12/12/19

Formal Language:- A formal language is an abstraction for general characteristics of programming language that can be defined as set of strings over alphabets.

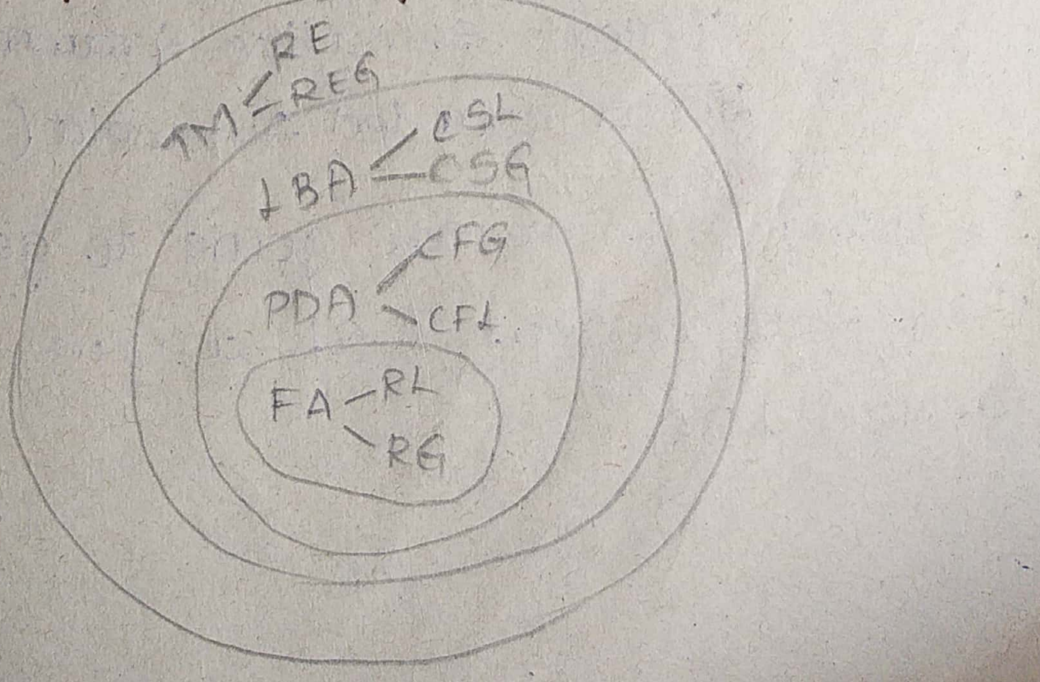
→ A Formal language is a set of finite length words drawn from some finite alphabet.

→ The language is denoted by L

Automaton:- It is defined as a system where energy, materials and information are transformed, transmitted and used for performing some function without direct participation of human.

Ex:- Automatic printing, packing

Chomsky's Hierarchy



level-1 :- Regular Language (RL)

(T-3) Regular Grammar (RG)

FA - Finite Autometa

By using RL we are going to generate RG to accept the RL we are designing FA (Finite Autometa)

level-2 :- context free Language (CFL)

(T-2) context free Grammar (CFG)

push down Autometa (PDA)

By using CFL we are going to generate CFG, to accept CFL we are going to draw PDA

level-3 :- Context sensitive Language (CSL)

(T-1) Context sensitive Grammar (CSG)

Linear Bounded Autometa (LBA)

By using CSL we are going to generate CSG, to accept CSL we are going to design LBA level-4 :- Recursively Enumerable language (RE)

T-0 Recursively enumerable Grammar [REG] or unrestricted Grammar [UG]

T.M - Turing Machine

By using RE we are going to generate REG, to accept RE we are going to design TM

Symbol :- symbol is a single object and it is a building block

Ex :- A to Z, a to z, 0 to ∞, *, & -- etc.

Alphabet :- It is any finite, collection of the symbol. It is denoted by "Σ"

wher $\Sigma = \{a,b\}$, $\{a,b,c\}$ or $\{0,1\}$

String :- It is a finite sequence of symbols choosen from some alphabet

(or)

A string is a finite sequence of symbols over an alphabet

→ It is denoted by "W" or "S"

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

Empty string:- It is a string with zero occurrences of the symbol.

It is denoted by "ϵ".

length of the string:- No. of symbols present in a string

Ex:- $|10111| = 5$

$|aba| = 3$

$|\epsilon| = 0$

Language:- A set of strings over some Alphabet where alphabet $\Sigma = \{a, b\}$

$L1$ = set of all strings of length 2

$L1 = \{aa, ab, bb, ba\}$

$L2$ = Set of all strings of length 3

$L2 = \{aaa, abb, aba \ldots\}$

$L3$ = Set of all strings where each string starts with a (Infinite)

$L3 = \{a, ab, aa, aba, aab, \ldots\}$

NOTE:- A language can be made over an alphabet Σ may be finite or Infinite

14/12/19

powers of Σ: [cardinality]

Set of all strings of certain length from alphabet and it is denoted by Σ^n where n is length

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

Σ^1 = set of all string over Σ of length 1

$2^1 = 2 = \{a, b\}$

Σ^2 = set of all strings over Σ of length 2

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

Σ^n = set of all strings over Σ of length n

2^n

Σ^0 = set of all strings over Σ of length 0

$\Sigma^0 = \{\epsilon\}$ $|\epsilon| = 0$

→ Σ closure : Σ^*

$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup --- \Sigma^n$

→ cardinality :

It is nothing but no. of strings or elements present in a set where $|\Sigma^n| = 2^n$

Finite Automaton: [FA]

It is an abstract computing device, it is a mathematical model of a system with discrete inputs, outputs, states and set of transitions from one state to another that occurs on Input symbols over alphabets

→ FA can be represented by using 3 ways

i) Graphical representation [Transition diagram]

ii) Tabular representation [Transition table]

iii) Mathematical representation [Transition function or mapping]

Formal definition of FA:

A FA contains 5 tupples they are

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

where Q is finite set of all states

Σ is finite set of Input symbols

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

is the transition function

q_0 is the initial state or start state

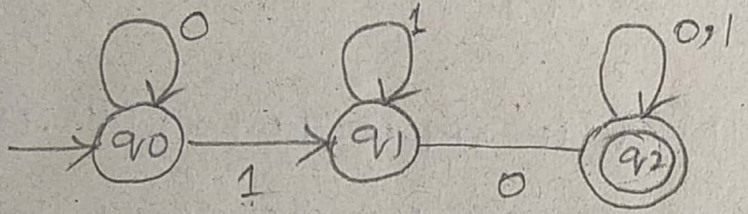
represented by →○ circle with an arrow

where $q_0 \in Q$

F is the set of acceptance states or final states.

represented by (O)

Graphical representation of FA:



Transition diagram

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

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

$q_0 = q_0$

$F = q_2$

It is a directed graph associated with the vertices of the graph corresponds to the states of FA.

Transition table:

It is basically a tabular representation of the transition function that takes 2 arguments [a state & a symbol] and returns a value as next state.

rows represent states

colum represent inputs, Entries represent next state

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

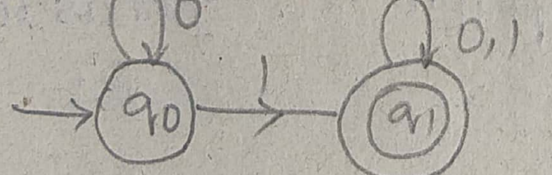
→ Start or initial state represent with arrow

→ final or acceptance state represent with
$*$ or q_2

16/12/19

Mapping function or Transition function:-

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



$\delta : q_0 \times 0 \rightarrow q_0$ or $\delta(q_0, 0) = q_0$

$q_0 \times 1 \rightarrow q_1$

$q_1 \times 0 \rightarrow q_1$

$q_1 \times 1 \rightarrow q_1$

Applications of FA:-

- It plays an important role in compiler design
- In switching theory & design & analysis of digital circuit automata theory is applied.
- design and analysis of complex hardware & software systems.
- To prove the correctness of the program automata theory is used.
- To design finite state machines such as Moore machine, mealy machine.
- It is the base for the formal Languages, and these formal languages are useful for the programming languages.

Types of FA:-

- FA
  - Without o/p
    - DFA
    - NFA
    - ε-NFA
  - With o/p
    - Moore
    - Mealy

DFA [Deterministic Finite Automata] :-

The finite automata are called DFA if the machine is read an input string one symbol at a time.

→ In DFA there is only one path for specific input from the current state to next state.

→ deterministic refers to the uniqueness of the computation.

→ DFA doesn't accept null moves (ϵ) i.e DFA cannot change state without any input character.

→ DFA can contain single initial state and multiple final states.

→ It is used in lexical analysis in compilers

Formal definition of DFA :-

$$DFA = \{Q, \Sigma, \delta, q_0, F\}$$

Acceptance of Language :-

→ A language acceptance is defined by

" If a string W is accepted by machine M i.e if it is reaching the final state F by taking the String W

→ Non-accepted :-

If the string is not reaching the final state

17/12/19

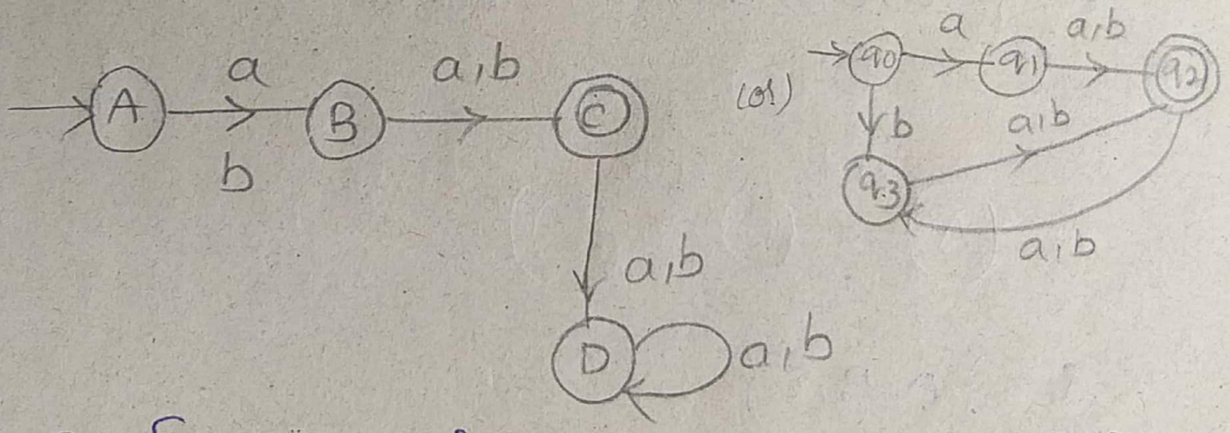
construct a DFA that accepts set of all strings over {a,b} of length 2

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

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

⇒ min string length + 1 ⇒ min no. of states

⇒ $2+1=3$



$Q = \{A,B,C,D\}$, $q_0 = \{A\}$, $F = \{C\}$

δ: $\delta(A,a) = B$ $\quad$ $\delta(C,a) = D$

$\delta(A,b) = B$ $\quad$ $\delta(C,b) = D$

$\delta(B,a) = C$ $\quad$ $\delta(D,a) = D$

$\delta(B,b) = C$ $\quad$ $\delta(D,b) = D$

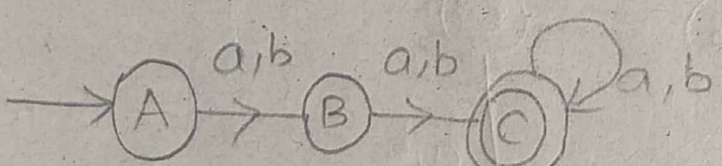
Scanned with CamScanner

Q \ Σ	a	b
A	B	B
B	C	C
C	D	D
D	D	D

→ construct a DFA that accepts set of all strings over $\{a, b\}$ such that the length of the string is at least 2

$|W| \geq 2$

$L = \{aa, ab, ba, bb, aaa, aab, aba, baa ---\}$



$Q = \{A, B, C\}$

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

$q_0 = \{A\}$

$F = \{C\}$

δ: $\delta(A, a) = B$ $\quad \delta(B, b) = C$

$\delta(A, b) = B$ $\quad \delta(C, a) = C$

$\delta(B, a) = C$ $\quad \delta(C, b) = C$

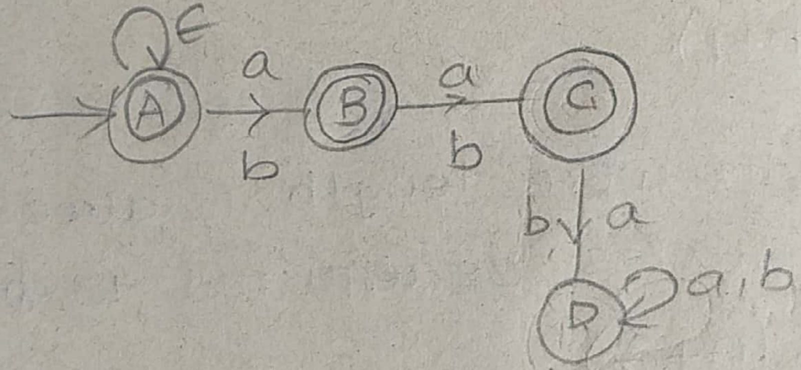
Q \ Σ	a	b
A	B	B
B	C	C
C	C	C

→ construct a DFA that accepts set of all strings over $\{a,b\}$ such that the length of the string is atmost 2

$$|W| \leq 2$$

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

but DFA doesn't accept ϵ



$Q = \{A, B, C, D\}$

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

$q_0 = \{A\}$

$F = \{A, B, C\}$

$\delta(A, \epsilon) = A$

$\delta = \{\delta(A, a) = B$

$\delta(A, b) = B$

$\delta(B, a) = C$

$\delta(B, b) C$

$\delta(C, a) = D$

$\delta(C, b) = D$

$\delta(D, b) = D, \delta(D, a) = D$

Q \ Σ	a	b
A	B	B
B	C	C
C	D	D
D	D	D

** whenever the length of the word is equal to n. then the required no. of states are $(n+2)$

* whenever the word length is atleast 'n' then the states required to draw a DFA state is $(n+1)$.

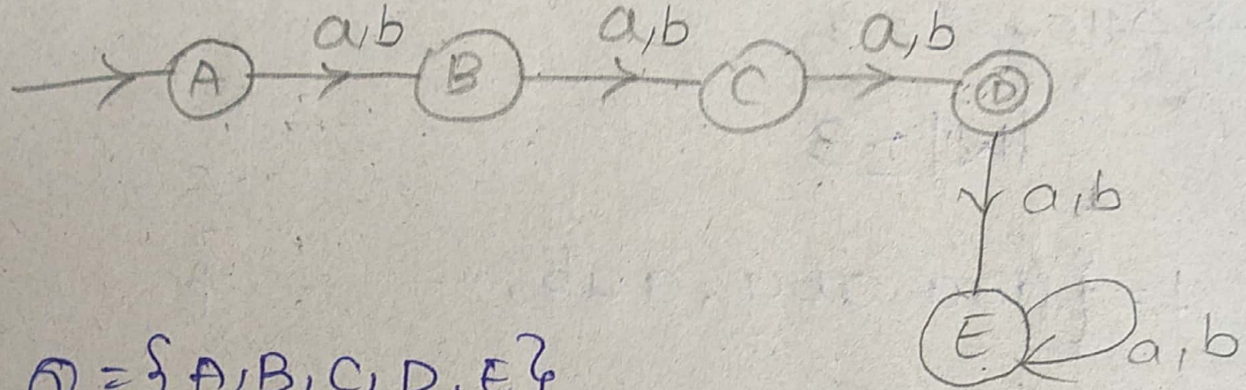
* whenever the word length is atmost 'n' then the states ~~de~~ required to draw a DFA is $(n+2)$.

→ construct a DFA that accepts set of all strings over $\{a, b\}$ of length 3

$|w| = 3$

$L = \{aaa, aab, aba, abb, baa, bab, bba, bbb\}$

Scanned with CamScanner



$Q = \{A, B, C, D, E\}$

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

$q_0 = \{A\}$

$F = \{D\}$

δ : $\delta(A, a) = B$ $\delta(C, a) = D$ $\delta(E, a) = E$

$\delta(A, b) = B$ $\delta(C, b) = D$ $\delta(E, b) = E$

$\delta(B, a) = C$ $\delta(D, a) = E$

$\delta(B, b) = C$ $\delta(D, b) = E$

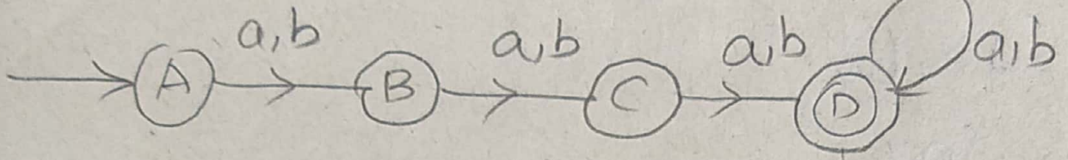
→ ~~Atleast 3~~

Q \ Σ	a	b
A	B	B
B	C	C
C	D	D
D	E	E
E	E	E

→ Atleast 3

$|W| \geq 3$

$L = \{aaa, aba, aab, \text{-------}\}$



$Q = \{A, B, C, D\}$

$q_0 = \{A\}$

$F = \{D\}$

$\Sigma = \{a, b)$

$\delta(A, a) = B$

$\delta(A, b) = B$

$\delta(B, a) = C$

$\delta(B, b) = C$

$\delta(C, a) = D$

$\delta(C, b) = D$

$\delta(D, a) = D$

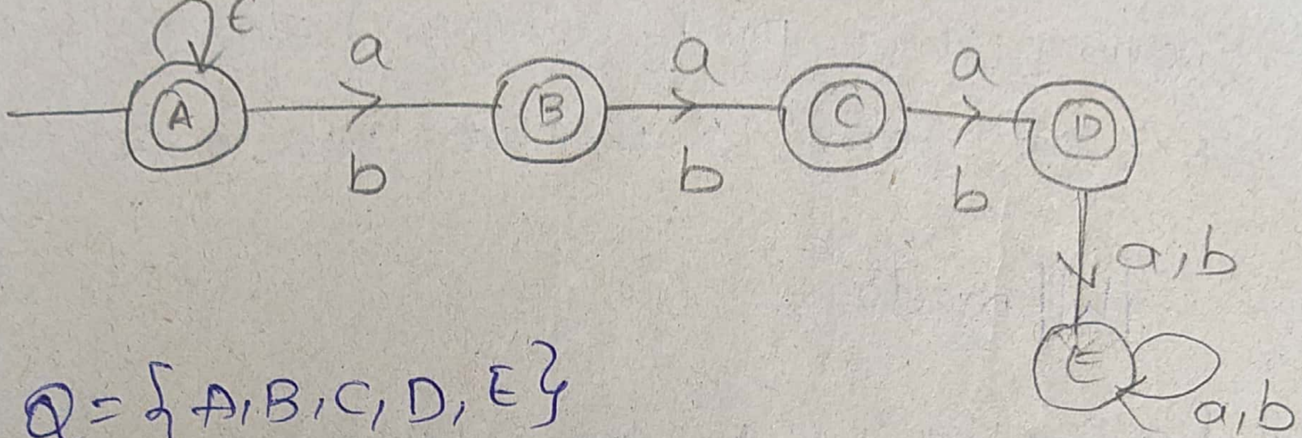
$\delta(D, b) = D$

$Q \backslash \Sigma$	a	b
A	B	B
B	C	C
C	D	D
D	D	D

→ Atmost 3

$|W| \leq 3$

$L = \{\epsilon, a, b, aa, ab, bb, ba, aaa, aab, aba, abb, baa, bab, bba, bbb\}$



$Q = \{A, B, C, D, E\}$

$q_0 = \{A\}$

$F = \{A, B, C, D\}$

δ: $\delta(A, \epsilon) = A$ $\delta(C, a) = D$ $\delta(E, a) = E$

$\delta(A, a) = B$ $\delta(C, b) = D$ $\delta(E, b) = E$

$\delta(A, b) = B$ $\delta(D, a) = E$

$\delta(B, a) = C$ $\delta(D, b) = E$

$\delta(B, b) = C$

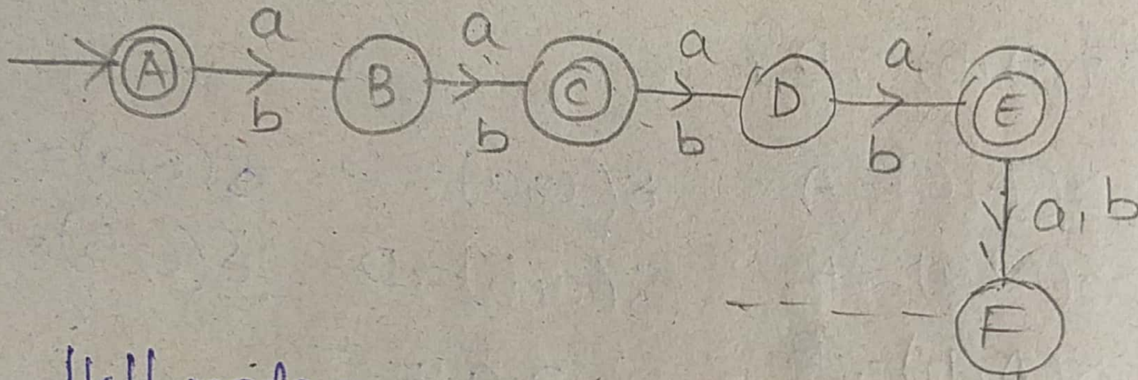
Q \ Σ	a	b
A	B	B
B	C	C
C	D	D
D	E	E
E	E	E

Scanned with CamScanner

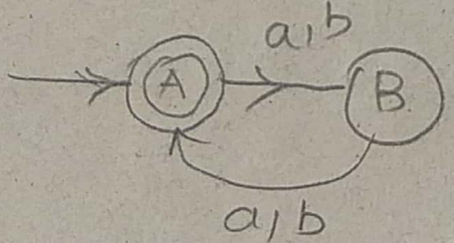
→ Construct a DFA where $\Sigma = \{a, b\}$ that is going to accept $|W| \bmod 2 = 0$

$|W| \bmod 2 = 0$

$L = \{\epsilon, aa, ab, bb, ba, aaaa, aaab - - - \}$
(even numbers)

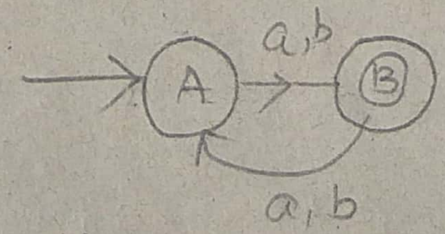


$|W|$ mode $n = 0$ $= n$ states



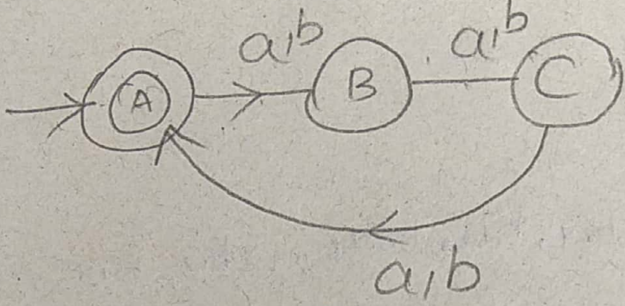
→ $|W| \bmod 2 = 1$

$L = \{a, b, aaa, - - - -\}$ (odd numbers)



→ $|W| \bmod 3 = 0$

$L = \{\epsilon, aaa, bbb, aaaaaa, ---\}$

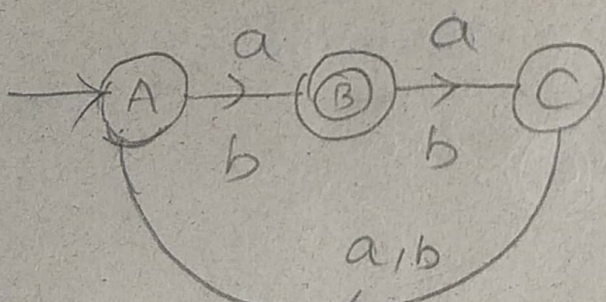


$q_0 = A$

$F = \{A\}$

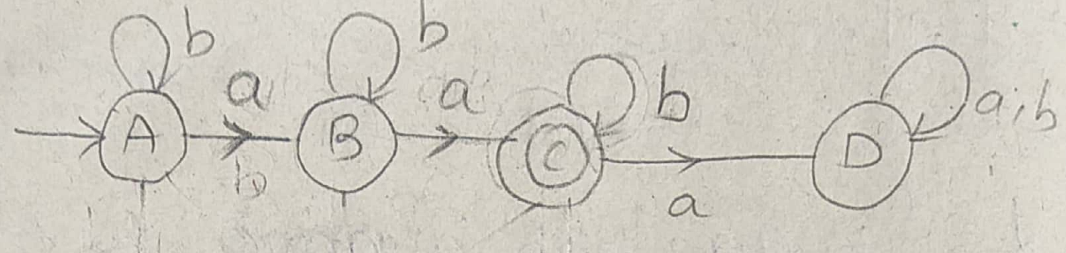
→ $|W| \bmod 3 = 1$

$L = \{a, b, aaaa, ---$



→ construct minimal DFA set of All strings over $\{a, b\}$ where number of a's in a string is 2.

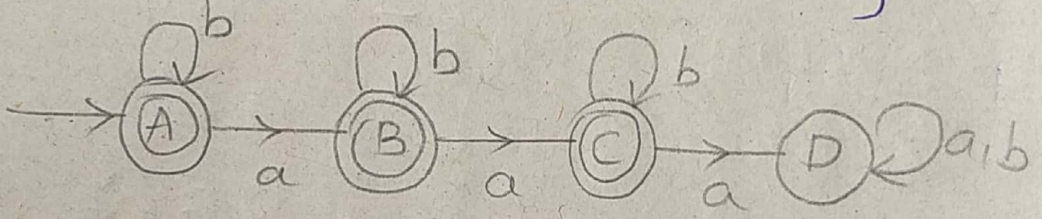
$L = \{aa, aba, aab, baa, bbaa,$



Atmost 2

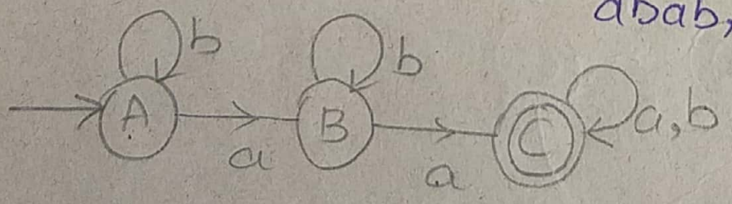
≤ 2

$L = \{ \epsilon, a, ab, ba, bb, aab, aba, bba, bbb,$
$abbb \ldots \}$



Atleast 2

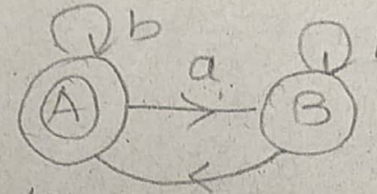
≥ 2 $L = \{ aa, aab, aaab, aba, baa,$
$abab, baaa \ldots \}$



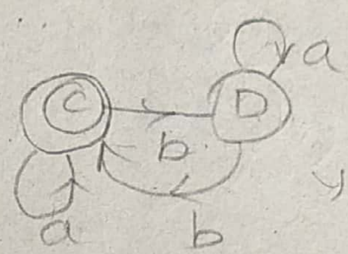
19/12/19

construct a DFA that accepts set of all strings over $\{a,b\}$ in $n_a(w) \cong 0 \bmod 2$ & $n_b(w) \cong 0 \bmod 2$.

$L = \{ aabb, abab, aaaabbbb, abababab \ldots \}$



even no. of a's



even no. of b's

$\{A, B\} \times \{C, D\} = \{AC, AD, BC, BD\}$

states of even no. of a's

states of even no. of b's

$AC \xrightarrow{a} BC$ $\left\{ \begin{array}{l} A \xrightarrow{a} B \\ C \xrightarrow{a} C \end{array} \right\} BC$

$AC \xrightarrow{b} AD$

$AD \xrightarrow{a} BD$

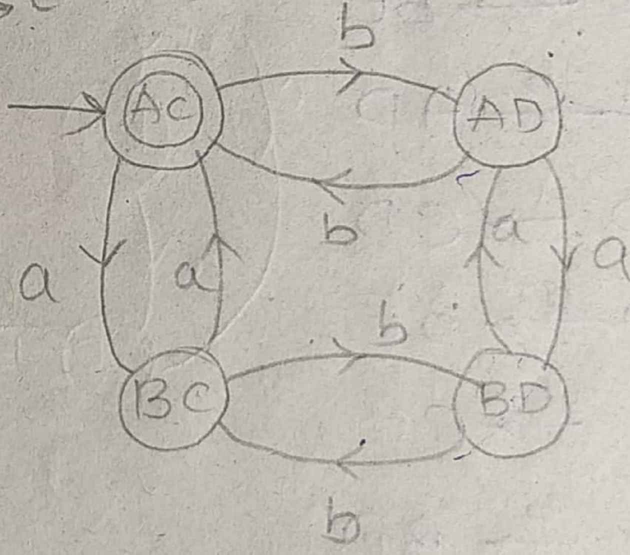
$AD \xrightarrow{b} AC$

$BC \xrightarrow{a} AC$

$BC \xrightarrow{b} BD$

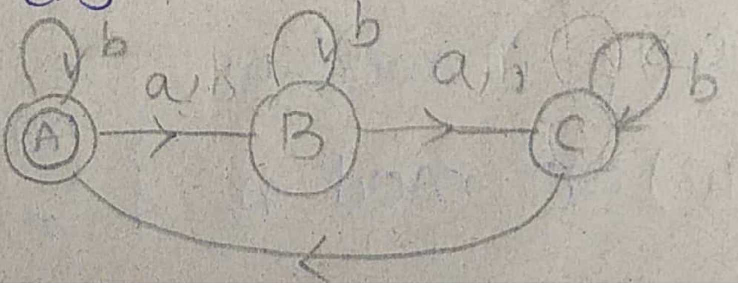
$BD \xrightarrow{a} AD$

$BD \xrightarrow{b} BC$

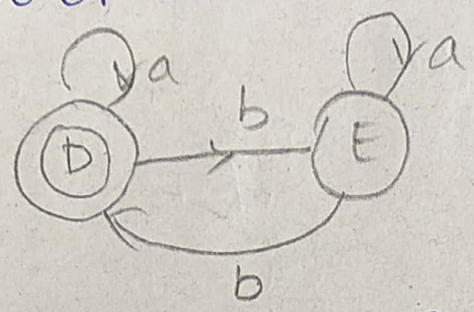


→ construct a DFA, $w \in \{a, b\}^*$ where $n_a(w) = 0 \bmod 3$, $n_b(w) = 0 \bmod 2$

odd no. of a's



even no. of b's



$\{A, B, C\} \times \{D, E\} \Rightarrow \{AD, AE, BD, BE, CD, CE\}$

$AD \xrightarrow{a} BD$

$AD \xrightarrow{b} AE$

$AE \xrightarrow{a} BE$

$AE \xrightarrow{b} AD$

$BD \xrightarrow{a} CD$

$BD \xrightarrow{b} BE$

$BE \xrightarrow{a} CE$

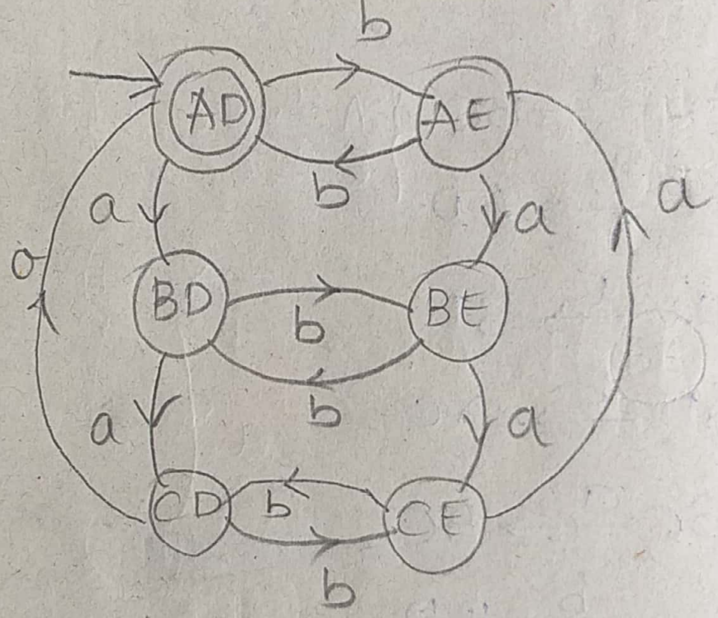
$BE \xrightarrow{b} BD$

$CD \xrightarrow{a} AD$

$CD \xrightarrow{b} CE$

$CE \xrightarrow{a} AE$

$CE \xrightarrow{b} CD$



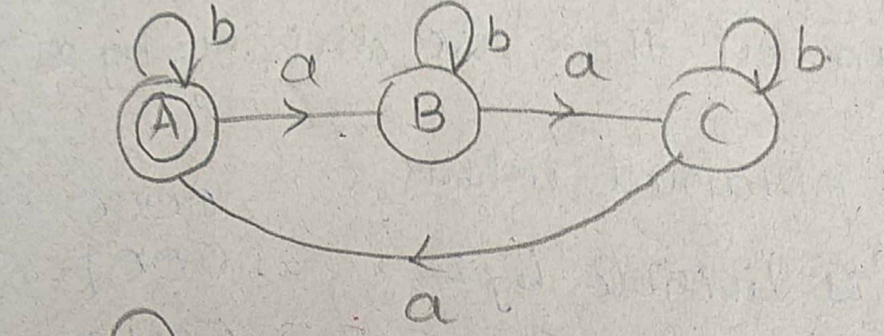
** $n_a(w) = 0 \bmod m$

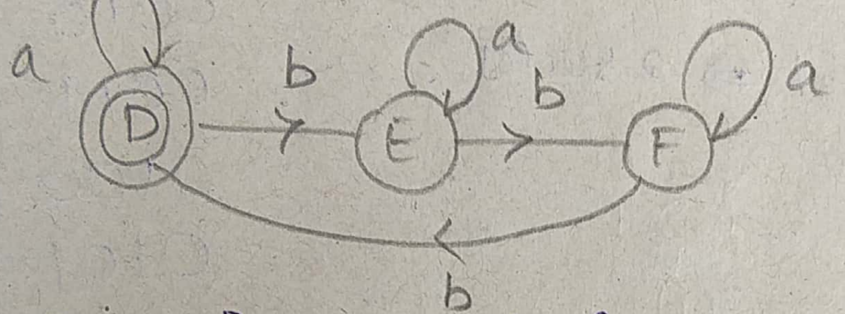
* $n_b(w) = 0 \bmod n$ } no. of states $= m \times n$

Scanned with CamScanner

→ construct a DFA, $w \in \{a,b\}^*$

$n_a(w) = 0 \bmod 3$, $n_b(w) = 0 \bmod 3$





$\{A,B,C\} \times \{D,E,F\} = \{AD, AE, AF, BD, BE, BF, CD, CE, CF\}$

$AD \xrightarrow{a} BD$

$AD \xrightarrow{b} AE$

$AE \xrightarrow{a} BE$

$AE \xrightarrow{b} AF$

$AF \xrightarrow{a} BF$

$AF \xrightarrow{b} AD$

$BD \xrightarrow{a} CD$

$BD \xrightarrow{b} BE$

$BE \xrightarrow{a} CE$

$BE \xrightarrow{b} BF$

$BF \xrightarrow{a} CF$

$BF \xrightarrow{b} BD$

$CD \xrightarrow{a} AD$

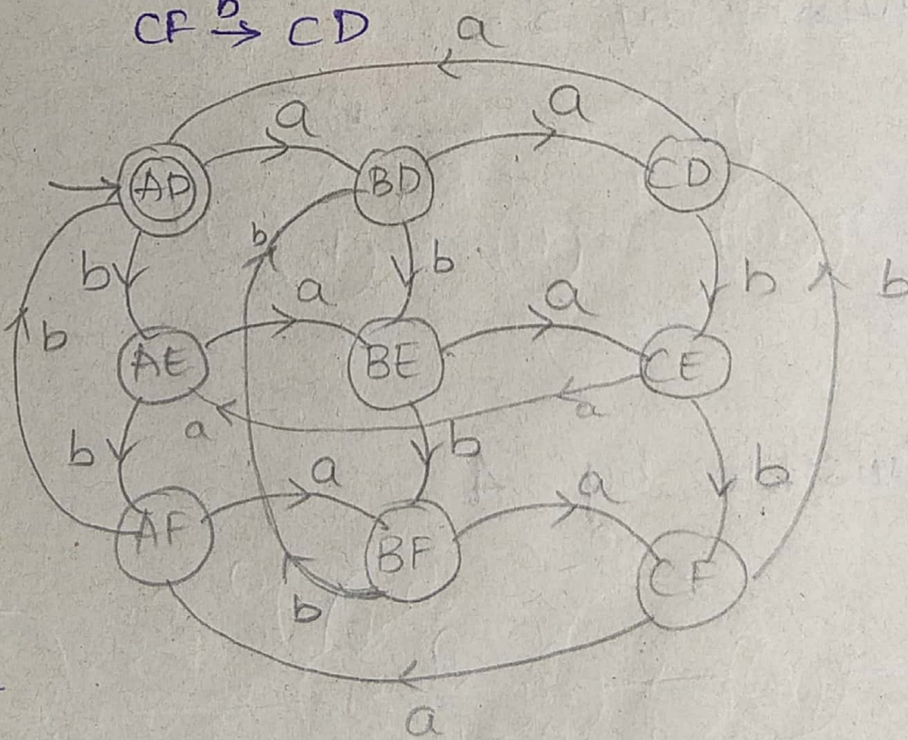
$CD \xrightarrow{b} CE$

$CE \xrightarrow{a} AE$

$CE \xrightarrow{b} CF$

$CF \xrightarrow{a} AF$

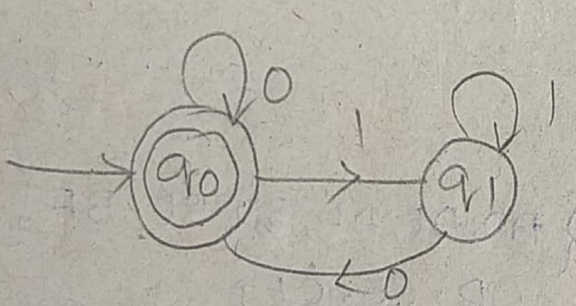
$CF \xrightarrow{b} CD$



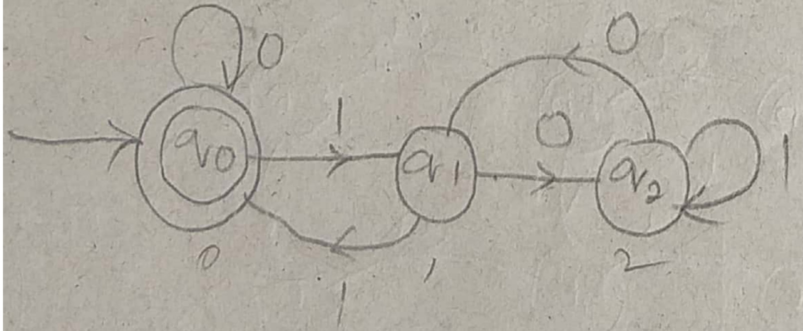
construct a minimal DFA which accepts set of all strings over {0,1} which accepts benary numbers those are divisible by 2

remainders obtained when
a number is divisible by 2
are 0,1 ⇒ 2 states

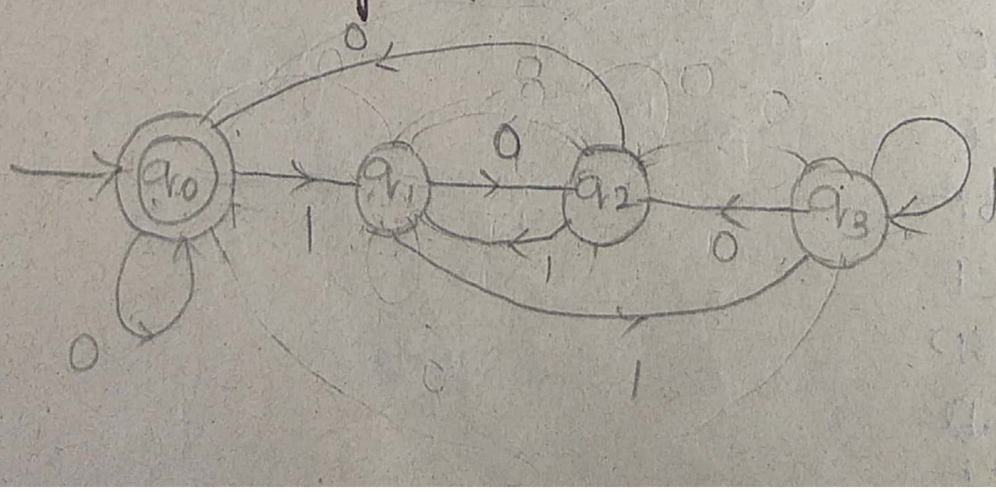
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	
1000	
1001	



divisible by 3



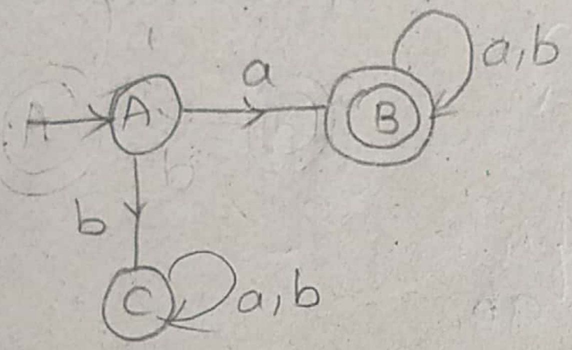
divisible by 4



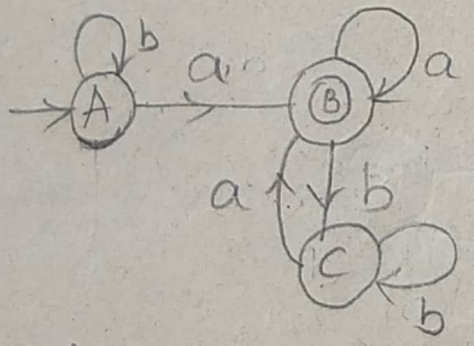
21/12/19

→ construct a DFA over {a,b} which accepts set of all strings starting with a.

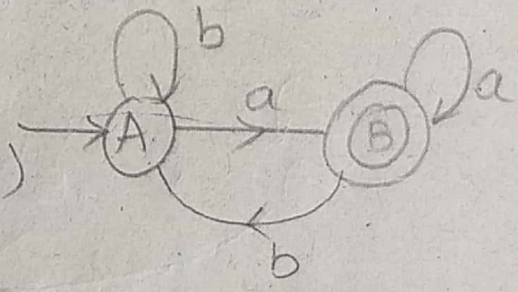
$L = \{a, aa, ab, aab, aa---\}$



→ ends with a $L = \{ba, a, bba, aabba--$

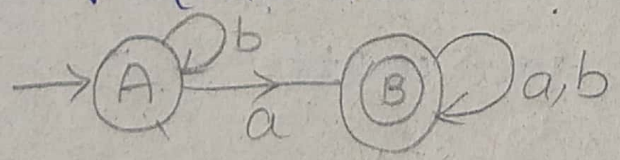


(or)



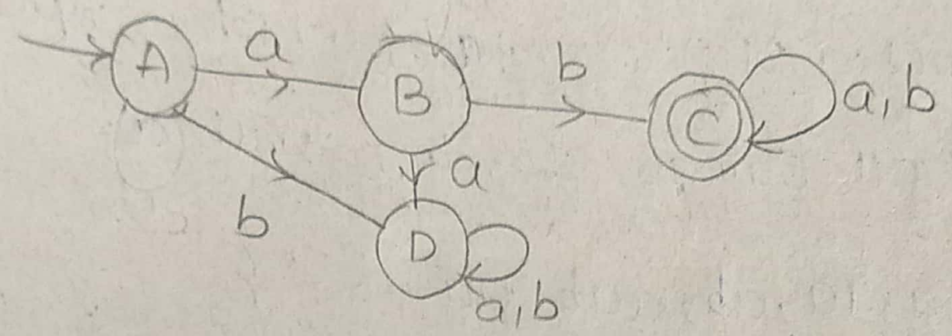
→ construct a DFA over {a,b} which accepts set of all strings containing a

$L = \{a, ab, ba, bba, aab, ---\}$



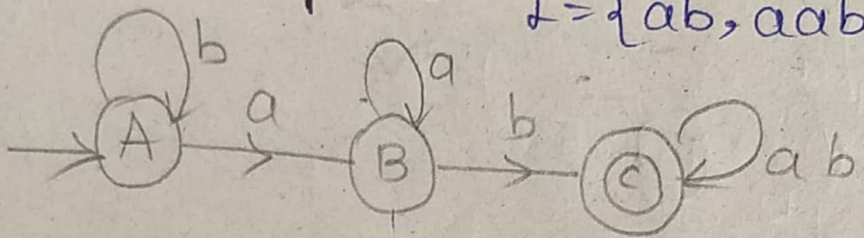
→ construct a DFA over {a,b} which accepts set of all string starting with ab

$L = \{ab, abb, aba, abba, abbb----$



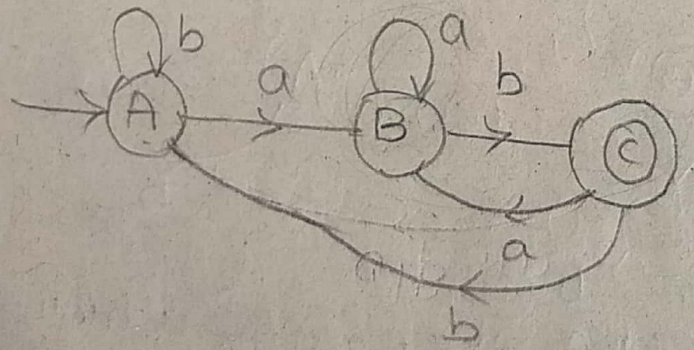
→ containing ab

$L = \{ab, aab, bab, ---\}$



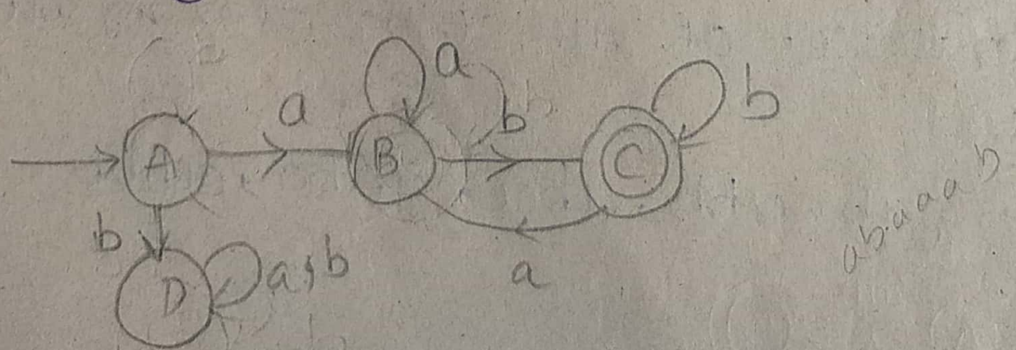
→ ending with ab

$L = \{ab, aab, bab, aaab, bbbab. ---\}$



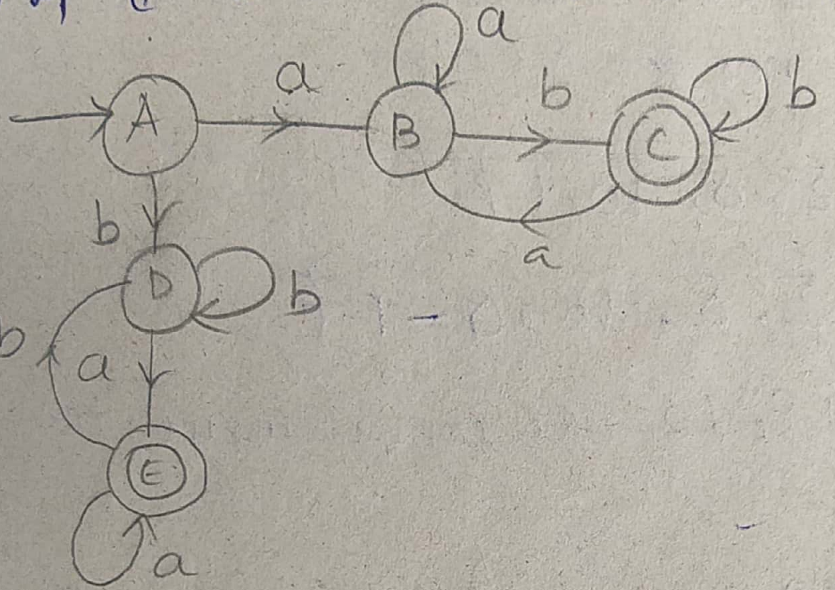
→ construct a DFA over {a,b} which accepts set of all strings starting with a and end with b

$L = \{ab, aabba, aaab, ---$

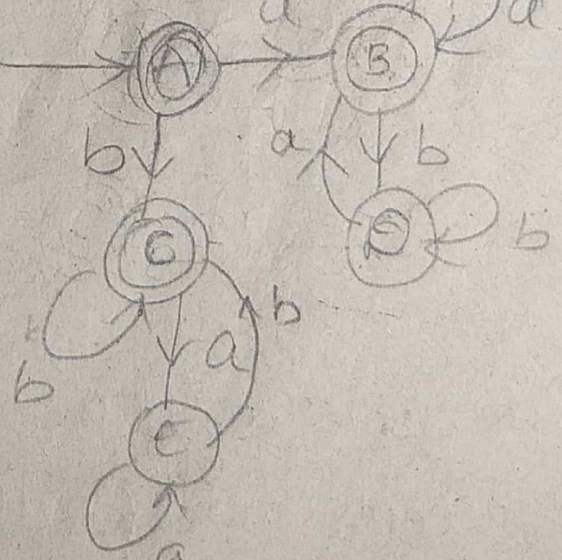


→ construct a DFA over {a,b} which accepts set of all strings starts and ends with i) different symbols. (a-b or b-a) ii) same symbol.

i) L_1 = {ab, aab, abb, abbbb, aabab, -- ba, baa, baaba, --



ii) L_2 = {a, b, aba, aa, bb, bab, ababa, bbbb, baab, - - - -}



23/12/19

→ complementation method is nothing but converting non-final state to final state & final states to non-final state.

→ Complementation method only applies for only DFA's nor for NFA's

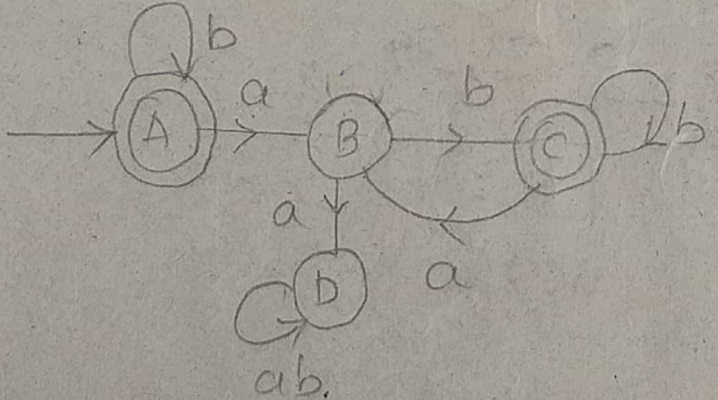
$L1 = \{Q, \Sigma, \delta, q_0, F\}$

$L2 = \{Q, \Sigma, \delta, q_0, Q-F\}$

here both L1 & L2 are complement to each other

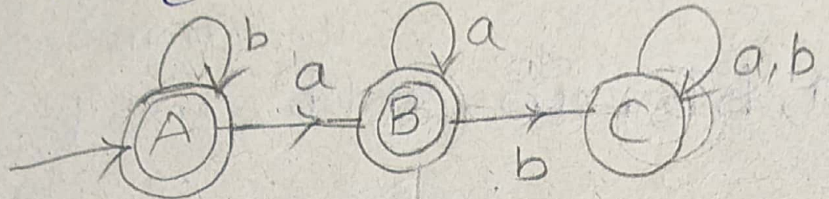
construct a DFA over {a,b} which accepts set of all strings where every 'a' should be followed by 'b'.

$L = \{\epsilon, ab, abab, bab, b, bb, ----\}$



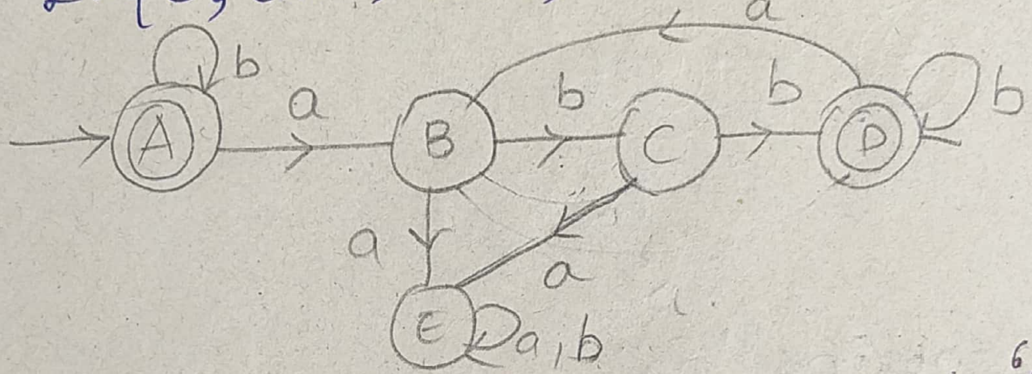
→ construct a DFA over {a,b} which accepts set of all string in which every 'a' is never followed by b

$L = \{\epsilon, aa, bb, baa, ba, ----$



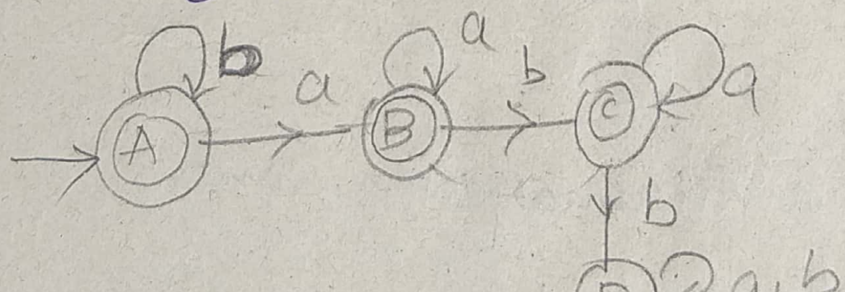
→ construct a DFA over {a,b} which accepts set of all strings where every 'a' should be followed by 'bb'

$L = \{\epsilon, abb, bbb, babbabb, ---$



→ 'a' should never followed by 'bb'

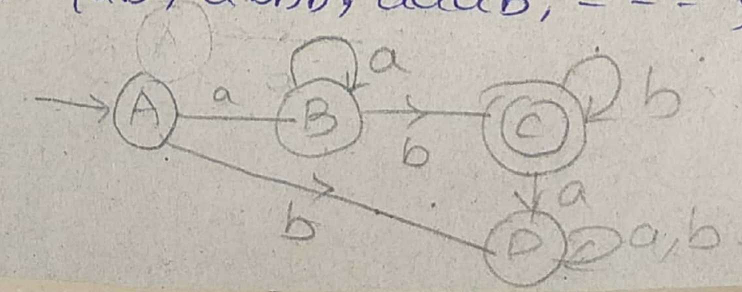
$L = \{\epsilon, aa, bb, ab, ba, baa$



→ construct a DFA over {a,b} where

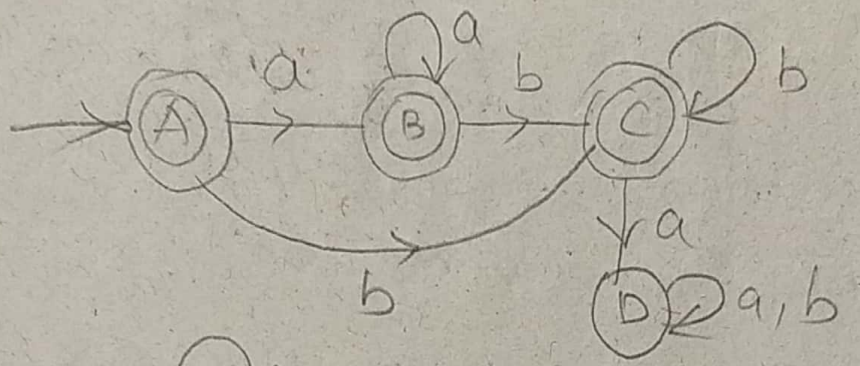
$L = \{a^n b^m \,/\, n, m \geq 1\}$

$L = \{ab, abbb, aaab, --- \}$

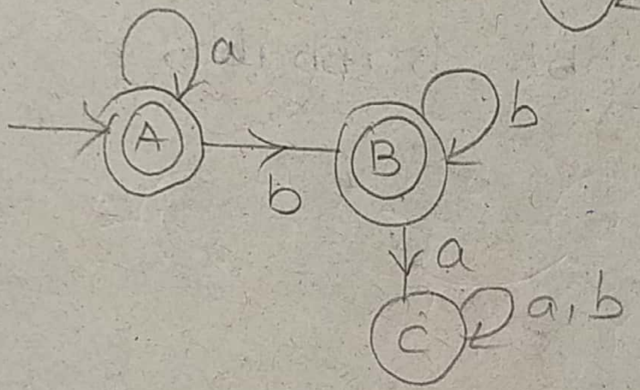


→ construct a DFA where $L = \{a^n b^m / n, m \geq 0\}$

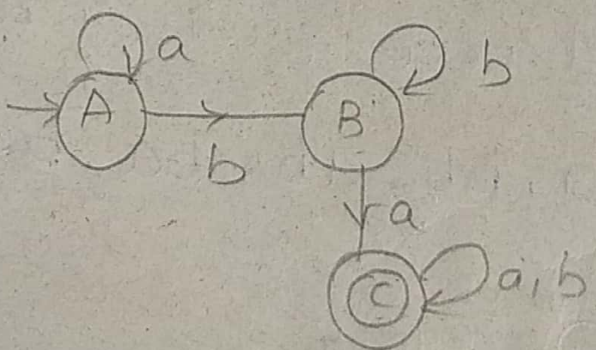
$L = \{\epsilon, aa, bb, ab, abbb, aaab, \ldots\}$



(or)



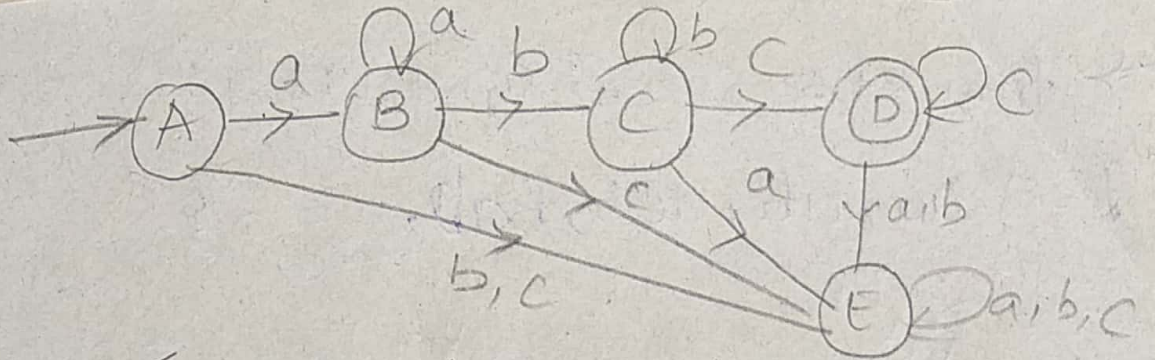
→ contains 'ba'



→ ~~construct a minimal DFA where~~

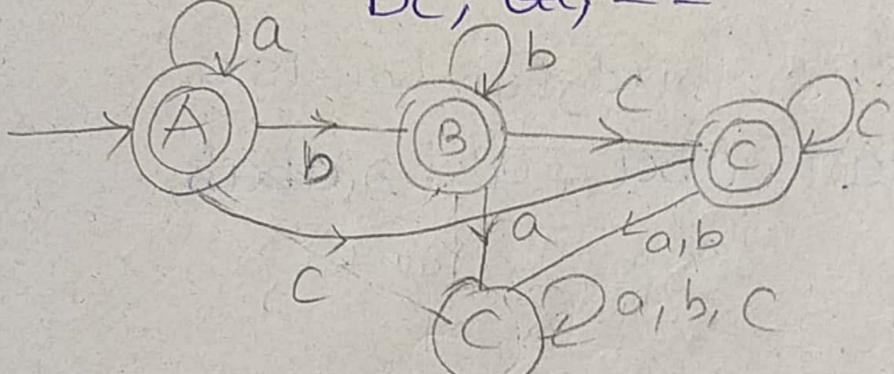
$L = \{a^n b^m c^l / n, m, l \geq 1\}$

$L = \{abc, aabbcc, abccc, aaabc, abbbc, \ldots\}$



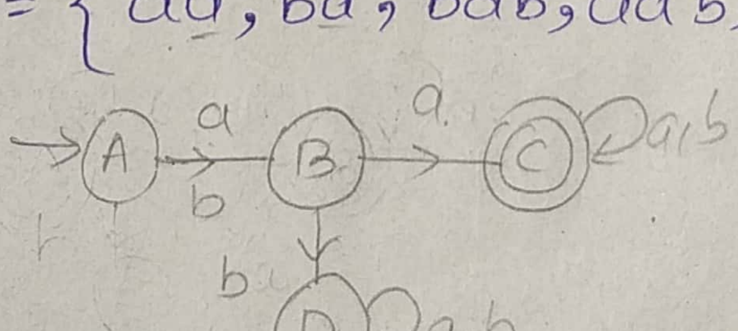
$L = \{a^n b^m c^l / n, m, l \geq 0\}$

$L = \{\epsilon$, aaa, bbb, cccc, abc, ab,
bc, ac, --



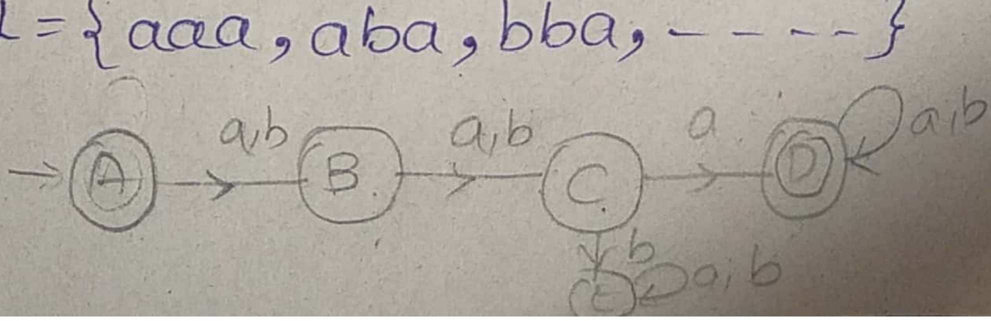
→ construct a minimal DFA which accepts
Set of all strings over {a,b} such that
second symbol from LHS is a

L = { aa, ba, bab, aab, baa - -



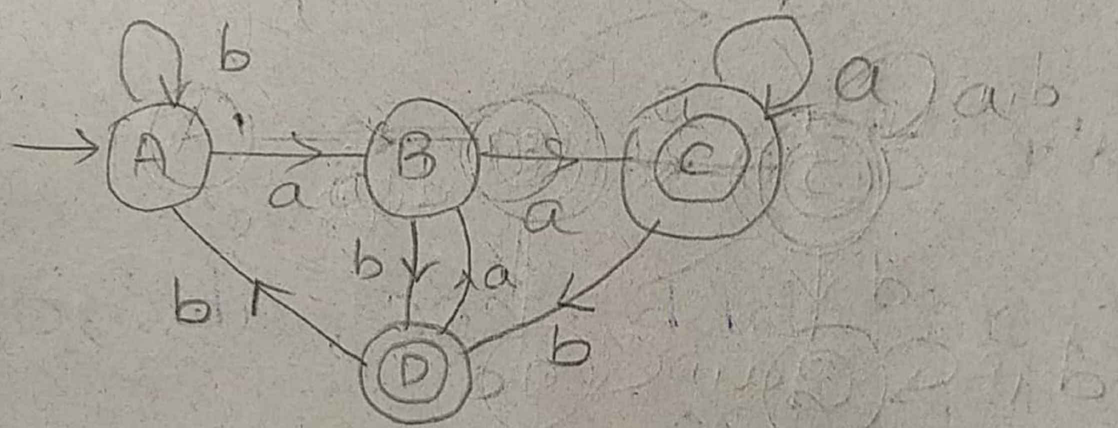
→ third symbol from LHS is 'a'

L = {aaa, aba, bba, - - - -}



→ from RHS second symbol is a

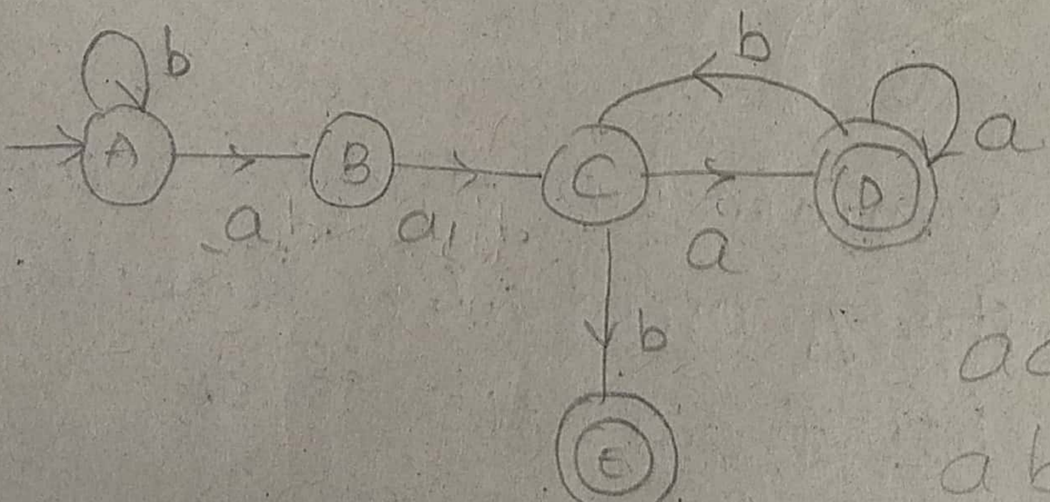
$L = \{aa, ab, aab, bab, ---\}$



→ from RHS third symbol is 'a'

$L = \{aaa, aba, abb, abab, -$

ababababa bbba



aab

abb

28/12/19

NFA or NDFA

→ NFA stands for Non-Deterministic finite Autometa

→ It is easy to construct a NFA than DFA for a given regular language

→ The Finite Autometa are called NFA when there exists many paths for specific input from current state to next state.

→ In NFA By taking an input symbol '0' transitions are available.

→ Every NFA is not DFA but each NFA can be translated into DFA

→ NFA contains a) Multiple next states
b) ϵ transitions

Formal definition of NFA:

→ It contains 5 tupples those are
$\{Q, \Sigma, \delta, q_0, F)$

Q = Set of states

Σ = Input symbols

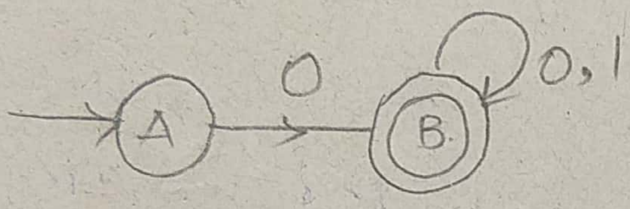
$\boxed{\delta \Rightarrow Q \times \Sigma = 2^Q}$

q_0 = Initial state that must be only one

F = final state

Construct NFA which accepts set of all strings ~~star~~ over $\{0,1\}$ starts with '0'.

$L = \{0, 00, 01, 000, 0001, ---\}$



δ: $\delta(A,0) = B$ $\quad q_0 = \{A\}$

$\delta(A,1) = \phi$ $\quad F = \{B\}$

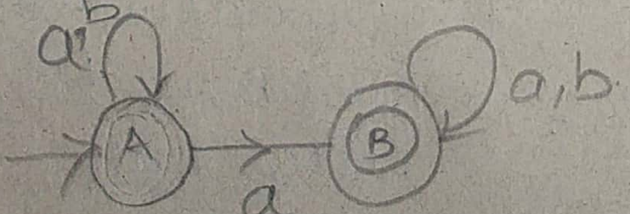
$\delta(B,0) = B$ $\quad Q = \{A, B\}$

$\delta(B,1) = B$

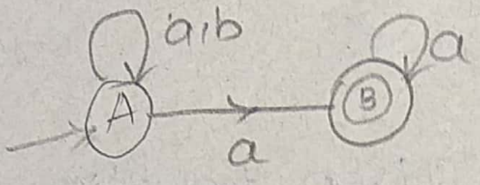
Q \ Σ	0	1
$\rightarrow$ A	B	ϕ
* B	B	B

→ contains 'a'.

$L = \{a, aa, ab, ba, bba, ---\}$

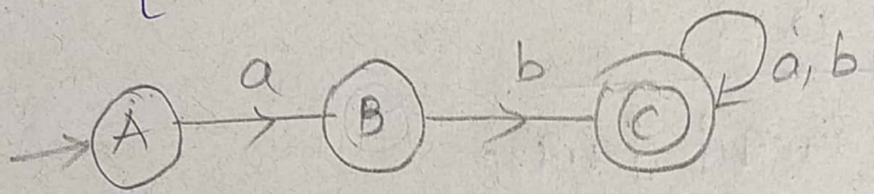


→ ends with 'a'. $L = \{ba, aa, bab, aba, ---\}$



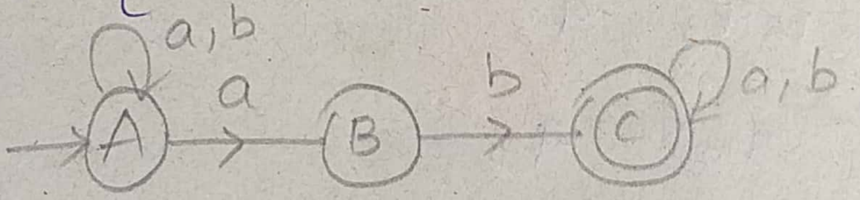
→ starts with 'ab'

$L = \{ab, aba, abab, abb, ---\}$



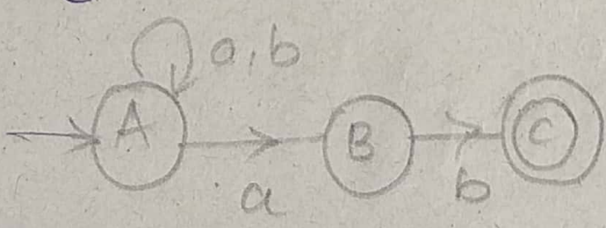
→ contains 'ab'

$L = \{aaba, baba, aaab, ---\}$



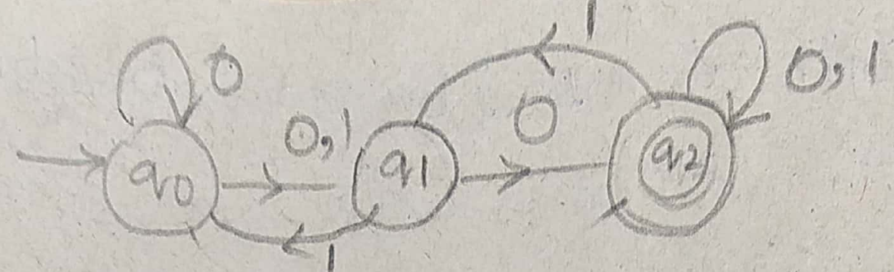
→ ends with 'ab'

$L = \{aab, bab, bbbab, ---\}$



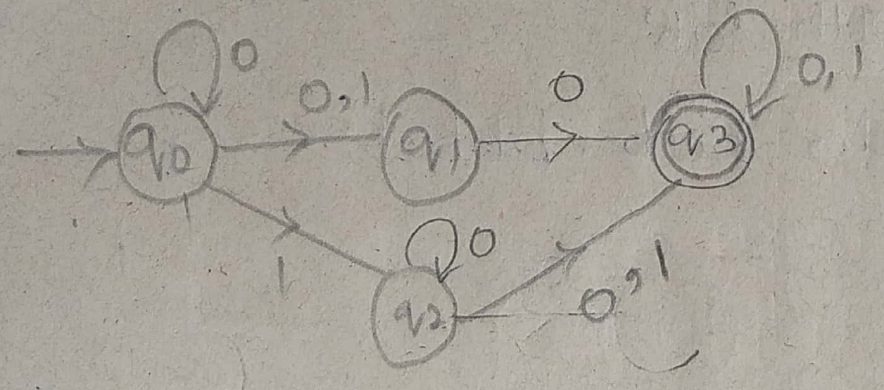
→ construct NFA from given transition table

Q/Σ	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	q_1
q_1	q_2	q_0
$* q_2$	q_2	$\{q_1, q_2\}$



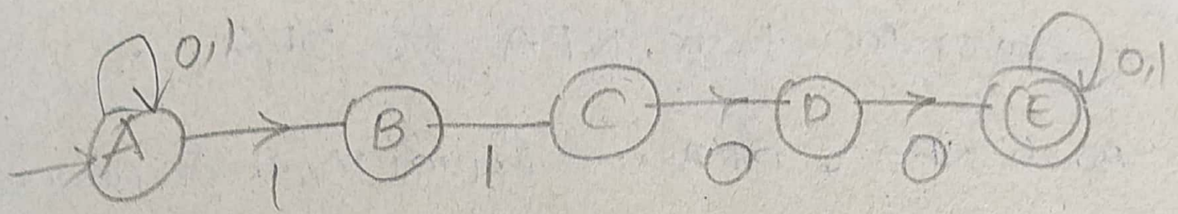
→ design a NFA for given transition table.

Q|Σ	0	1
→ q_0	$\{q_0 q_1\}$	$\{q_1, q_2\}$
q_1	q_3	ϕ
q_2	$q_2 q_3$	q_3
q_3^*	q_3	q_3



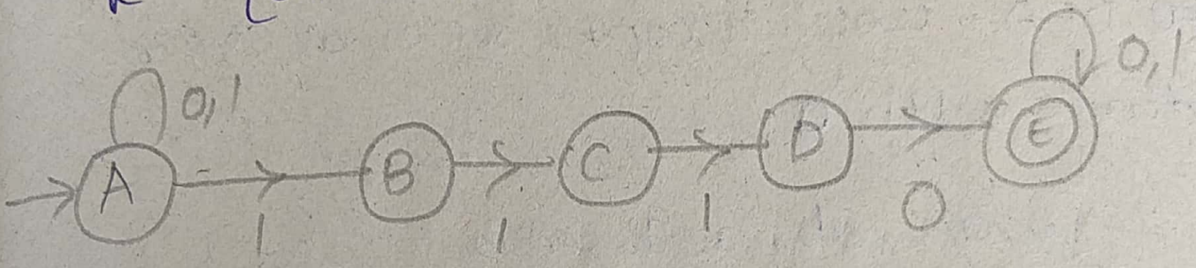
→ design an NFA with $\epsilon = 0, 1$ in which 11 is followed by 00.

$L = \{1100, 01100, 101100, \ldots\}$



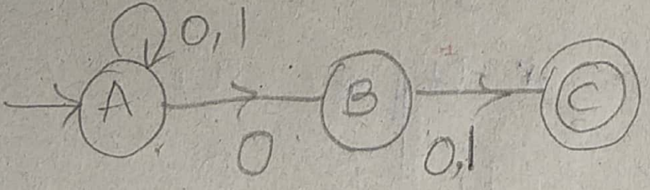
→ design an NFA in which all the strings containing a substrings 1110.

L = {01110, 11100, 00011100, ---}



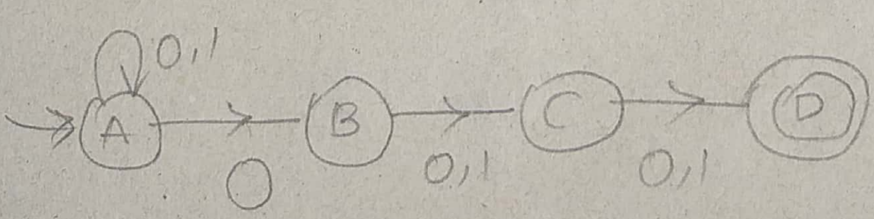
→ design an NFA with {0,1} accepts in which second symbol from RHS is '0'.

L = {01, 000, 001, 101, 1001, 1000, -- }



→ Third symbol from RHS is '0'.

L = {000, 011, 1001, ---}



→ Conversion from NFA to DFA

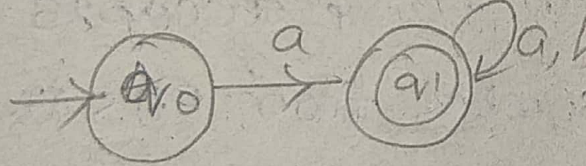
→ Both NFA & DFA are equal in their powers

NFA $\cong$ DFA

→ To convert NFA to DFA we use the subset construction method.

construct NFA and convert the given NFA to DFA. which accepts set of all strings starts with 'a'.

$L = \{a, aa, ab, aab, abb, \dots\}$



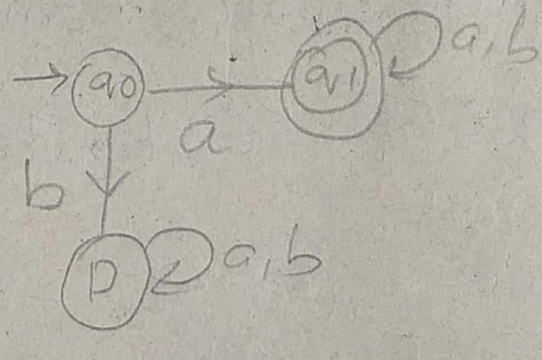
(NFA)

Q/Σ	a	b
$\rightarrow q_0$	q_1	ϕ
q_1^*	q_1	q_2

(NFA transition table)

DFA transition table

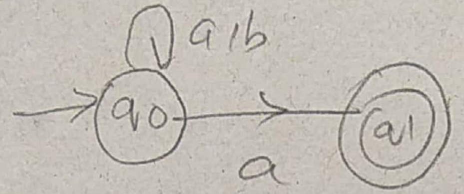
Q/Σ	a	b
$\rightarrow q_0$	q_1	D
$* q_1$	q_1	q_1
D	D	D



(DFA)

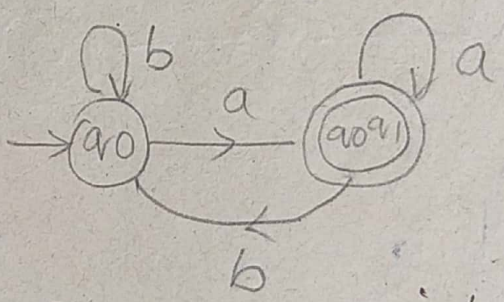
→ ends with 'a'

$L = \{aa, ba, baaa, baba, \ldots\}$



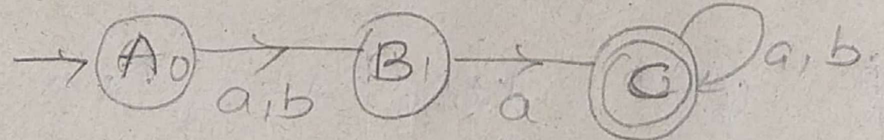
Q|Σ	a	b
→ q0	{q0q1}	q0
q1*	φ	φ

Q|Σ	a	b
→ q0	q0q1	q0
* q0q1	q0q1	q0



→ Second symbol from LHS is 'a'

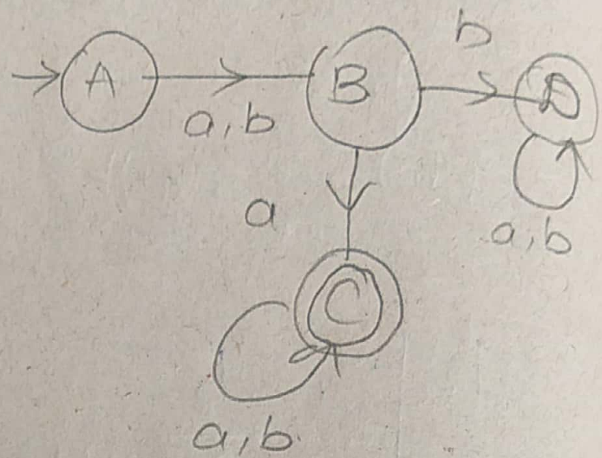
$L = \{aa, ba, baba, \ldots\}$



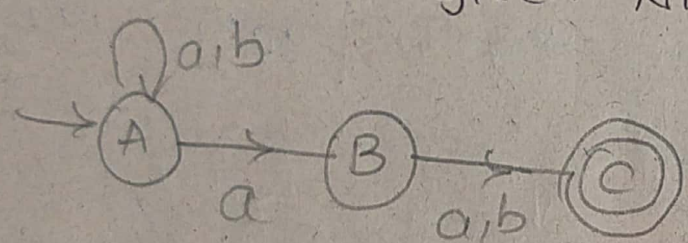
Q|Σ	a	b
→ A	B	B
B	C	φ
*C	C	C

Q|

Q|Σ	a	b
→A	B	B
B	C	D
C*	C	C
D	D	D



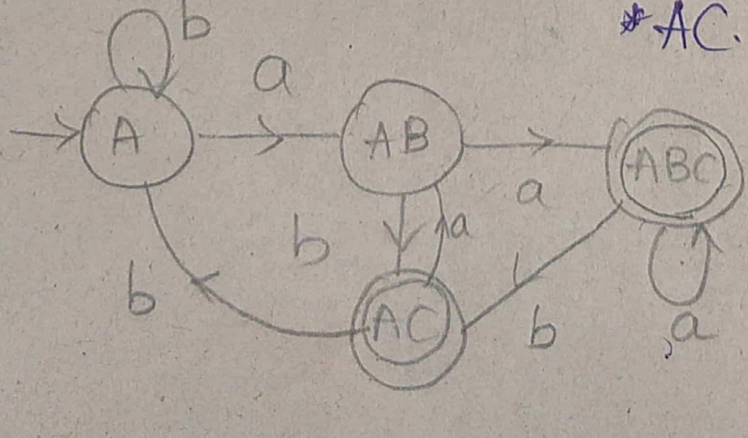
→ convert the given NFA to DFA

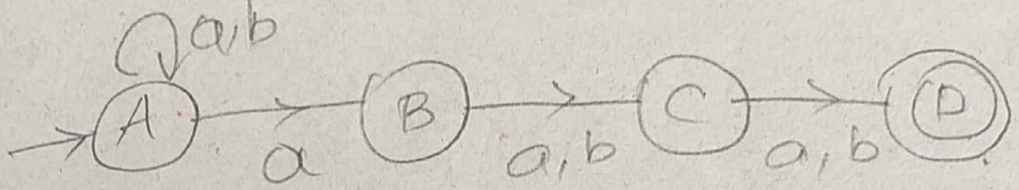


{ Second element from RHS is a }

Q|Σ	a	b
→A	{A,B}	A
B	C	C
C*	φ	φ

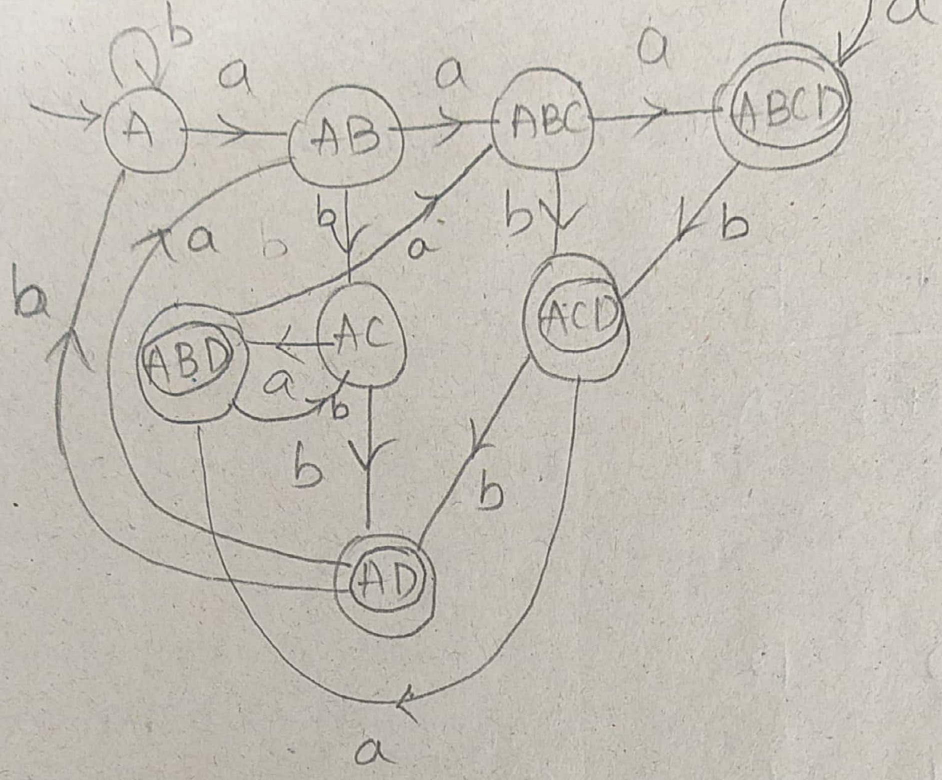
Q|Σ	a	b
→A	AB	A
AB	ABC	AC
*ABC	ABC	AC
*AC	AB	A





Q|Σ	a	b
→ A	{AB}	A
B	C	C
C	D	D
D*	ϕ	ϕ

Q/Σ	a	b
→ A	AB	A
AB	ABC	AC
ABC	ABCD	ACD
AC	ABD	AD
*ABCD	ABCD	ACD
*ACD	ABD	AD
*ABD	ABC	AC
*AD	AB	A



1. Strings having i) length 2 ii) atleast 2

iii) atmost 2

Construct NFA &

Convert NFA to DFA.

2. Find the equivalent DFA for the NFA

$M = [\{A, B, C\}, \{a, b\}, \delta, A, \{C\}]$

where δ is

$Q \mid \Sigma$	a	b
$\rightarrow A$	$\{AB\}$	C
B	A	B
C	ϕ	$\{A, B\}$

(3) construct NFA which accepts set of all string over $\{0,1\}$ that ends with 01 & convert NFA to DFA.

(4) design an NFA for a language that accepts all strings over $\{0,1\}$ in which the second last symbol is always 1 then convert it to its equivalent DFA.

(5) construct a DFA equivalent to

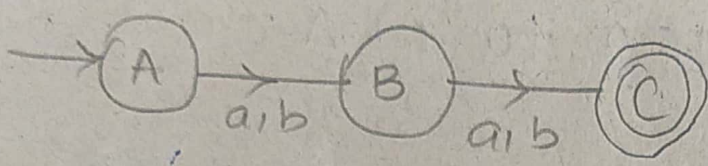
$$M = [\{q_0, q_1, q_2, q_3\}, \{0,1\}, \delta, q_0, \{q_3\}]$$

where δ is

Q/Σ	0	1
$\rightarrow q_0$	$\{q_0 q_1\}$	q_0
q_1	q_2	q_1
q_2	q_3	q_3
q_3	ϕ	q_2

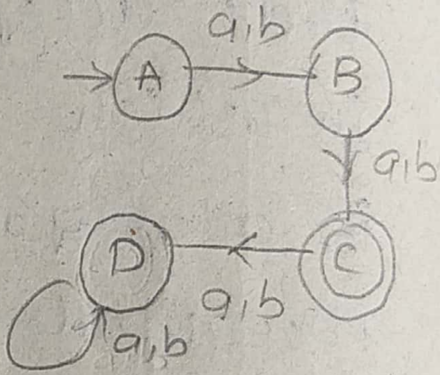
1) i) length 2

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



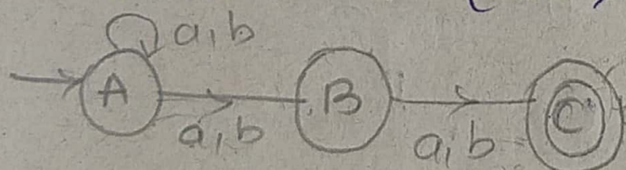
Q|Σ	a	b
→ A	B	B
B	C	C
C*	ϕ	ϕ

Q|Σ	a	b
→ A	B	B
B	C	C
C*	D	D
D	D	D



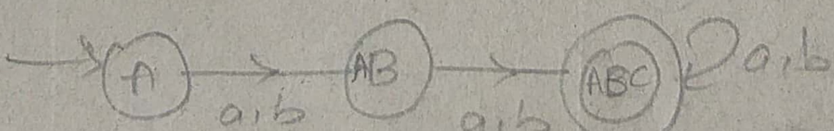
ii) atleast 2

$|W| \geq 2$ $L = \{ab, aa, bb, aba, aab, aaab, \ldots\}$



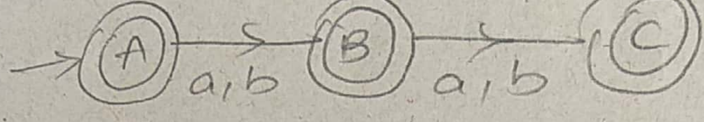
Q|Σ	a	b
→ A	{A,B}	{A,B}
B	C	C
C*	ϕ	ϕ

Q|Σ	a	b
→ A	AB	AB
AB	ABC	ABC
ABC*	ABC	ABC



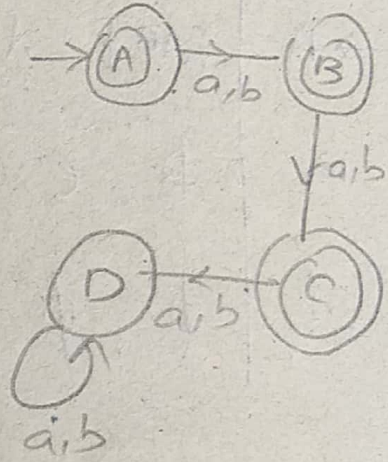
iii) atmost 2

$|w| \le 2$ $L = \{\epsilon, a, b, aa, bb, ab, ba\}$



Q|Σ	a	b
→A*	B	B
B*	C	C
C*	φ	φ

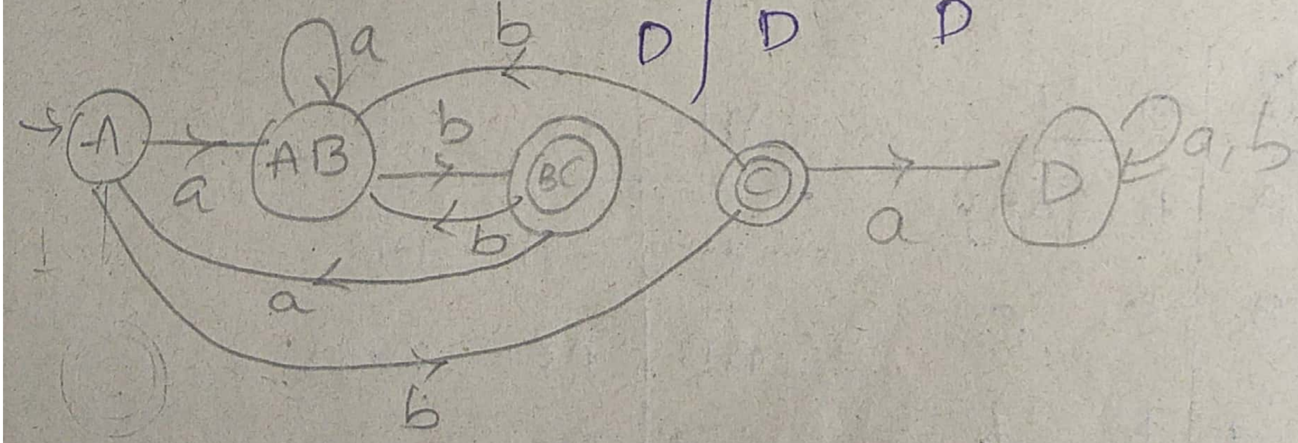
Q|Σ	a	b
→A*	B	B
B*	C	C
C*	D	D
D	D	D



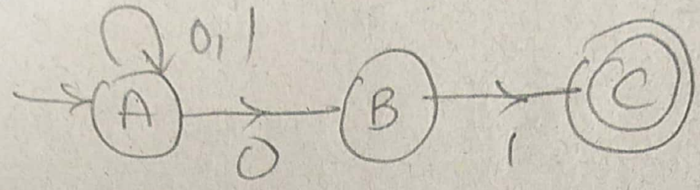
2

Q|Σ	a	b
→A	{AB}	C
B	A	B
©	φ	{AB}

Q|Σ	a	b
→A	AB	C
AB	AB	BC
BC*	A	AB
C*	D	AB
D	D	D

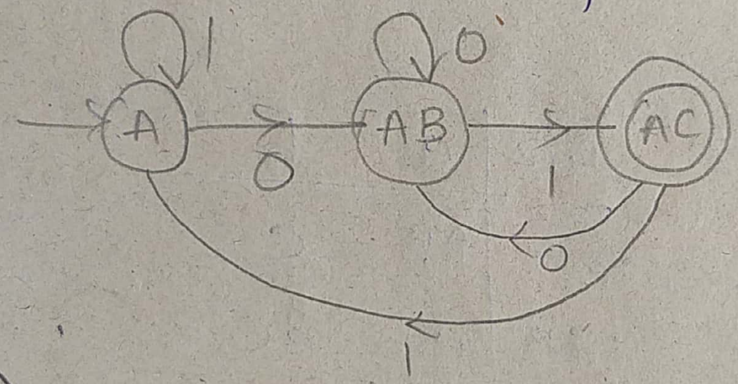


(3) $L = \{01, 001, 101, 0101, --\}$

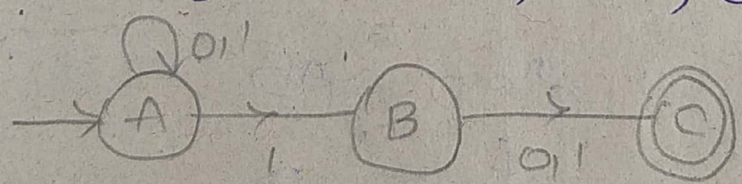


$Q \mid \Sigma$	0	1
$\rightarrow A$	{AB}	A
B	ϕ	C
C^*	ϕ	ϕ

$Q \mid \Sigma$	0	1
$\rightarrow A$	AB	A
AB	AB	AC
AC^*	AB	A

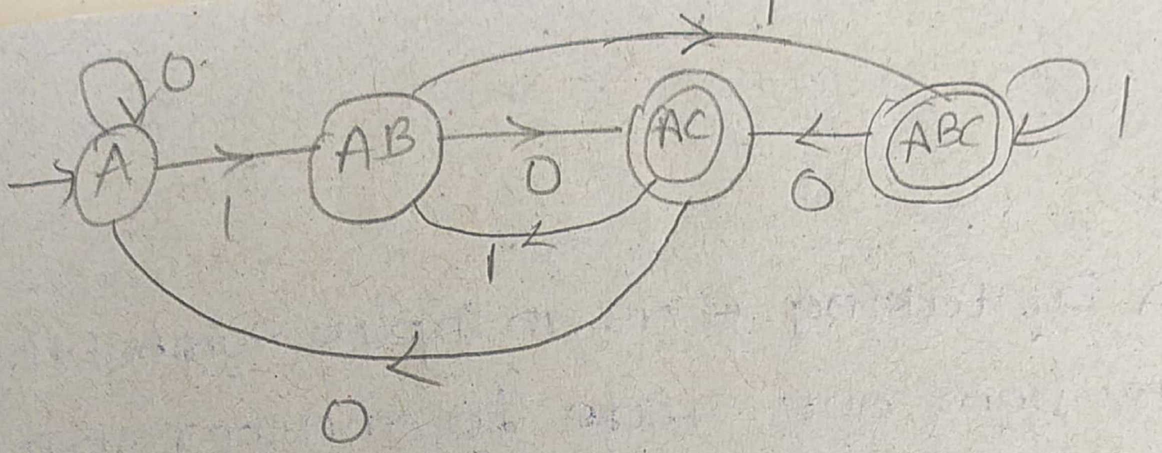


(4) $L = \{10, 11, 010, 011, 0010, --\}$



$Q \mid \Sigma$	0	1
$\rightarrow A$	A	{AB}
B	C	C
C^*	ϕ	ϕ

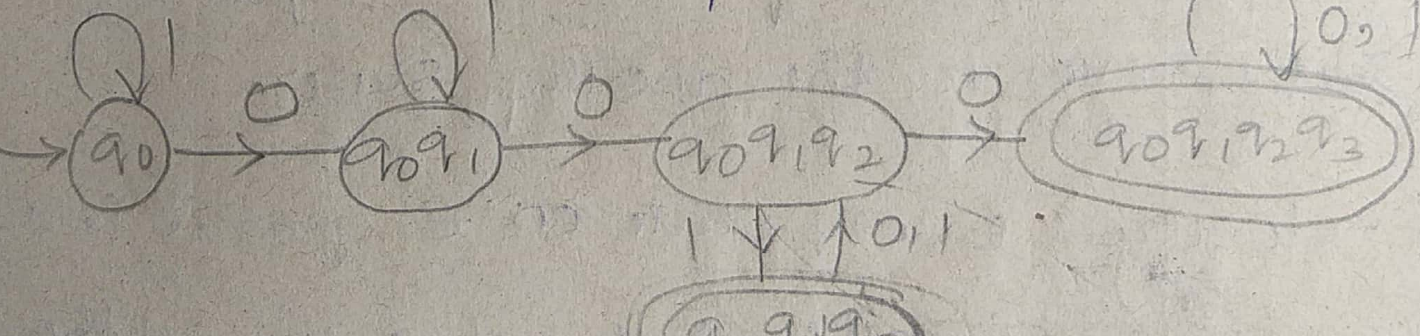
$Q \mid \Sigma$	0	1
$\rightarrow A$	A	AB
AB	AC	ABC
AC^*	A	AB
ABC^*	AC	ABC



(5)

Q/Σ	0	1
→ q_0	$\{q_0 q_1\}$	q_0
q_1	q_2	q_1
q_2	q_3	q_3
q_3	ϕ	q_2

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



31/12/19

ε-NFA:

In ε-NFA By taking ε as an input symbol it will move from one state to another state

Formal definition of ε-NFA:

It contains 5 tupples those are

$(Q, \Sigma, \delta, q_0, F)$.

Q, Σ, q_0, F those are same for DFA, NFA & ε-NFA only difference is transition function i.e $\boxed{\delta : Q \times \Sigma \cup \epsilon \rightarrow 2^Q}$

ε-move: By taking ε as an input one states moves to the another state

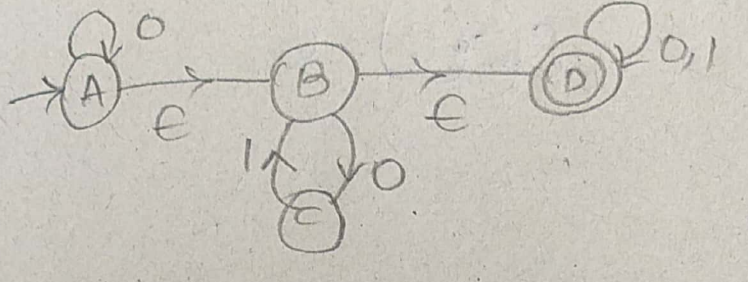
ε-closure: Every state on ε goes to itself (ϵ^*)

→ ε-closure is nothing but what are all the states reach on seeing ε.

→ DFA, NFA, ε-NFA are very powerful we are able to convert DFA to NFA, NFA to DFA, ε-NFA to NFA & ε-NFA to DFA

→ ~~Ever~~ Every state is ε-closure to itself.

conversion from ε-NFA to NFA.



$\{\epsilon\text{-closure}[\delta(\epsilon\text{-closure}), 0]\}$.

A

ϵ^*	0	ϵ^*
A	$\xrightarrow{0}$ A	A, B, D
B	$\xrightarrow{0}$ C	C
D	$\xrightarrow{0}$ D	D

$\delta(A,0) = \{A,B,C,D\}$.

A

ϵ^*	1	ϵ^*
A	$\xrightarrow{1} \phi$	
B	$\xrightarrow{1} \phi$	
D	$\xrightarrow{1}$ D	D

$\delta(A,1) = \{D\}$

B

ϵ^*	0	ϵ^*
B	$\xrightarrow{0}$ C	C
D	$\xrightarrow{0}$ D	D

$\delta(B,0) = \{C,D\}$

B

ϵ^*	1	ϵ^*
B	$\xrightarrow{1} \phi$	
D	$\xrightarrow{1}$ D	D

$\delta(B,1) = \{D\}$

Scanned with CamScanner

C $\quad \epsilon^* \quad 0 \quad \epsilon^*$

$C \xrightarrow{0} \phi \quad \phi$ $\qquad \delta(C,0) = \phi$

C $\quad \epsilon^* \quad 1 \quad \epsilon^*$

$C \xrightarrow{1} B \to B, D$ $\qquad \delta(C,1) = \{B, D\}$

D $\quad \epsilon^* \quad 0 \quad \epsilon^*$

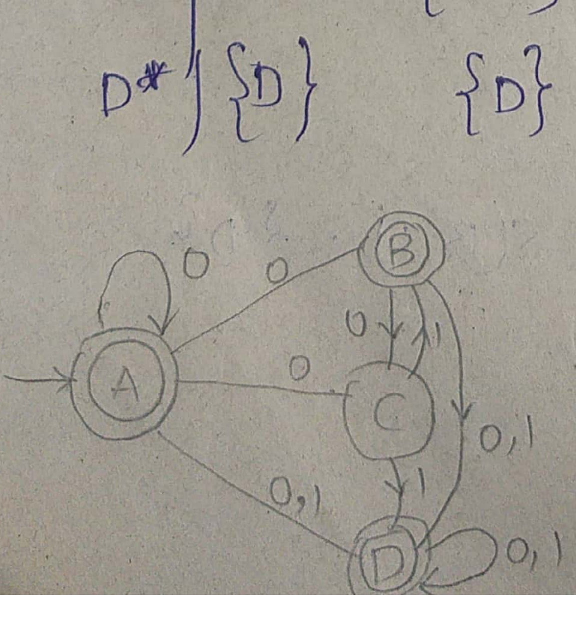
$D \xrightarrow{0} D \to D$ $\qquad \delta(D,0) = \{D\}$

D $\quad \epsilon^* \quad 1 \quad \epsilon^*$

$D \xrightarrow{1} D \to D$ $\qquad \delta(D,1) = \{D\}$

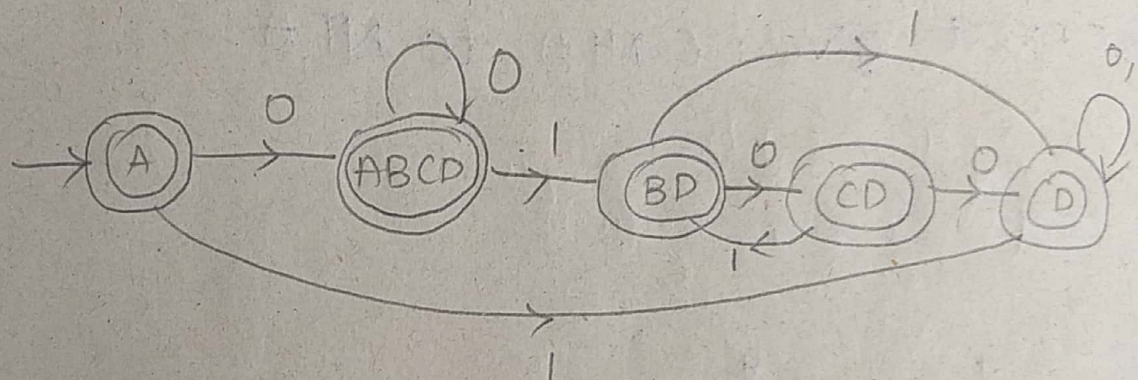
Q|Σ	0	1
→ A*	{ABCD}	{D}
B*	{CD}	{D}
C	φ	{BD}
D*	{D}	{D}

(NFA) from (ε-NFA)



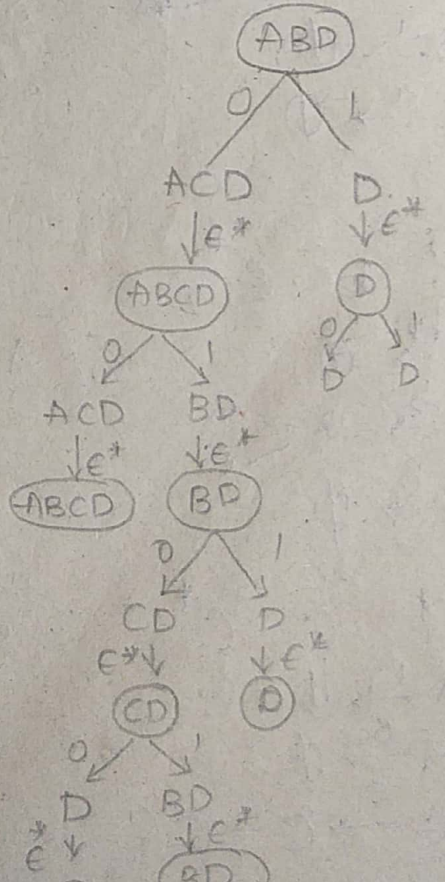
Q/Σ	0	1
→A*	ABCD	D
ABCD*	ABCD	BD
BD*	CD	D
CD*	D	BD
D*	D	D

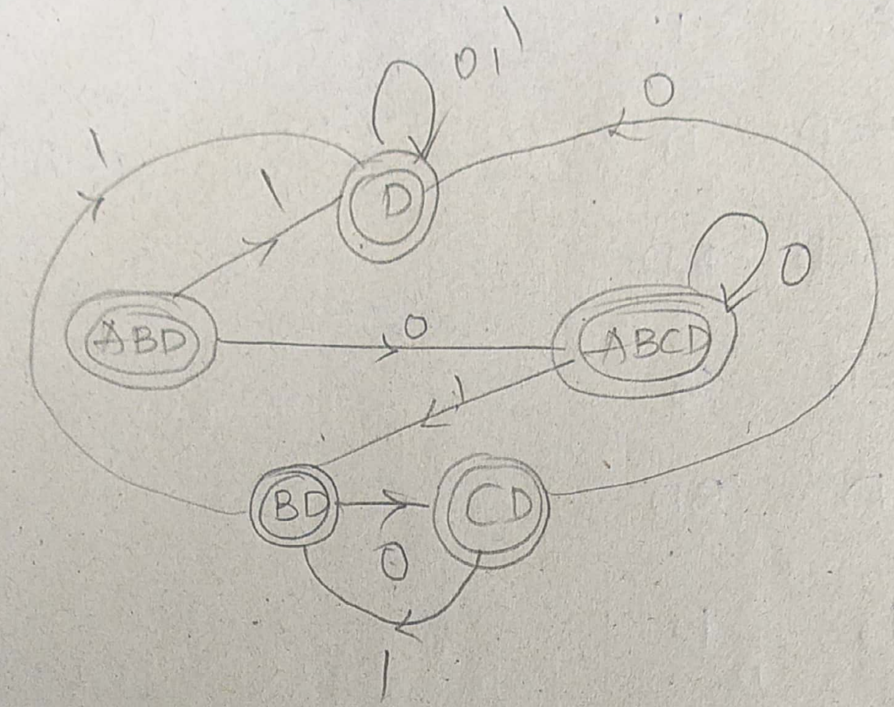
(DFA from NFA)



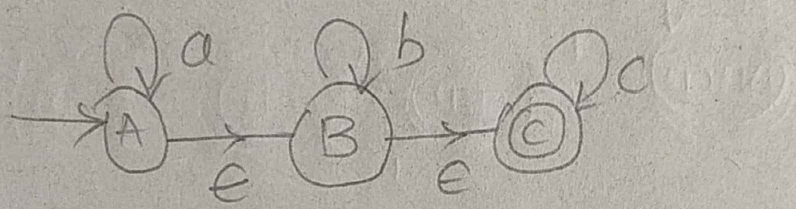
Indirect ε-NFA to DFA

	0	1	ε*
→A	A	ϕ	A,B,D
B	C	ϕ	B,D
C.	ϕ	B.	C
D*	D	D.	D.





Convert given ε-NFA to NFA



A) ϵ^* $\quad a \quad$ ϵ^*

$A \xrightarrow{a} A <$ A, B, C

$B \xrightarrow{a} \phi$

$C \xrightarrow{a} \phi$

$\delta(A,a) = \{ABC\}$

A) ϵ^* $\quad b \quad$ ϵ^*

$A \xrightarrow{b} \phi$

$B \xrightarrow{b} B <$ B, C

$C \xrightarrow{b} \phi$

$\delta(A,b) = \{BC\}$

A) ϵ^* $\quad c \quad$ ϵ^*

$\delta(A,c) = \{C\}$

$A \xrightarrow{c} \phi$

$B \xrightarrow{c} \phi$

$C \xrightarrow{c} C - C$

B) $\epsilon^* \quad a \quad \epsilon^*$

$B \xrightarrow{a} \phi$

$C \xrightarrow{a} \phi$

$\delta(B, a) = \phi$

B) $\epsilon^* \quad b \quad \epsilon^*$

$B \xrightarrow{b} B < \begin{matrix} B \\ C \end{matrix}$

$C \xrightarrow{b} \phi$

$\delta(B, b) = \{B C\}$

B) $\epsilon^* \quad c \quad \epsilon^*$

$B \xrightarrow{c} \phi$

$C \xrightarrow{c} C - C$

$\delta(B, c) = \{C\}$

C) $\epsilon^* \quad a \quad \epsilon^*$

$C \xrightarrow{a} \phi$

$\delta(C, a) = \phi$

C) $\epsilon^* \quad b \quad \epsilon^*$

$C \xrightarrow{b} \phi$

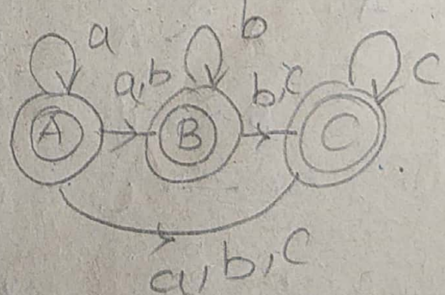
$\delta(C, b) = \phi$

C) $\epsilon^* \quad c \quad \epsilon^*$

$C \xrightarrow{c} C \rightarrow C$

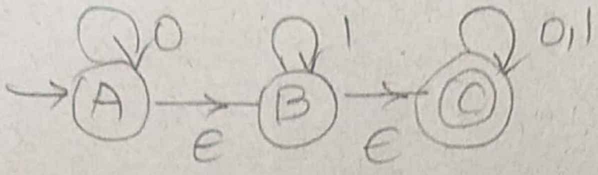
$\delta(C, c) = \{C\}$

	a	b	c
→A*	{A,B,C}	{B,C}	C
B*	ϕ	{B,C}	C
C*	ϕ	ϕ	C

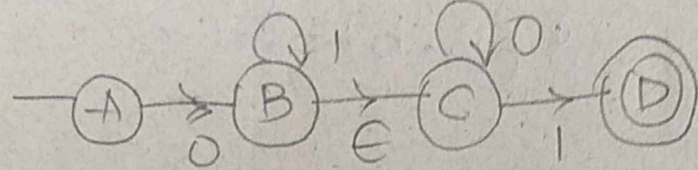


→ convert ε-NFA to NFA & ε-NFA - DFA

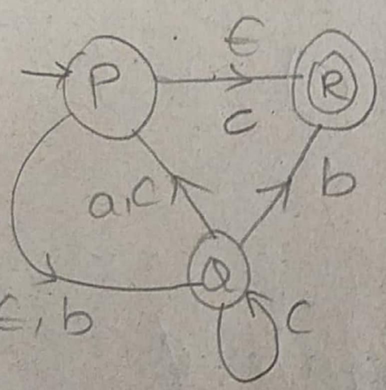
1)



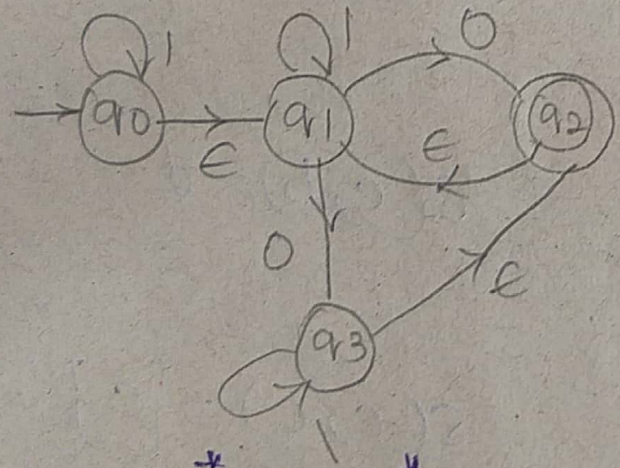
2)



3)



4)



1) A. ϵ^* 0 ϵ^*

$A \xrightarrow{0} A$ — A, B, C

$B \xrightarrow{0} \phi$

$C \xrightarrow{0} C$ — C

$\delta(A,0) = \{A,B,C\}$

ϵ^* 1 ϵ^*

A $A \xrightarrow{1} \phi$

$B \xrightarrow{1} B$ — B, C

$C \xrightarrow{1} C$ — C

$\delta(A,1) = \{B,C\}$

$$\epsilon^* \quad 0 \quad \epsilon^*$$

B $\quad B \xrightarrow{0} \phi$

$\quad C \xrightarrow{0} C - C$

$\delta(B,0) = \{C\}$

$$\epsilon^* \quad 1 \quad \epsilon^*$$

B $\quad B \xrightarrow{1} B - B, C$

$\quad C \xrightarrow{1} C - C$

$\delta(B,1) = \{B, C\}$

$$\epsilon^* \quad 0 \quad \epsilon^*$$

C $\quad C \xrightarrow{0} C - C$

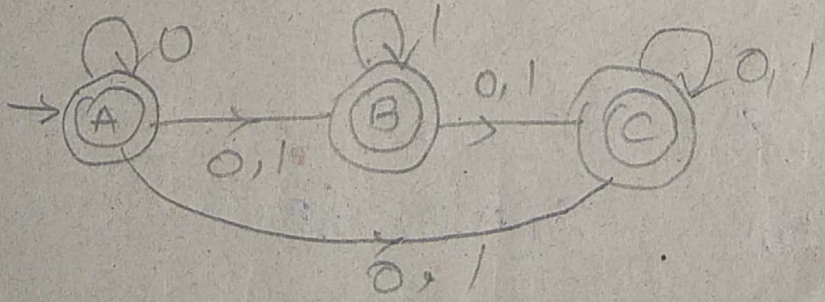
$\delta(C,0) = \{C\}$

$$\epsilon^* \quad 1 \quad \epsilon^*$$

C $\quad C \xrightarrow{1} C - C$

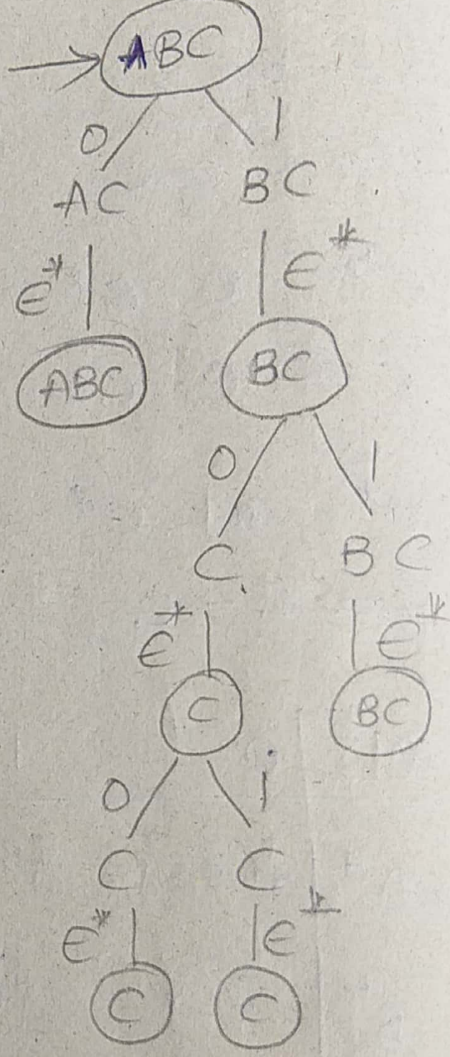
$\delta(C,1) = \{C\}$

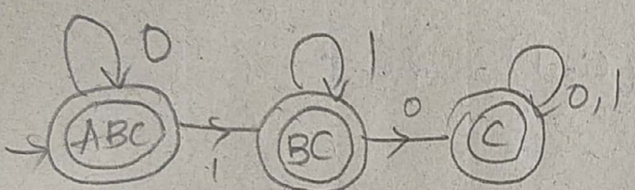
Q|Σ	0	1
→A*	{A,B,C}	{B,C}
B*	{C}	{B,C}
C*	{C}	{C}



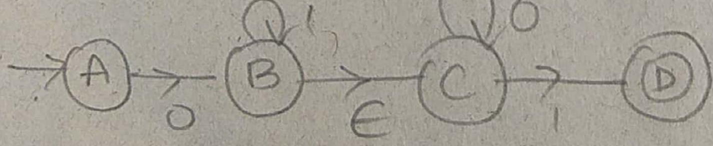
ϵ-NFA to DFA

	0	1	ϵ^*
$\rightarrow A$	A	ϕ	A, B, C
B	ϕ	B	B, C
C^*	C	C	C





2) ϵ-NFA to NFA



ϵ^* 0 ϵ^*

A $\quad A \xrightarrow{0} B - B, C \qquad \delta(A, 0) = \{B, C\}$

ϵ^* 1 ϵ^*

A $\quad A \xrightarrow{1} \phi \quad \phi \qquad \delta(A, 1) = \phi$

$$\epsilon^* \quad 0 \quad \epsilon^*$$

B $\quad B \xrightarrow{0} \phi$

$C \xrightarrow{0} C - C$ $\qquad \delta(B,0) = \{C\}$

$$\epsilon^* \quad 1 \quad \epsilon^*$$

B $\quad B \xrightarrow{1} B - B, C$ $\qquad \delta(B,1) = \{B, C, D\}$

$C \xrightarrow{1} D - D$

$$\epsilon^* \quad 0 \quad \epsilon^*$$

C $\quad C \xrightarrow{0} C - C$ $\qquad \delta(C,0) = \{C\}$

$$\epsilon^* \quad 1 \quad \epsilon^*$$

C $\quad C \xrightarrow{1} D - D$ $\qquad \delta(C,1) = \{D\}$

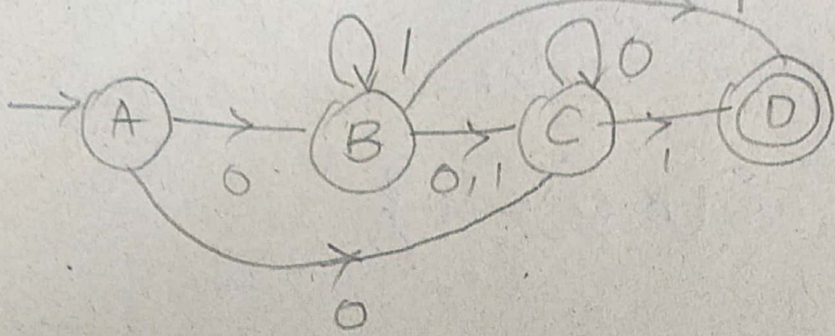
$$\epsilon^* \quad 0 \quad \epsilon^*$$

D $\quad D \xrightarrow{0} \phi \quad \phi$ $\qquad \delta(D,0) = \phi$

$$\epsilon^* \quad 1 \quad \epsilon^*$$

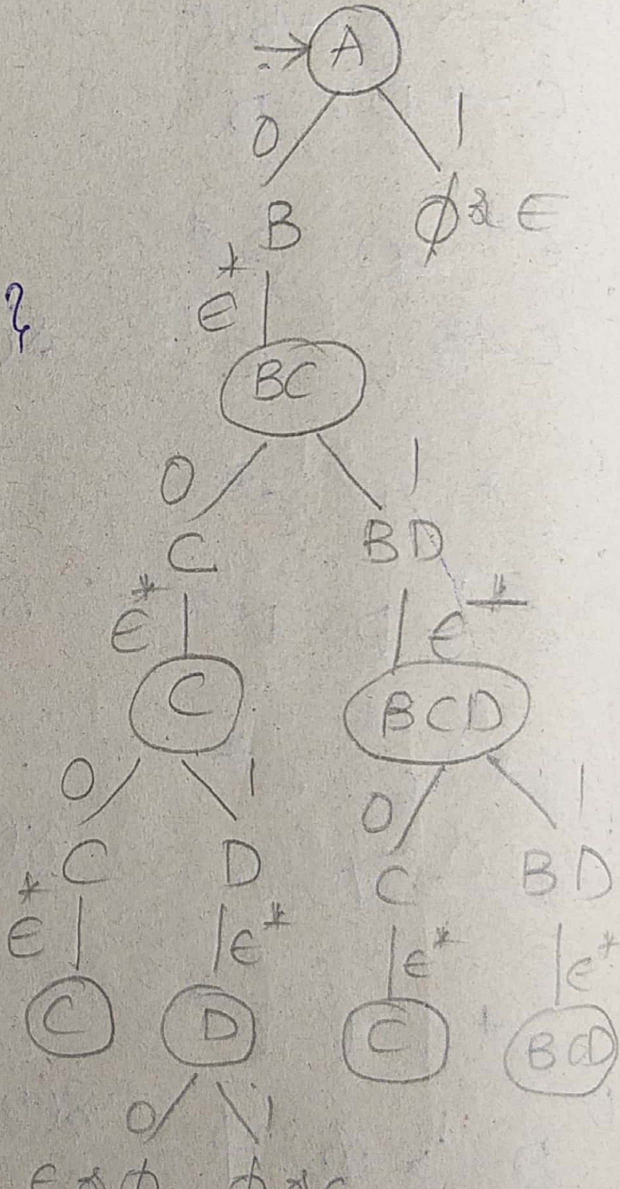
D $\quad D \xrightarrow{1} \phi \quad \phi$ $\qquad \delta(D,1) = \phi$

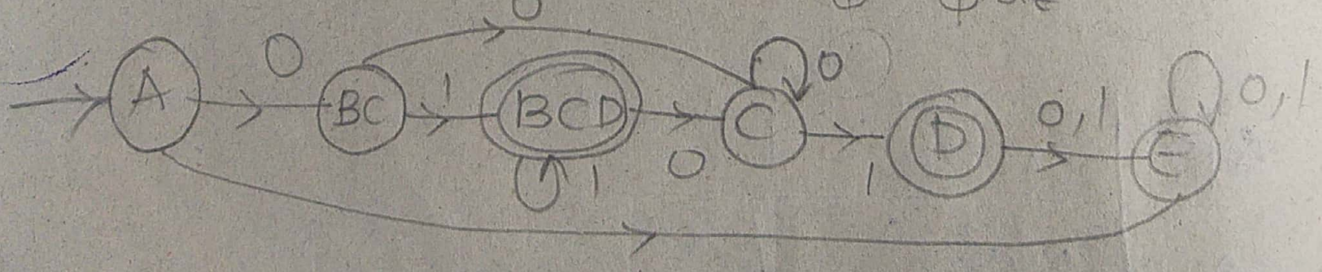
Q|Σ	0	1
→ A	{B,C}	ϕ
B	{C}	{B,C,D}
C	{C}	{D}
D*	ϕ	ϕ



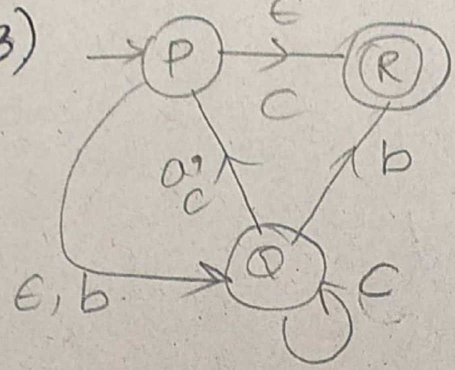
ε-NFA to DFA

	0	1	ϵ^*
→ A	B	ϕ	A
B	ε ϕ	B	{B, C}
C	C	D	C
D^*	ϕ ε	ϕ ε	D
			E
E	E	E	





3)



P $\quad \epsilon^* \quad a \quad \epsilon^*$

$P \xrightarrow{a} \phi$

$Q \xrightarrow{a} P - P, Q, R$

$R \xrightarrow[a]{} \phi$

$\delta(P, a) = \{P, Q, R\}$

P $\quad \epsilon^* \quad b \quad \epsilon^*$

$P \xrightarrow{b} Q - Q$

$Q \xrightarrow{b} R - R$

$R \xrightarrow{b} \phi$

$\delta(P, b) = \{Q, R\}$

P $\quad \epsilon^* \quad c \quad \epsilon^*$

$P \xrightarrow{c} R - R$

$Q \xrightarrow{c} P, Q - PQR$

$R \xrightarrow{c} \phi$

$\delta(P, c) = \{P, Q, R\}$

Q $\quad \epsilon^* \quad a \quad \epsilon^*$

$Q \xrightarrow{a} P - P, Q, R$

$\delta(Q, a) = \{P, Q, R\}$

Q $\quad \epsilon^* \quad b \quad \epsilon^*$

$Q \xrightarrow{b} R - R$

$\delta(Q, b) = \{R\}$

Q $\quad \epsilon^* \quad c \quad \epsilon^*$

$Q \xrightarrow{c} P, Q - P, Q, R$

$\delta(Q, c) = \{P, Q, R\}$

R ϵ^* a ϵ^*

$R \xrightarrow{a} \phi$ $\delta(R,a) = \phi$

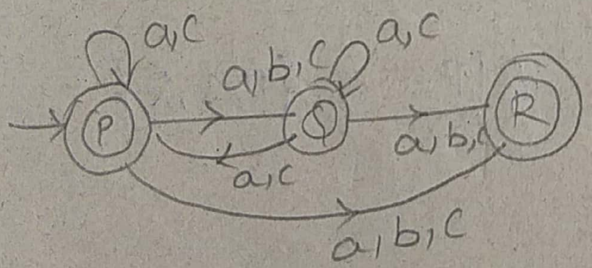
R ϵ^* b ϵ^*

$R \xrightarrow{b} \phi$ $\delta(R,b) = \phi$

R ϵ^* c ϵ^*

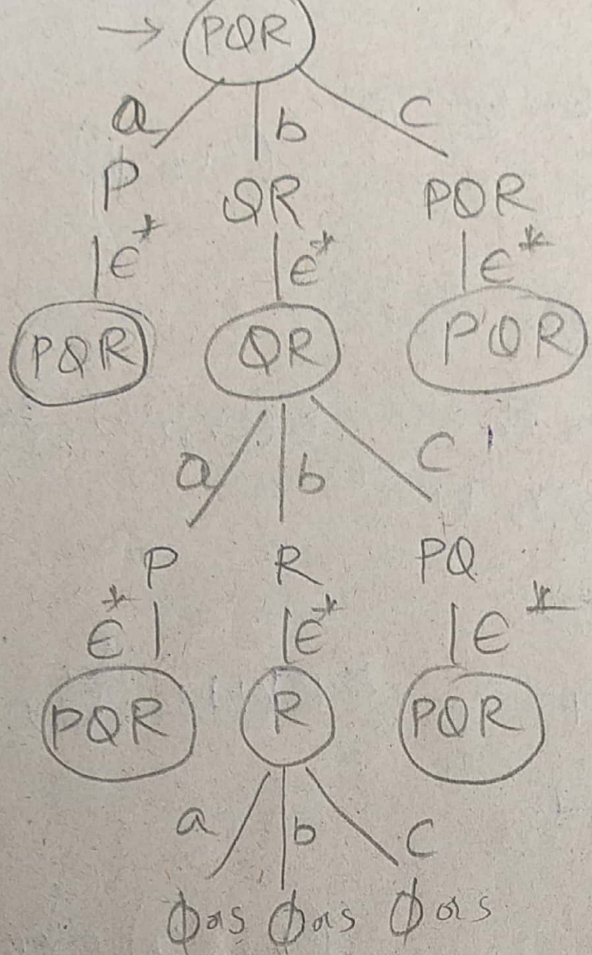
$R \xrightarrow{c} \phi$ $\delta(R,c) = \phi$

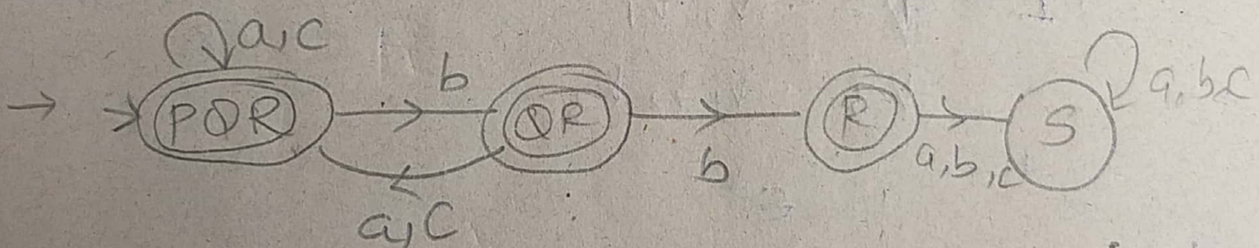
$Q \mid \Sigma$	a	b	c
$\rightarrow P^*$	{P,Q,R}	{Q,R}	{P,Q,R}
Q	{P,Q,R}	{R}	{P,Q,R}
R^*	ϕ	ϕ	ϕ



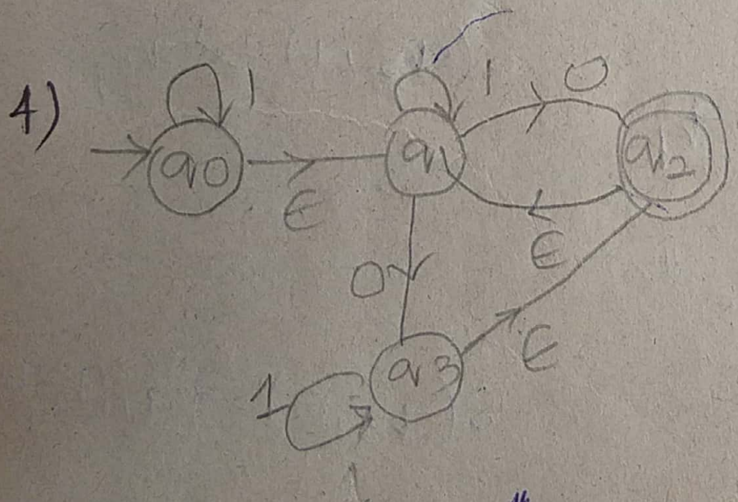
ϵ-NFA to DFA

	a	b	c	ϵ^*
$\rightarrow P$	ϕ	Q	R	P, Q, R
Q	P	R	{P,Q}	Q
R^*	ϕ	ϕ	ϕ	R





4)



ϵ^* 0 ϵ^*

q_0 $q_0 \xrightarrow{0} \phi$

$q_1 \xrightarrow{0} q_2$ - $q_1 q_2$

q_3 - $q_3, q_2 q_1$

$\delta(q_0, 0) = \{q_1 q_2 q_3\}$

Scanned with CamScanner

$\epsilon^* \quad 1 \quad \epsilon^*$

$q_0 \quad q_0 \xrightarrow{1} q_0 - q_0, q_1$ $\qquad \delta(q_0, 1) = \{q_0, q_1\}$

$q_1 \xrightarrow{1} q_1 - q_1$

$\epsilon^* \quad 0 \quad \epsilon^*$

$q_1 \quad q_1 \xrightarrow{0} q_2 - q_2, q_1$ $\qquad \delta(q_1, 0) = \{q_1, q_2, q_3\}$

$q_3 - q_3 q_2 q_1$

$\epsilon^* \quad 1 \quad \epsilon^*$

$q_1 \quad q_1 \xrightarrow{1} q_1 - q_1$ $\qquad \delta(q_1, 1) = \{q_1\}$

$\epsilon^* \quad 0 \quad \epsilon^*$

$q_2 \quad q_2 \xrightarrow{0} \phi$ $\qquad \delta(q_2, 0) = \{q_1, q_2, q_3\}$

$q_1 \xrightarrow{0} q_2 - q_2 q_1$

$q_3 - q_1 q_2 q_3$

$\epsilon^* \quad 1 \quad \epsilon^*$

$q_2 \quad q_2 \xrightarrow{1} \phi$ $\qquad \delta(q_2, 1) = \{q_1\}$

$q_1 \xrightarrow{1} q_1 - q_1$

$\epsilon^* \quad 0 \quad \epsilon^*$

$q_3 \quad q_3 \xrightarrow{0} \phi$ $\qquad \delta(q_3, 0) = \{q_1, q_2, q_3\}$

$q_2 \xrightarrow{0} \phi$

$q_1 \xrightarrow{0} q_2 - q_2 q_1$

$q_3 - q_3 q_2 q_1$

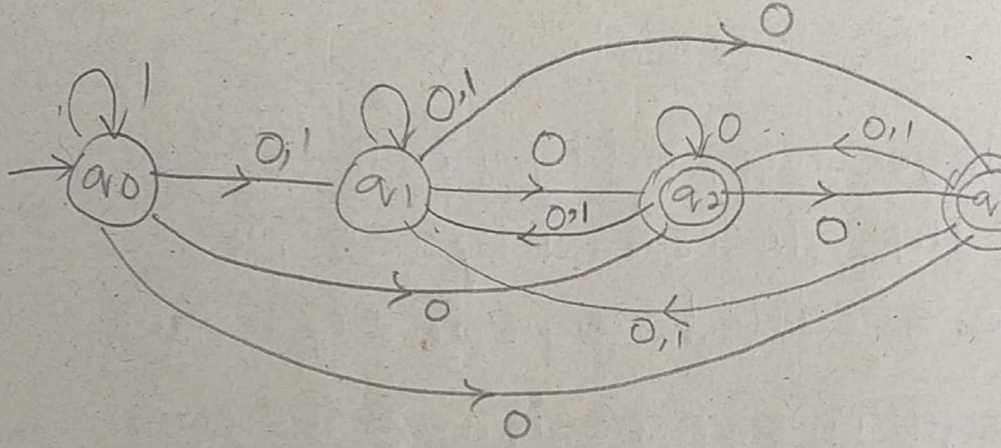
$\epsilon^* \quad 1 \quad \epsilon^*$

$q_3 \quad q_3 \xrightarrow{1} q_3 - q_1 q_2 q_3$ $\qquad \delta(q_3, 1) = \{q_1 q_2 q_3\}$

$q_2 \xrightarrow{1} \phi$

$q_1 \xrightarrow{1} q_1 - q_1$

Q\Σ	0	1
→ q_0	$\{q_1, q_2, q_3\}$	$\{q_0, q_1\}$
q_1	$\{q_1 q_2 q_3\}$	$\{q_1\}$
q_2^*	$\{q_1 q_2 q_3\}$	$\{q_1\}$
q_3^*	$\{q_1 q_2 q_3\}$	$\{q_1 q_2 q_3\}$



ε-NFA to DFA

	0	1.	ε*
→ q_0	φ	q_0	$\{q_0, q_1\}$
q_1	$\{q_2 q_3\}$	q_1	q_1
q_2	φ	φ	q_2, q_1
q_3	φ	q_3	$q_1 q_2 q_3$