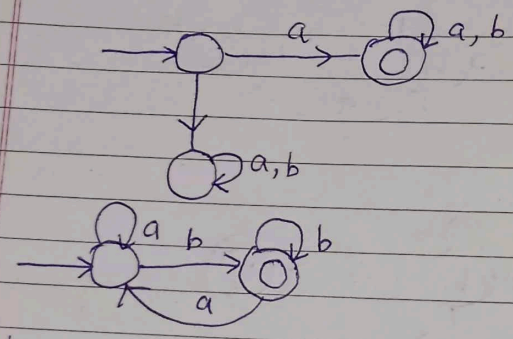


2. Concatenation :- $L_1 \cdot L_2$

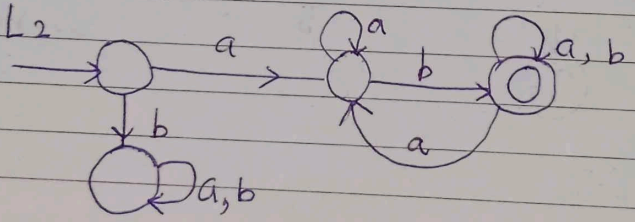
$RE_1 \cdot RE_2 = RE$
$RE_1 = (0+1)^*0$
$RE_2 = 0(0+1)^*$
$L_1 = \{0, 01, 001, 010 \text{----}\}$
$L_2 = \{01, 001, 010 \text{---}\}$
$L_1 \cdot L_2 = \{001, 0010 \cdot$

Ex 2:- Starts with a & End with b

L_2 L_2



$L_1 \cdot L_2$



Date _____ Page _____

$L = \{a, ab, aabb\}$

$INIT(L) = \{\epsilon, a, \epsilon, a, b, \epsilon, a, aa, aab, aabb\}$

Homomorphism :- For every symbol in I/P if you substitute a string from

Ex: $h: \Sigma \rightarrow \Gamma^*$

$\Sigma = \{a, b\}$ $\Gamma = \{0, 1, 2\}$

$h(a) = 01$

$h(b) = 112$

for $h(aba) = 0111201$

or for language $= L = a^*b$

$(01)^* 112$

Inverse Homomorphism :- The Inverse homomorphis -m image of any language.

$h^{-1}[L] = \{x / h(x) \text{ is in } L\}$

for a string ω, $h^{-1}(\omega) = \{x / h(x) \in L\}$

Date _____ Page _____

3. Complement property :-

$\overline{L_1} = \Sigma^* - L_1$

$RE_1 = (aa)^*$ Even no of a's including ϵ

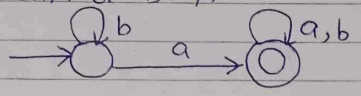
(L1) $RE_2 = a(aa)^*$ $\boxed{\Sigma^* = L}$

$L = \{\epsilon, a, aa, aaa \cdots\}$

$L_1 = \{\epsilon, aa, aaaa \cdots\}$

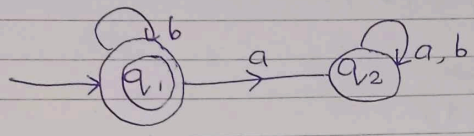
$\Sigma^* - L_1 = \{a, aaa, \cdots\}$

Contain a DFA



Initial State
final State
ban jati hai.

Ka L' = doesn't containing a



Reversal -

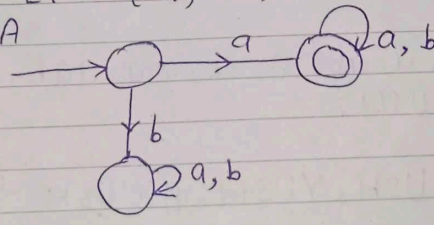
$RE(L) = 01 + 10 + 11 + 10$

$RE(L^R) = 10 + 01 + 11 + 01$

L_1 = Starts with a

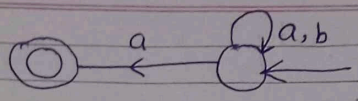
$L_1 = \{a, ab, aab \cdots\}$

DFA



Reverse karo
initial State final State
final State initial State
loop same
dead State delete and direction opposite

Date Page



Difference :- $RE_1 = a(a)^*$ a+ 2 ya jisko sakte hai.

$RE_2 = (aa)^*$

$L_1 = \{a, aa, aaa \text{---} \}$

$L_2 = \{\epsilon, aa, aaaa \text{---} \}$

Difference = $\{a, aaa, aaaaaa \text{---} \}$

$RE = a(aa)^*$

* Substitution :- Every symbol of a language is substituted by the other lang.

given $L(a+b^*) = ?$

$\Sigma = (a, b)$

$f(a) = 0^*$

$f(b) = 01^*$

$L = a + b^* = 0^* + (01^*)^*$

Substitution is closed under RE.

INIT Operation :- Given a lang.

Prefix $\{a, ab, aabb\}$

$= \{\epsilon, a, \epsilon, a, ab, \epsilon, a, aa, aab, aabb\}$

Suffix = $\{AYUSH\}$

$= \{\epsilon, H, SH, USH, YUSH, AYUSH\}$

Date

Page

$L = \{a, ab, aabb\}$

INIT(L) = {ε, a, ε, a, b, ε, a, aa, aab, aabb}

Homomorphism :- For every symbol in I/P if you substitute a string from

Ex: $h: \Sigma \rightarrow \Gamma^*$

$\Sigma = \{a, b\}$ $\Gamma = \{0, 1, 2\}$

h(a) = 01

h(b) = 112

for h(aba) = 0111201

or for language = L = a* b

(01)* 112

Inverse Homomorphism :- The Inverse homomorphis - m image of any language.

$h^{-1}[L] = \{x / h(x) \text{ is in } L\}$

for a string w, $h^{-1}(w) = \{x / h(x) \in L\}$