2020

1.a Define alphabet and string. Explain the operations of string.

Solution

An alphabet is a finite set of symbols denoted as Σ. For example $\Sigma = \{0, 1\}$

A string is a finite sequence of symbols over an alphabet Σ.

operations of string:

1. Concatenation: The concatenation of two strings u and v is ~~denoted~~ denoted by $u.v$, the string is formed by the string u followed by the string v. For example $u = man$, $v = city$ the $u.v = mancity$

2. Length: The length of a string u is the number of symbols in u, denoted by $|u|$ For example, $u = Rodrigo$ the $|u| = 7$

3. Reversal operation: Reversal of a string u (denoted by u^R) is a string obtained by reversing the order of symbols in u. for example $u = erdogan$ $u^R = nagodre$

substring: If w is a string then v is substring of w if there exists string x and y such that $w = xvy$. x is called prefix, y is called ~~suffix~~ suffix. For example: W = bitwise xor V = [redacted]

some additional definitions:

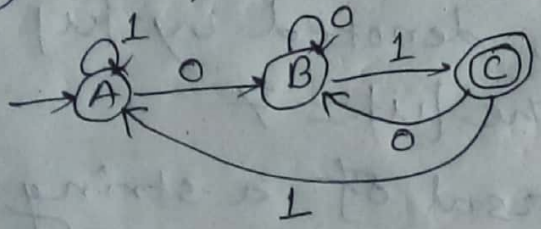
Symbol: Symbols are the basic building blocks of Theory of computation.

Language: Any set of strings (empty or not empty, finite or infinite) is called a language or collection of strings over alphabet (Σ) under certain condition.

Example: $L = \{$ All strings of length 2 over $\Sigma = \{a, b\}\}$

~~2.a~~ 1.b Construct DFAs for the following regular languages

i) $L = \{w \in \{0,1\}^* \mid w$ ending with '01'$\}$



ii) $\{w \in \{0,1\}^* \mid w$ is a binary number whose decimal equivalent is divisible by 3$\}$

Let w be any binary number

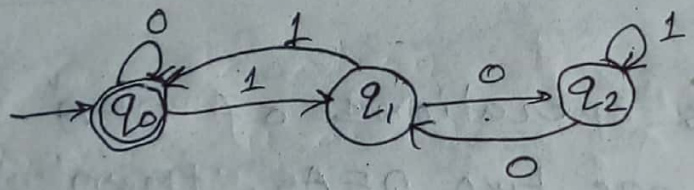
then w mod 3 = 0, 1, 2. So the no. of states 3

Let they are q_0, q_1, q_2

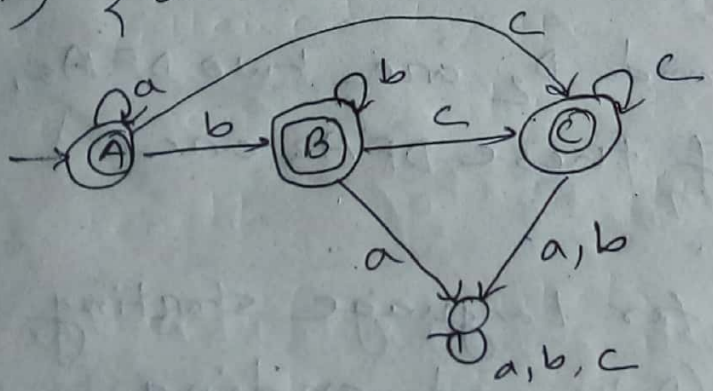
Let's form the state transition table

	0	1
→*q_0	q_0	q_1
q_1	q_2	q_0
q_2	q_1	q_2

Here q_0 is the starting and final or accepting state because we need divisible by 3.



iii) $\{a^m b^n c^l : m, n, l \geq 0\}$



2.a Define Finite Automata (FA)

A Finite Automata (FA) is a 5-tuple

$M = (Q, \Sigma, \delta, q, F)$, where

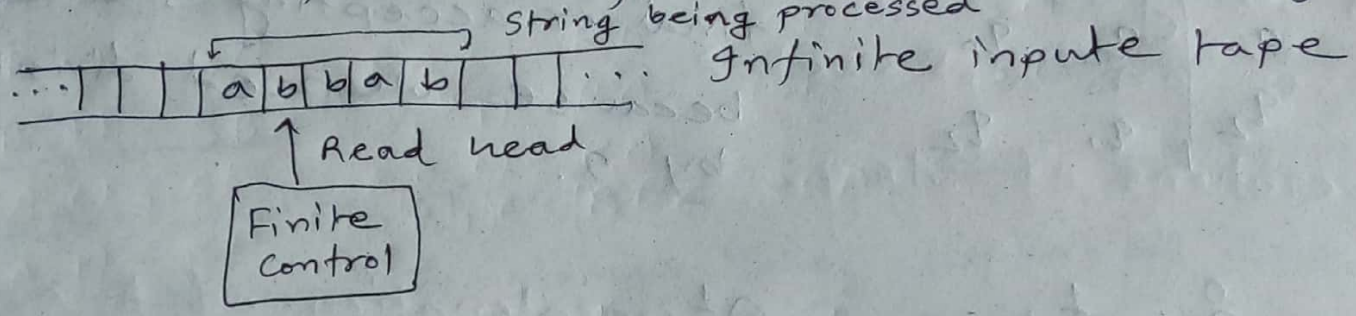
1. Q is a finite set of states
2. Σ is a finite set of symbols known as alphabet

3. $\delta : Q \times \Sigma \to Q$ is a function, called the transition function.

4. q is an element of Q; it is called the starting state. $q \in Q$

5. F is a subset of Q; the elements of F are called accept states. $(F \subseteq Q)$

Show the features of FA with block diagram



2.b Discuss the various operations of DFA

1. Union: If D_1 and D_2 are two DFAs, then union of these DFAs are defined as $D = D_1 \cup D_2$

2. Concatenation: If D_1 and D_2 are two DFAs, then intersection of these DFAs are denoted by $D = D_1 \cdot D_2$.

For example: DFA for Language starting with a (D_1) ~~and ending~~

DFA for language ending with b (D_2)

Then $D = D_1 \cdot D_2$ is the language starting with a and ending with b

the union is defined as

D_1 = Language starting with a and ending with b

D_2 = " " " b " " a

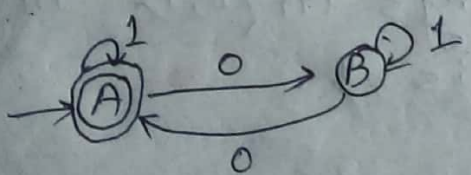
then $D = D_1 \cup D_2$ is the DFA for Language starting and ending with different symbol.

Complement of DFA: If DFA accepts language L, then $\bar{L}$ is accepted by $\overline{DFA}$, a version of DFA where the accepting and non-accepting states are swapped.

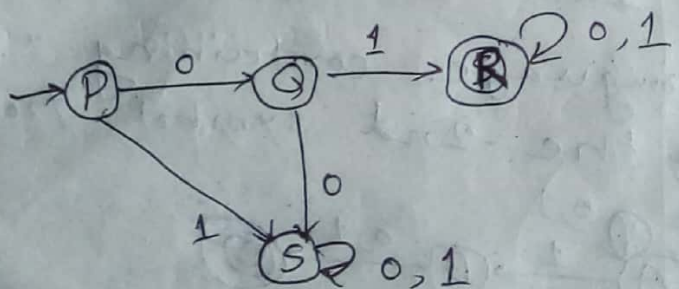
Difference of DFAs. iff D_1 and D_2 are two DFAs then $D_1 - D_2$ is defined as $D_1 \cap \overline{D_2}$

Minimization of DFA. It means reducing the number of states from give DFA.

2.c Construct a deterministic machine that accepts the string (say w) of Language over $\{0,1\}$ in which to no. of 0's of w is even and w is started with 01.



D_1



D_2

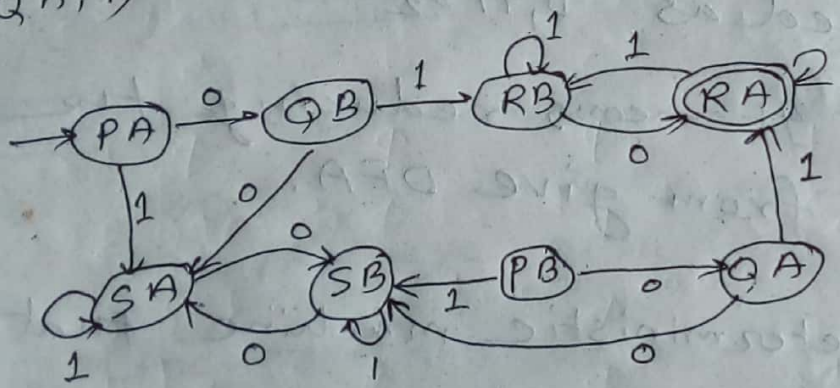
$D_1 = \{\{A, B\}, \{0,1\}, A, \delta_{D_1}, \{A\}\}$

$D_2 = \{\{P, Q, R, S\}, \{0,1\}, P, \delta_{D_2}, \{R\}\}$

Now $D = D_1 \cdot D_2$

$D = \{\{PA, PB, QA, QB, RA, RB, SA, SB\}, \{0,1\}, AP, \delta_D, \{RA\}\}$

$(PA, 0) = QB$ $(QB, 0) = SA$ $(SA, 0) = SB$
$(PA, 1) = SA$ $(QB, 1) = RB$ $(SA, 1) = SA$
$(PB, 0) = QA$ $(RA, 0) = RB$ $(SB, 0) = SA$
$(PB, 1) = SB$ $(RA, 1) = RA$ $(SB, 1) = SB$
$(QA, 0) = SB$ $(RB, 0) = RA$
$(QA, 1) = RA$ $(RB, 1) = RB$

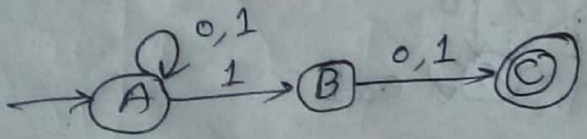


Note: PB and QA are unreachable states

The formula is

$A \cap B = \{Q_A \times Q_B, \Sigma, \delta, q_A \times q_B, F_A \times F_B\}$

3. d) ~~Ex~~ ~~Construct~~ Convert an NFA to DFA for the language containing all strings in $\{0,1\}^*$ in which the 2nd symbol from RHS is 1.



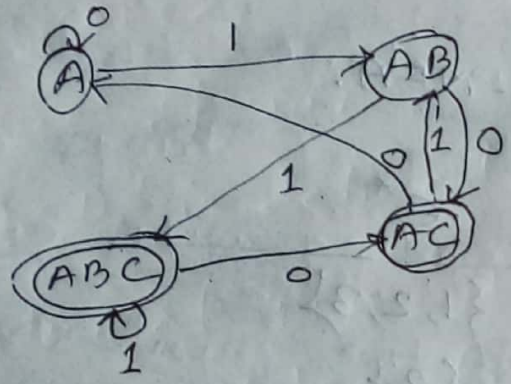
The transition table for NFA is

	0	1
→ A	{A}	{A, B}
B	{C}	{C}
*C	φ	φ

state transition table for DFA

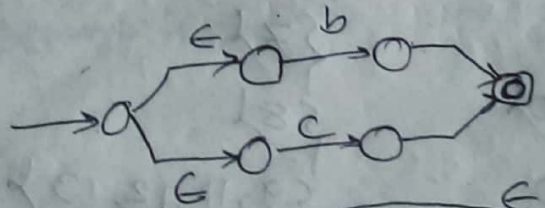
	0	1
→ A	A	AB
AB	AC	ABC
*ABC	AC	ABC
~~AB~~		AB
*AC	A	

Draw the DFA

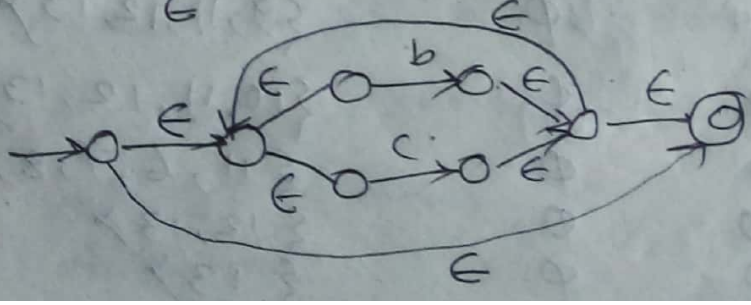


3.e Given a Regular expression $a(b|c)^*|a$. convert it to an ϵ-NFA using Thomsom construction. Then convert the ϵ-NFA directly to DFA.

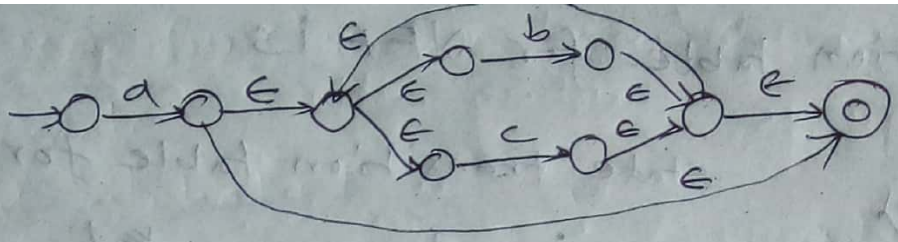
(i) b|c



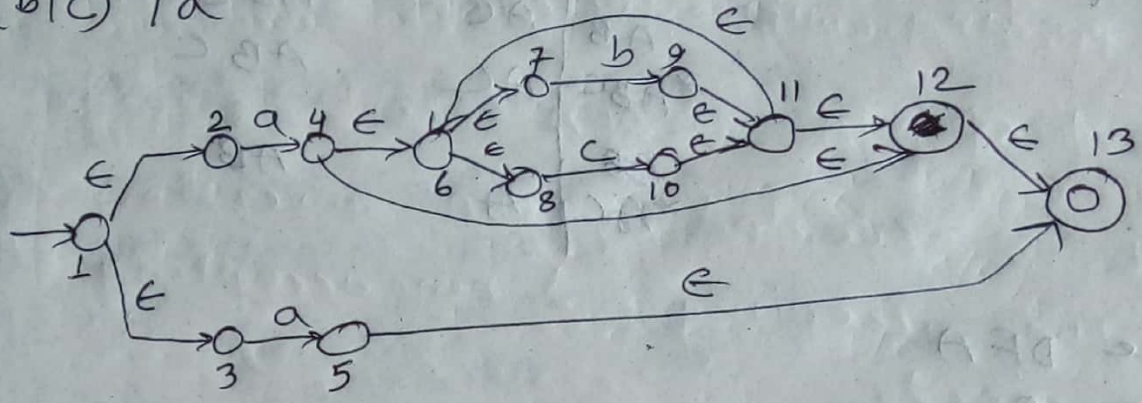
(ii) $(b|c)^*$



(iii) $a(b|c)^*$



(iv) $a(b|c)^*|a$



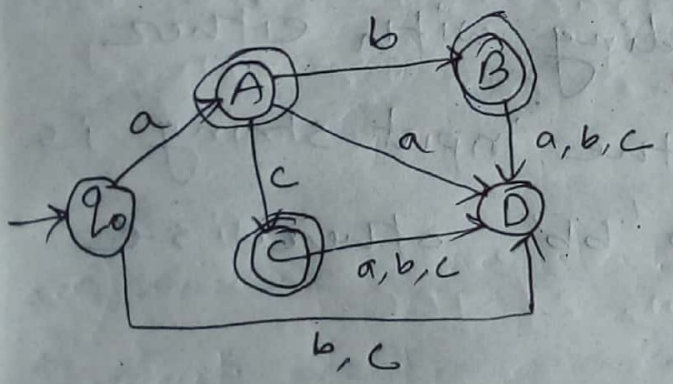
state transition table for ∈-NFA.

	a	b	c	∈*
→ 1	φ	φ	φ	{1, 2, 3}
2	{4}	φ	φ	{2}
3	{5}	φ	φ	{3}
4	φ	φ	φ	{4, 6, 7, 8, 12, 13}
5	φ	φ	φ	{5, 13}
6	φ	φ	φ	{6, 7, 8}
7	φ	{9}	φ	{7}
8	φ	φ	{10}	{8}
9	φ	φ	φ	{9, 11, 12, 13}
10	φ	φ	φ	{10, 11, 12, 13}
11	φ	φ	φ	{11, 6, 7, 8, 12, 13}
12	φ	φ	φ	{12, 13}
*13	φ	φ	φ	{13}

state transition table for DFA

	a	b	c
→ 123	4,5,6,7,8,12,13	ϕ	ϕ
*4567812,13	ϕ	9,11,12,13	10,11,12,13
*9,11,12,13	ϕ	ϕ	ϕ
*10,11,12,13	ϕ	ϕ	ϕ

Let, 4567812,13 = A, 9,11,12,13 = B, 10,11,12,13 = C and ϕ denotes the Dead end (D); q_0 = 123



4a Define Mealy Machine by 6-tuple. Give an example with state table and state Diagram

A mealy machine is a 6-tuple $M(Q, \Sigma, \delta, q_0, \Delta, \lambda)$ consisting of the following

Q = finite set of states

Σ = input alphabet

δ = Transition function $Q \times \Sigma \to Q$

q_0 = initial state

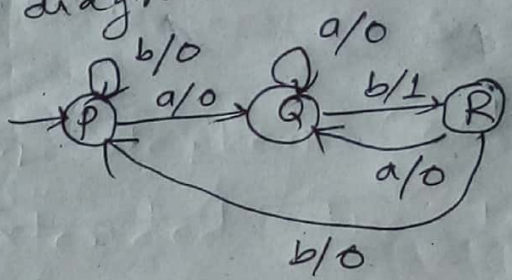
Δ = output alphabet

λ = output function ($\lambda: Q \times \Sigma \to \Delta$)

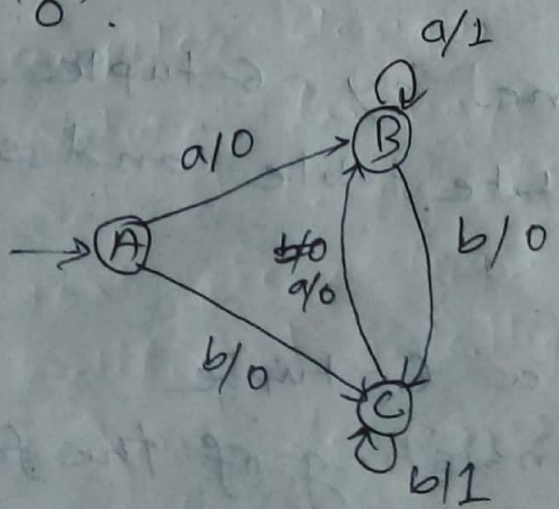
state transition table of Mealy Machine

	a	b
P	Q,0	P,0
Q	Q,0	R,1
R	Q,0	P,0

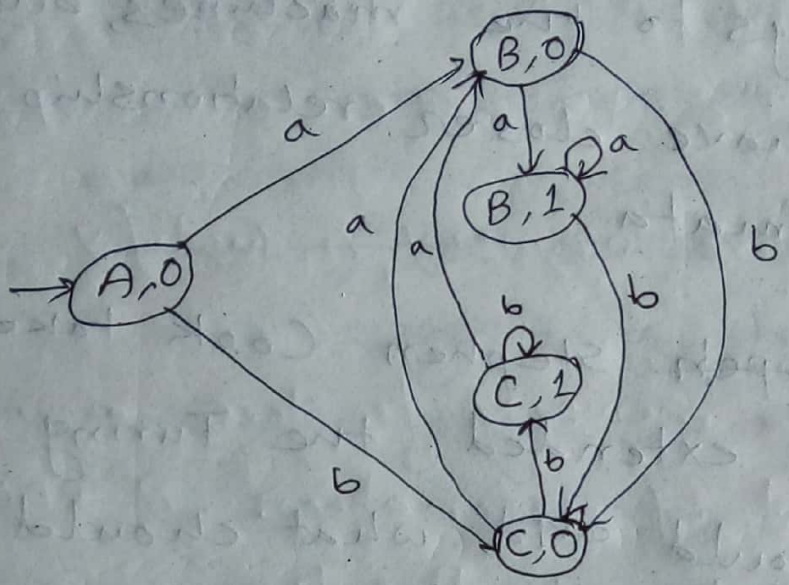
diagram



4b construct a Mealy machine that accepts the language ~~eos~~ consisting of strings over {a,b} and string should be ending with either 'aa' or 'bb'. Print '1' if the input string is ending with either 'aa' or 'bb', otherwise print '0'.



4c Convert the above Mealy Machine to Moore Machine.



5a What is Computation? Briefly discuss the evolution of TOC.

Computation: A computation is any type of arithmetic or non-arithmetic calculation that is well defined.

Before 1930's: Alan Turing studied an abstract machine that had all the capabilites of today's computers to solve problems.

1931's to 1950's: Simpler kinds of Machines were used, which we called Finite Automata

Late 1950's to 1960's: Noam Chomsky began the study of formal 'grammar' that are not strictly belongs to the machines, but these grammars have closer relationships to abstract automata.

After 1960's: ~~stepeh~~ Stephen Cook takes the charge and extended the Turing's study of what could and what chould not be computed. Finally in 1971 S. Cook succeed to separate intractable problems from those problems that can be solved efficiently by computer.

5b Write down applications of TOC

1. compiler design
2. Natural language processing
3. Natural language understanding
4. Text processing and searching applications
5. Artificial intelligence
6. circuit design

5c Why FA can't recognize other than regular Languages? Explain with example.

two properties of Finite Automata.

i) the amount of memory is finite and independent of length of string ~~(i.e. is finite number of states)~~

ii) it can recogniz a string as a part of Regular Language, is the string contains a non-empty substring such that if it is replaced with an arbitrary number of repetitions of itself, the result must again be in the R.Language.

Language like $\{a^n b^n \mid n \geq 0\}$ can't be recognized by FA. because it need infinite amount of memory. Also is we take a string $s = aaabbb$ and substring of it $y = ab$. By replacing this part with two copies of y yields $aaababbb$ does not belong to $\{a^n b^n \mid n \geq 0\}$. That is FA can't recognize other than RL.

6a Construct a CFG to generate even and odd set of palindromes over $\Sigma = \{a, b\}$.

$S \rightarrow aSa \mid bSb \mid aa \mid bb \mid a \mid b \quad |w| \geq 1$

$S \rightarrow aSa \mid bSb \mid a \mid b \mid \in ; |w| \geq 0$

6b Does a PDA have memory? Justify

Yes.
PDA uses a stack memory to store an unbounded amount of information. A PDA can push an element onto the top of the stack and pop off an element from the stack.

6c Write and explain the algorithm for minimization of a DFA.

Algorithm:

~~Step-1: Divide~~

Suppose there is a DFA $D(Q, \Sigma, q_0, \delta, F)$ which recognizes a language L. Then the minimized DFA $D'(Q', \Sigma, q_0, \delta, F')$ can be constructed for Language L as:

Step-1: Divide Q into two sets. One set will contain all final states and other set will contain non-final states. This partition is

called P_0 ('0' equivalent)

Step-2: Initialize $K=1$

Step-3: Find P_K by partitionting the different sets of P_{K-1}. In each set of P_{K-1}, take all possible pair of states. If two states of a set are distinguishable, split the sets into two different sets in P_K.

Step-4: Stop when $P_K = P_{K-1}$

Step-5: All states of one set are merged into one. Number of states in minimized DFA will be equal to no. of sets in P_K.

Note: Two states (q_i, q_j) are distinguishable in partition P_K if for any input symbol a, $\delta(q_i, a)$ and $\delta(q_j, a)$ are in different sets in partion P_{K-1}

7a What is an instantaneous description of a PDA? How will you represent it

The instantaneous description is called as an informal notation and explains how a Pushdown automata computes the given string and makes a deision that the given string is accepted or rejected.

it is represented by 3-tuples (q, w, α)

1. q is the current state
2. w is the remaining input
3. α is the stack contents.

7b Find PDA that accept the give CFG:

i) $S \rightarrow XaaX$ ii) $X \rightarrow aX \mid bX \mid \epsilon$

① $\delta(q, \epsilon, S) = (q, XaaX)$

② $\delta(q, \epsilon, X) = (q, aX), (q, bX), (q, \epsilon)$

③ $\delta(q, a, a) = (q, \epsilon)$

④ $\delta(q, b, b) = (q, \epsilon)$

Let's accept bbaa

$\delta(q, bbaa, S)$ initial

$\delta(q, bbaa, \underline{X}aaX)$ ①

$\delta(q, bbaa, b\underline{X}aaX)$ ②

~~$\delta(q, bbaa, baaX)$ ②~~

~~$\delta(q, baa, aaX$~~

$\delta(q, bbaa, bb\underline{X}aaX)$ ②

$\delta(q, bbaa, bbaaX)$ ③

$\delta(q, baa, baaX)$ ④

$\delta(q, aa, aaX)$ ⑨

$\delta(q, a, aX)$ ③

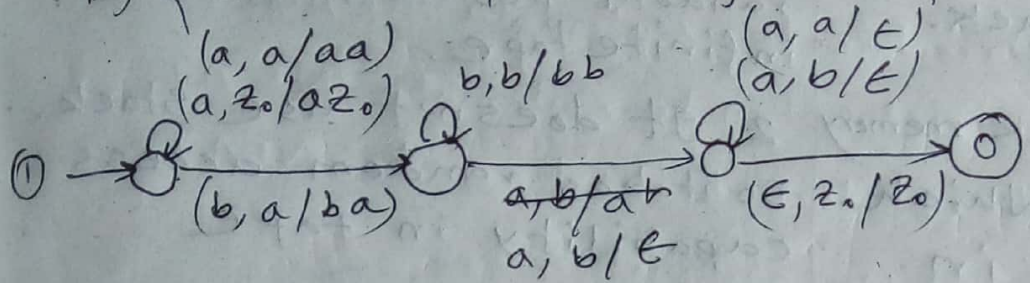
$\delta(q, \epsilon, X)$ ③

$\delta(q, \underline{\epsilon}, \epsilon)$ ②

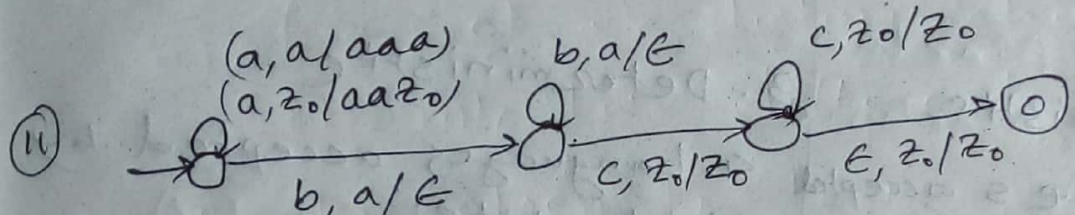
As the stack is empty string is accepted

7c Construct PDAs for the languages:

i) $\{a^n b^m a^{n+m} \mid m, n \geq 1\}$



ii) $\{a^n b^{2n} c^m \mid m, n \geq 1\}$



8a Define Turing Machine (TM). Differentiate between TM and PDA

TM: A TM is a 7-tuple

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

Q: Non empty finite set of states
Σ: input alphabet
δ: Transition function
Γ: ~~Tape~~ Tape alphabet (contains blank symbol)
q_0: Initial state
B: special symbol called blank
F: set of final states.

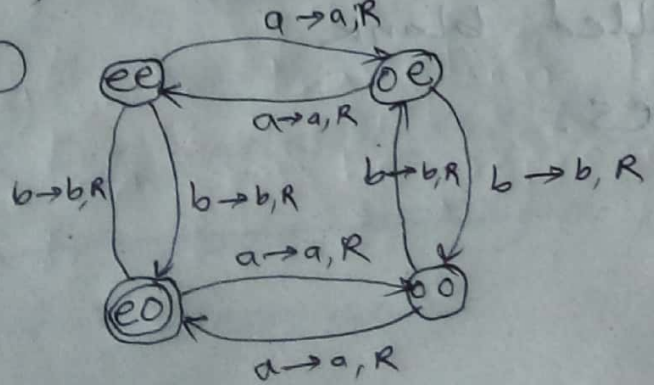
$\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$

PDA	TM
1. A PDA can access only the top of it's stack.	1. A TM can access any random position in the infinite tape.
2. It uses a stack memory for computation. Which limits the access in memory	2. It doesn't use stack. so it has random access capability in tape
3. Non-deterministic	3. Deterministic
4. The Languages accepted by either empty stack or final state	4. Languages accepted by final state.
5. Contains a read only head	5. Have read/write head
6. The head can move in only forward direction	6. The head is bidirectional (forward/backward)

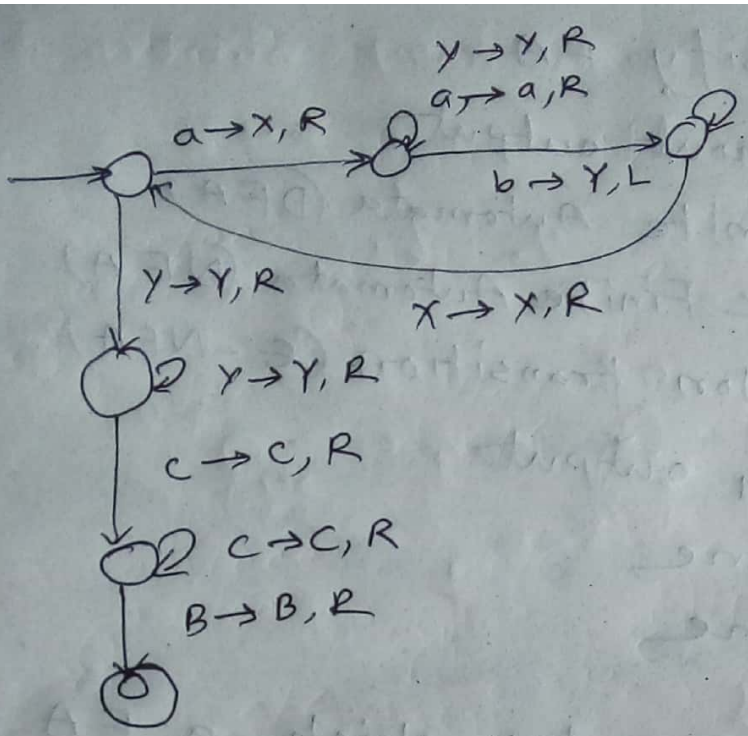
8b Construct a TM accepting the following language i) $\{w \in \{a,b\}^* \mid w$ is consist of even number of a's and odd number of b's$\}$

ii) $a^n b^n c^m \mid m, n \geq 1\}$

①



(11)



Here X and Y are written in place of a and b in input tape and B is the Blank symbol

2019

1a Discuss the basic operations of languages.

1. Union: Let L_1 and L_2 be the two languages then
$L_1 \cup L_2 = \{x, y \mid x \in L_1 \text{ and } y \in L_2\}$

2. Concatenation: $L_1 \cdot L_2 = \{x \cdot y \mid x \in L_1 \text{ and } y \in L_2\}$

3. Closure of a language: Let L be the language
$$L^* = \bigcup_{i=0}^{\infty} L^i$$

4. Intersection: Intersection is defined as $L_1 \cap L_2$. that is the language which satisfies both conditions of L_1 and L_2. (string belongs to both L_1 and L_2)

1b Define and classify FA

Finite automata without output

i) Deterministic Finite Automata (DFA)

ii) Non-deterministic Finite Automata (NFA)

iii) NFA with epsilon transition (ε-NFA)

Finite Automata with output

i) Moore Machine

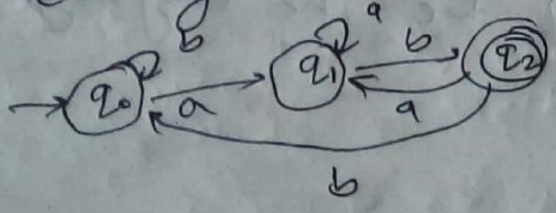
ii) Mealy Machine

1c What are the ways by which a FA can be represented? Explain with example.

FA can be represented in 3 ways.

1) State transition diagram: A state transition Diagram for FA is a diagram that is used to illustrate the operation of a FA.

For example: state transition diagrams for ending with 'ab' is



2) State transition table: The transitions of a FA can be represented as tabular form.

same example

	a	b
$\rightarrow q_0$	q_1	q_0
q_1	q_1	q_2
$*q_2$	q_1	q_0

3) Using formal Definition: The formal definition of FA is $FA = \{Q, \Sigma, q_0, \delta, F\}$

For example (ending with ab)

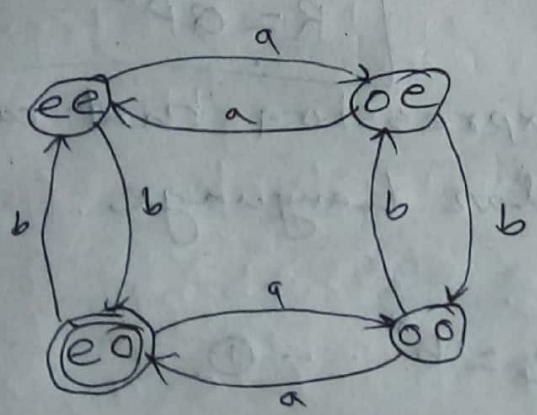
$FA = \{\{q_0, q_1, q_2\}, \{a, b\}, q_0, \delta, \{q_2\}\}$

$\delta(q_0, a) = q_1$ $\quad \delta(q_1, a) = q_1$ $\quad \delta(q_2, a) = q_1$

$\delta(q_0, b) = q_0$ $\quad \delta(q_1, b) = q_2$ $\quad \delta(q_2, b) = q_0$

2a) minimal DFAs for the following Languages

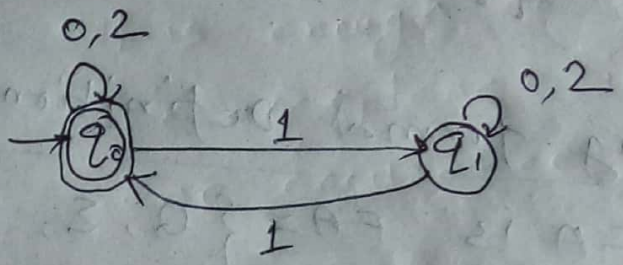
1) even number of a's and odd number of b's



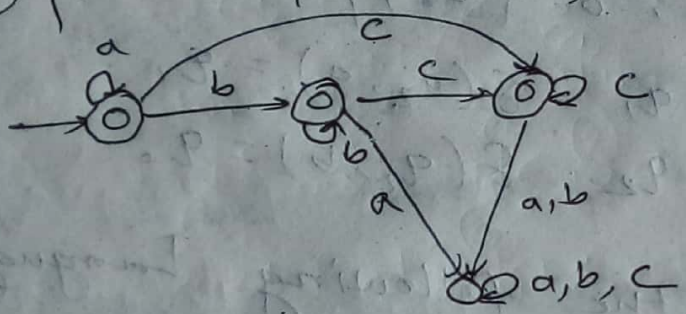
ii) $\{w \in \{0,1,2\}^* \mid w$ is a ternary number whose decimal equivalent is divisible by 2

remainders: 0, 1 ∴ states: q_0, q_1

	0	1	2
→*q_0	q_0	q_1	q_0
q_1	q_1	q_0	q_1



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



2b Define RE. Derive a RE from the give FA

$R = Q + RP$

$R = QP^*$

RE: RE is a mathematical expression that can be used to represent regular Language.

Using Arden's Method:

$q_0 = \epsilon + q_0 b$ or $q_0 = \epsilon b^*$ or $q_0 = b^*$ ... ①

$q_1 = q_0 a + q_1 b + q_2 a$ ... ⑪

$q_2 = q_1 b + q_2 a$ or $q_2 = q_1 b a^*$ ... ⑪①

from (i) (ii) and (iii)

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

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

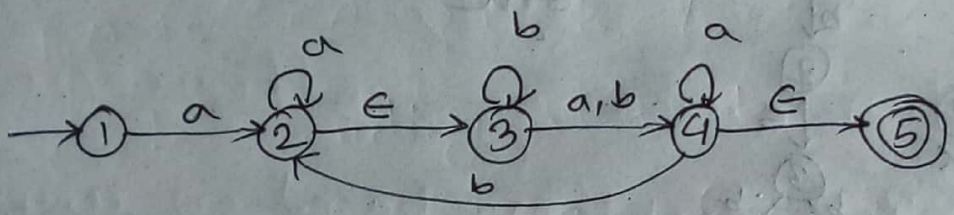
or, $q_1 = b^*a(b + ba^*a)^*$

we are using

(i) $R = Q + RP$

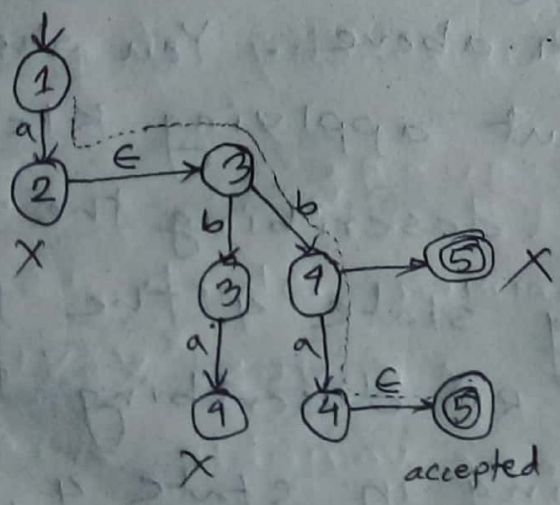
$R = QP^*$

3a

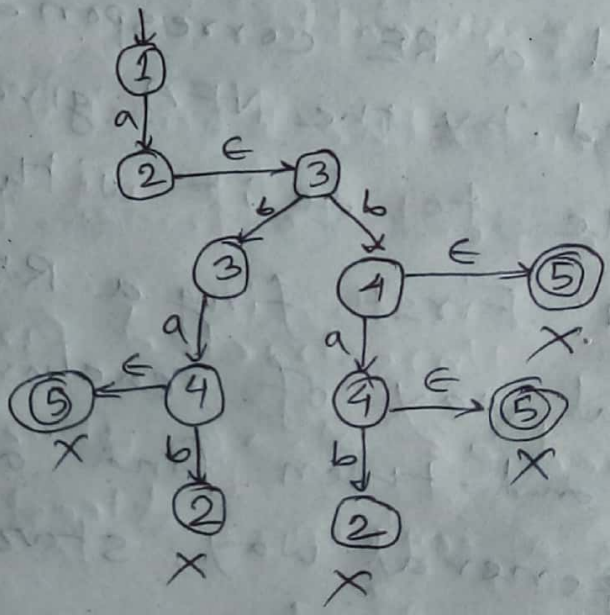


the figure shows a transition diagram for an NFA. For each string below, say whether the NFA accepts it.

i) aba

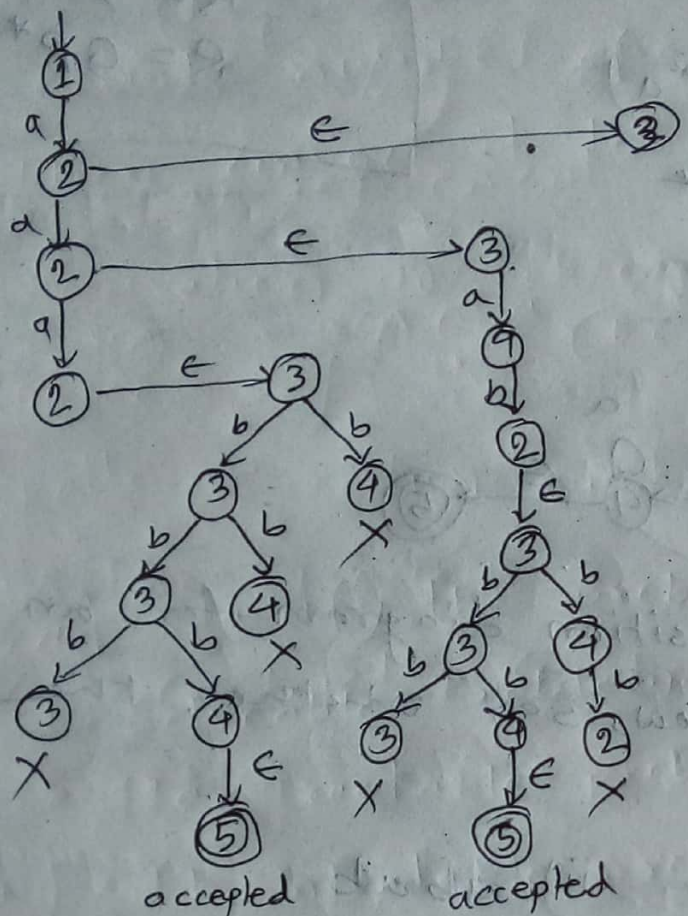


ii) abab



Not accepted

iii) aaabbb



It is enough to show the acceptance in only one occurance

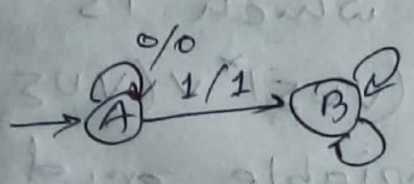
3b Find a RE corresponding to the Language accepted by the NFA given above. You should be able to do it without applying Kleen's theorem; First fing a RE describing the most general way of reaching state 4 the first time, and then find a RE describing the most general way, starting in state 4, of moving to state 4 the next time.

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

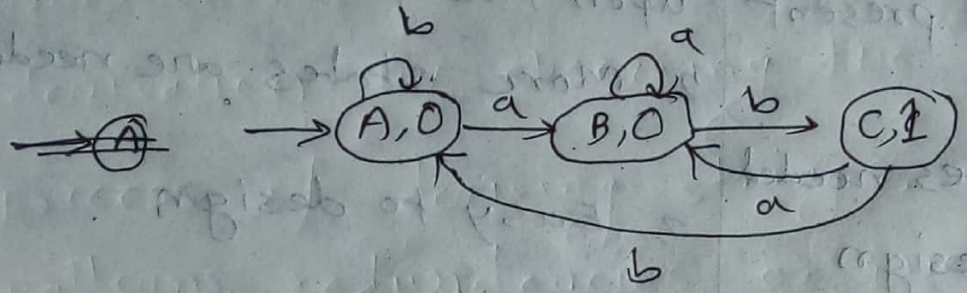
4a Define Mealy Machine. Distinguish between Moore machine and mealy Machine.

Mealy Machine	Moore Machine
1. Output depends on the present state and present input.	1. Output depends only upon the present state
2. Less no. of states needed	2. More states are needed
3. Difficult to design	3. Easy to design
4. It reacts comparatively faster to inputs.	4. Slower
5. If input changes, o/p also changes	5. If input changes, o/p doesn't change.

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



4c Construct a Moore machine that takes strings over alphabet {a,b} as input and counts number of 'ab' as a substring. Then convert it to a Mealy machine



5a Classify the problems based on computation

1. Solveable problem
   1a. Tractable problem (solvable in polynomial time
   1b. Intractable problem (takes too long time)
2. Unsolveable

5b Discuss the Chomsky hierarchy of grammers with respect to fomal languages.

Type-0: Unrestricted Grammer which is defined as $\alpha \to \beta$ where $\alpha \in (V \cup \Sigma)^* V (V \cup \Sigma)^*$ that is α contains atleast one variable and $\beta \in (V \cup \Sigma)^*$

Type-1: Context sensitive Grammer which is defined as $\alpha \rightarrow \beta$ where, $\alpha, \beta \in (V \cup \Sigma)^+$ and $|\alpha| \leq |\beta|$

Type-2: Context free Grammer: $A \rightarrow \alpha$ where $A \in V$, $\alpha \in (V \cup \Sigma)^*$

Type-3: Regular Grammer: $A \rightarrow aB$, $A \rightarrow \epsilon$ where, $A, B \in V$, $a \in \Sigma$. And the rules must be left or Right linear/ regular.

6 a) What is derivation? What is the function of it?

Derivation is a sequence of grammer rule applications that starts with an initial symbol and generates a string of symbols following the production rules of the grammar.

Derivation: The sequence of substitutions to obtain a string is called a derivation

Parse tree: The pictorial representation of derivation of a string is a parse tree.

The prime function of derivation is to generate a Language. As well as it can be used for language recognition. Grammar refinement is also possible by derivation. By observing the step-by-step process of deriving strings, we can identify patterns, redundancies, or inefficiencies in the grammar.

6b Construct CFG that generate the following Languages

i) $\{w \in \{a,b\}^* \mid w$ is any string starting and ending with same symbols$\}$

$S \to aXa \mid bXb \mid a \mid b \mid \in$

$X \to aX \mid bX \mid \in$

ii) $\{w \in \{a,b\}^* \mid w$ is odd length string$\}$

$S \to aX \mid bX$

$X \to aS \mid bS \mid \in$

iii) $\{0^l 1^l 2^n 3^n \mid l, n > 0\}$

6c construct a machine that accepts the language generated from the following grammar. What is the language for this grammar? $A \to Aa \mid Ab \mid Ba$, $B \to Cb$, $C \to \in$

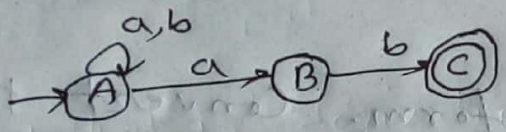
First Reverse the Grammar (Left to Right linear)

$A \to aA \mid bA \mid aB$

$B \to bC$

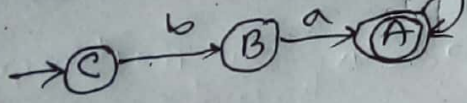
$C \to \varepsilon$

Now draw Finite automata



Now reverse the FA

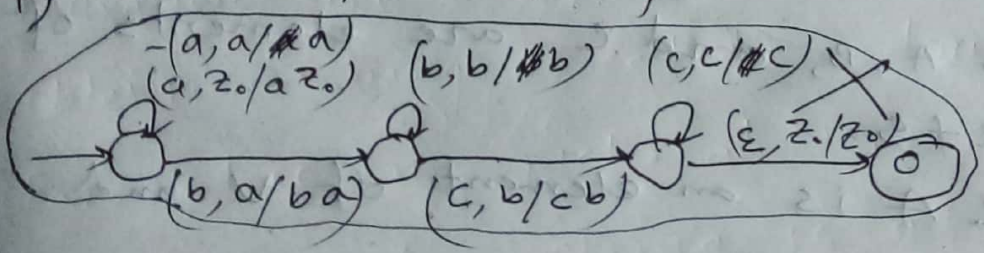
[initial and final states are interchanged

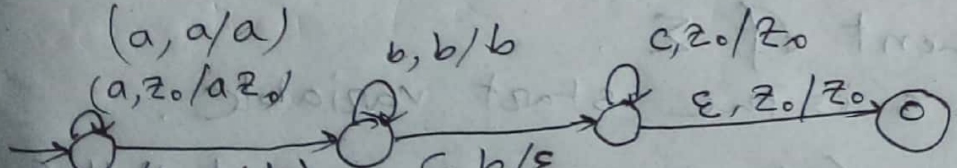


Hence the language is $\{w \in \{a,b\}^* \mid w$ starts with 'ba'$\}$

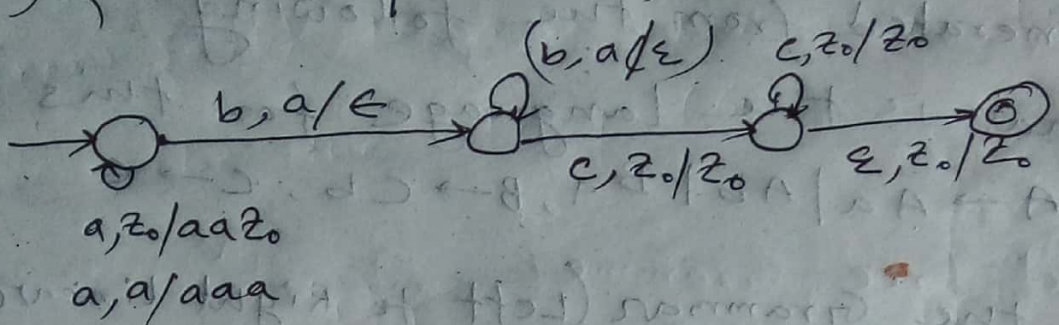
7a Construct PDA for the following CFG

i) $\{a^n b^l c^m \mid l,m,n \geq 1\}$

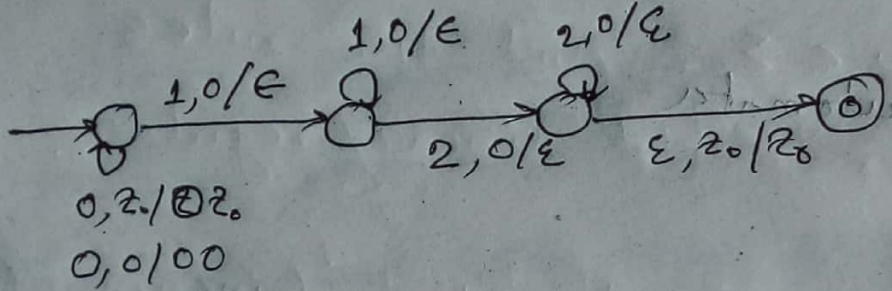




ii) $\{a^n b^{2n} c^m \mid m,n \geq 1\}$



iii) $\{0^{n+m} 1^m 2^n \mid m,n \geq 1\}$



7b Define Chomsky normal form. Convert the following CFG to CNF:

$S \to aX \mid Yb$, $X \to S \mid \varepsilon$, $Y \to bY \mid b$

A CFG $G = (V, \Sigma, R, S)$ is said to be in Chomsky Normal Form, if every rule in R has one of the following three forms:

1. $A \to BC$, where A, B, and C are elements of V, $B \neq S$ and $C \neq S$.
2. $A \to a$, where A is an element of V and a is an element of Σ.
3. $S \to \varepsilon$, where S is the start variable.

Note: $S \to \varepsilon$ iff $\varepsilon \in L(G)$. if so S can't be at R.H.S

~~First remove null production~~

$S \to aX \mid Yb \mid a$

$X \to S$

$Y \to bY \mid b$

~~Remove unit production~~

First introduce a new starting state (because S at RHS)

$S' \to S$

$S \to aX \mid Yb \mid a$

$X \to S \mid \varepsilon$

$Y \to bY \mid b$

(1) Remove null production

$S' \to S$

$S \to aX \mid Yb \mid a$

$X \to S$

$Y \to bY \mid b$

Remove unit production

$S' \to aX \mid Yb \mid a$

$S \to aX \mid Yb \mid a$

$X \to aX \mid Yb \mid a$

$Y \to bY \mid b$

Remove the unreachable state

$S' \to aX \mid Yb \mid a$

$X \to aX \mid Yb \mid a$

$Y \to bY \mid a$

Now construct CNF

$S' \to MX \mid YN \mid M$

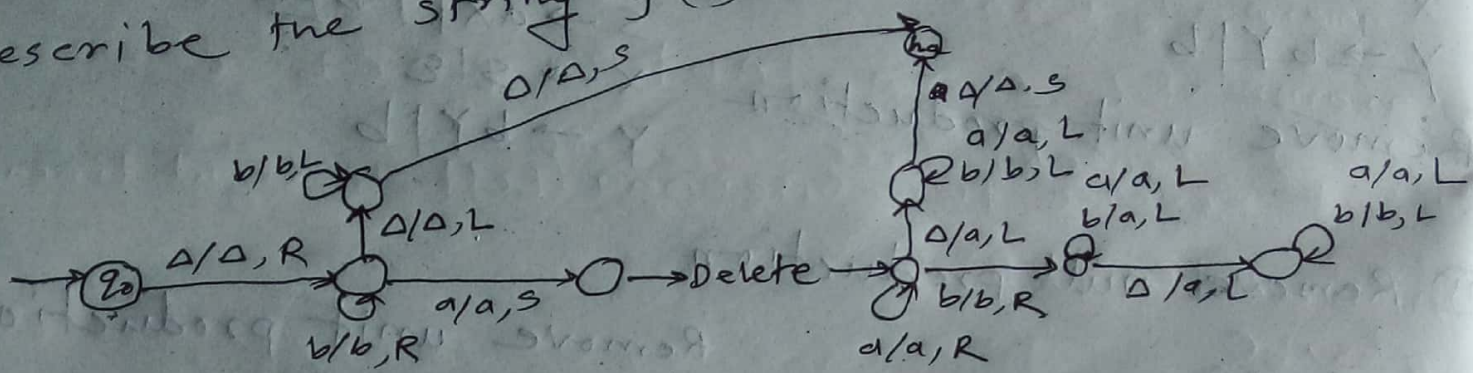
$X \to MX \mid YN \mid M$

$Y \to NY \mid M$

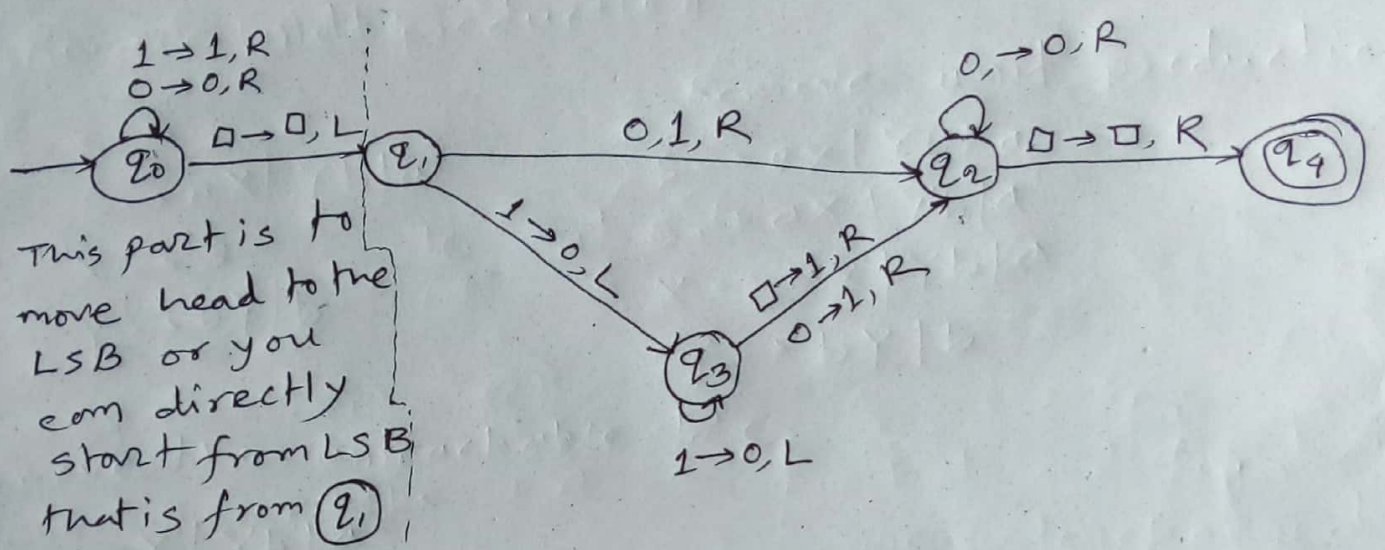
$M \to a$

$N \to b$

8a The TM shown below computes a function from $\{a,b\}^*$ to $\{a,b\}^*$. For any string $x \in \{a,b\}^*$, describe the string $f(x)$



8b Draw a transition diagram for a TM with input alphabet {0, 1} that interprets the input string as the binary representation of a nonnegative integer and adds 1 to it.



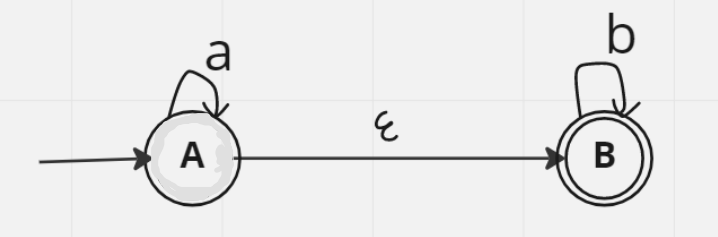
CT-1

Q1: Define Ɛ-NFA. Construct an Ɛ-NFA for the language {$a^m b^n$ where m, n >= 0}

Answer:

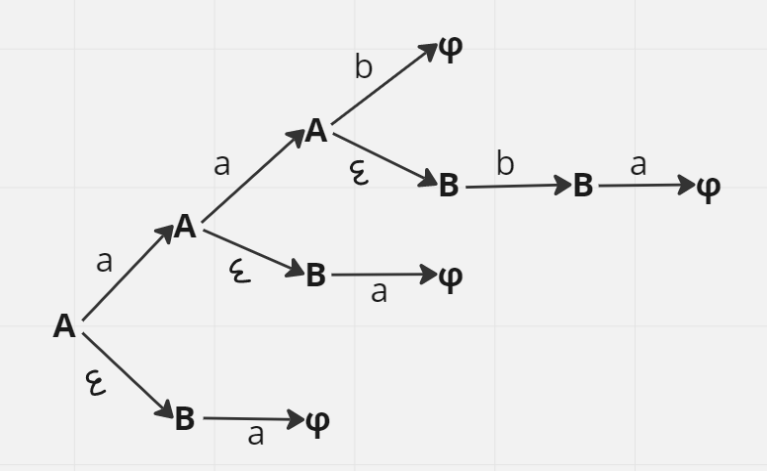
An Ɛ-NFA is a 5-tuple M = (Q, Σ, δ, q, F), where

1. Q is a finite set, whose elements are called states
2. Σ is a finite set, called the alphabet; the elements of Σ are called symbols
3. $\delta : Q \times (\Sigma \cup \varepsilon) \rightarrow 2^Q$ is a function, called the transition function
4. q is an element of Q; it is called the start state
5. F is a subset of Q; the elements of F are called accept states



Q2: Prove that “aabab” is rejected by the above ε-NFA.

Answer:



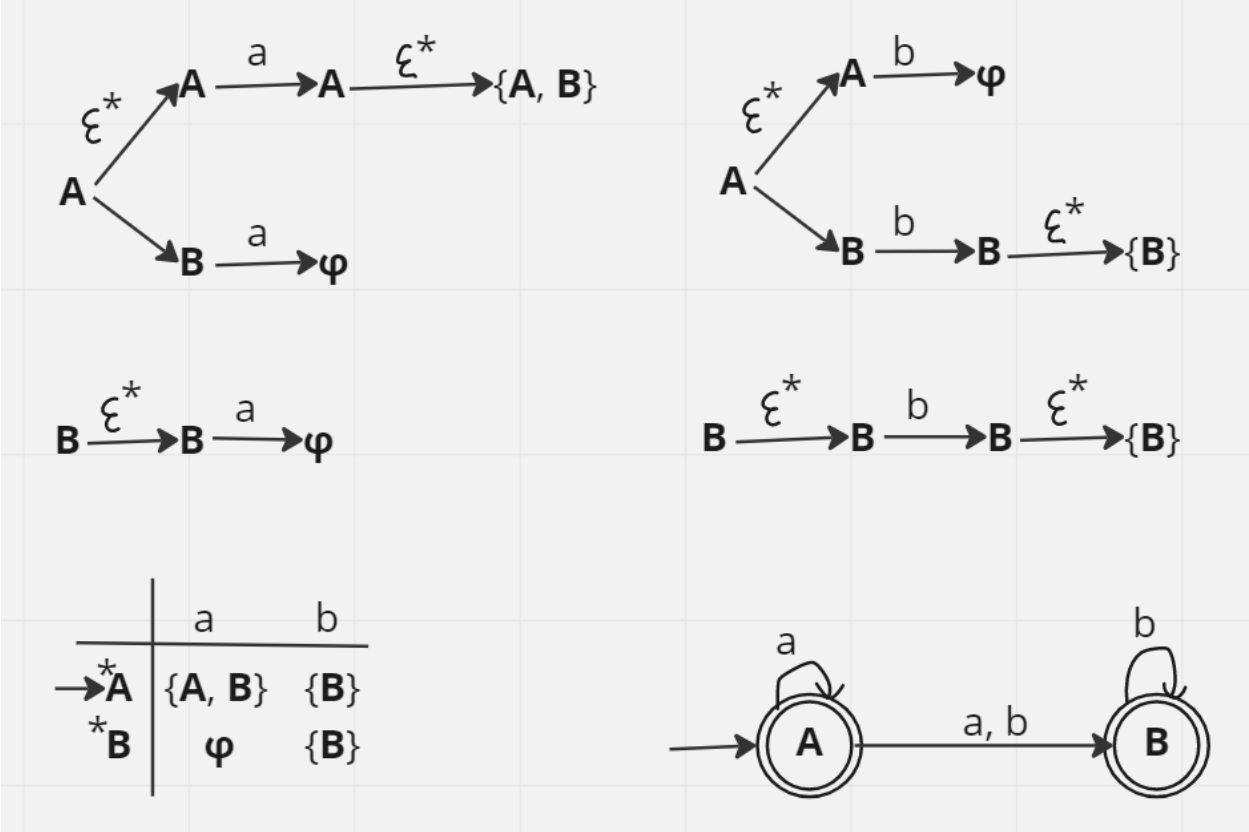
As all the transitions end in a death configuration, the string “aabab” is rejected by the ε-NFA mentioned in Question number 1.

Q3: Convert the ε-NFA to NFA

Answer:

ε^*(closure):- All the states that can be reached from a particular state only by seeing the ε symbol.

Here ε^* of state **A** is {**A, B**} because both are reachable from **A** without consuming any input symbol.



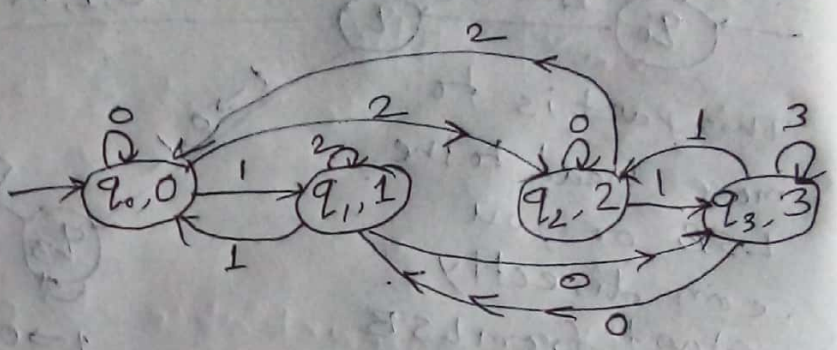
ET-2

Q-1 Construct a Moore Machine that takes base-3 number as input and produce modulo 4 number of its decimal equiv. as input.

Hence, $\Sigma = \{0, 1, 2\}$ and $\Delta = \{0, 1, 2, 3\}$

So, the states are q_0, q_1, q_2 and q_3

	0	1	2	Δ
→q_0	q_0	q_1	q_2	0
q_1	q_3	q_0	q_1	1
q_2	q_2	q_3	q_0	2
q_3	q_1	q_2	q_3	3



2i) Construct grammars for the following languages

{w ∈ {a,b} | w is a string starting and ending with same symbol}

$S \rightarrow aAa \mid bAb \mid a \mid b$

$A \rightarrow aA \mid bA \mid \varepsilon$

ii) {set of all plo palindromic strings over {a, b}}

$S \rightarrow aSa \mid bSb \mid a \mid b \mid \varepsilon$

[3] construct a FA from the following LLG.

① $A \rightarrow Aa \mid Ab \mid Ba$

$B \rightarrow Cb$

$C \rightarrow \varepsilon$

Sec: 2019. [6C]