MODULE - 1

TERMINOLOGIES

Alphabet

* An alphabet is a finite
* non - Empty Symbols
* Represented by Σ

eg :- $\Sigma = \{0, 1\}$

$\Sigma = \{a, b, c \cdots, z\}$

Strings.

* finite Sequence of Symbols

eg :- 01011O, 111 } are the strings of the $\Sigma = \{0, 1\}$

Empty String :-

* no Symbol String

denoted by ϵ

Empty String :- strings with zero operands on Symbols

length of a String

denoted by $|w|$

$w = 010$

$|w| = 3.$

length of an Empty String

$|\epsilon| = 0$

$\Sigma^k =$ Set of String of length A

$k = 2.$

$\Sigma^k = \{01, 10, 00, 11\}$

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

$\Sigma^1 = \{a, b, c\}$

$\Sigma^2 = \{aa, ab, ac, bb, ba, bb, bc, ca, cb, cc\}$

Σ^* [Kleen closure of Σ]

Set of all Strings over an alphabet Σ

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

$\Sigma^* = \{\epsilon, 0, 1, 01, 00, 10, 11, 010, 000, 111, 100, \ldots\}$

OR

$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \ldots\ldots$

$\Sigma^+ \Rightarrow$ Set of non Empty Strings of alphabet Σ

$\rightarrow$ (Positive closure of Σ)

$\Sigma^+ = \{0, 1, 01, 10, 11, 10, 000 \ldots\}$

$\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \Sigma^4 \ldots$

$\Sigma^+ = \Sigma^* - \{\epsilon\}$

Contamination of Strings

* Let $x = 1011$ and $y = 0110$ are two Strings

xy denotes the contamination of x & y

$xy = 10110110$

$yx = 01101011$

* Language:

- Set of strings of all of which are chosen from some Σ^*

- Language is the subset of Σ^*

  If $L \subseteq \Sigma^*$, L is a language over Σ.

Eg:- Set of all strings consisting of n 0's followed by n 1's for some $n \geq 0$

$n=0, n=1, n=2, n=3,$

$L = \{\epsilon, 01, 0011, 000111, \ldots\}$

- Set of strings 0's and 1's with an equal number of each

$L = \{\epsilon, 01, 10, 0011, 0101, 1100, \ldots\}$

- Set of binary no's whose value is a prime.

$L = \{001, 10$

$L = \{10, 11, 101, 111, 1011, \ldots \}$

* ϕ - Empty language

$\{\epsilon\}$ - language consist of only the Empty String.

$\{\epsilon\}$ - empty string not Empty language

* finite automata :-

- Deterministic finite automata (DFA)
- Non-deterministic finite automata (NFA)

Deterministic finite automata (DFA)

Formally defined with 5 topples $(Q, \Sigma, \delta, q_0, F)$.

$Q \Rightarrow$ finite set of state

$\Sigma \Rightarrow$ finite set of alphabet input symbols.

$\delta \Rightarrow$ transition function that takes as arguments a state and an input symbols and returns a state

$\delta = Q \times \Sigma \Rightarrow Q.$

$q_0 \Rightarrow$ Start state

$q_0 \in Q$.

$F \Rightarrow$ Set of final or accepting states

$F \subseteq Q$ $F \subset Q$.

∘ F is a subset of Q.

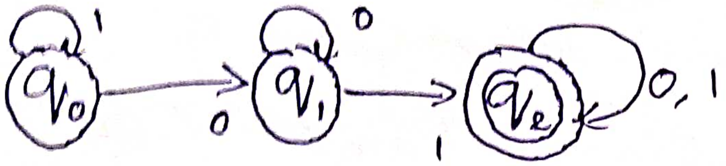
Transition diagram

States - Nodes

$\delta(p, a) = q$ — Arc

Start state $\rightarrow$ q_0

Final state $\rightarrow$ q_0



$\{0, 1\}$

Each transition for Each Symbol exactly one.

DFA → has restriction
NFA → does not have transition

transition Table

row → States
colomn → input Symbols.

$\delta(q_0, 0)$
$\delta(q_0, 1)$
$\delta(q_1, 0)$
$\delta(q_1, 1)$
$\delta(q_2, 0)$
$\delta(q_2, 1)$

	0	1
$\rightarrow q_0$	q_1	q_0
q_1	q_1	q_2
$*q_2$	q_2	q_2

Note :-

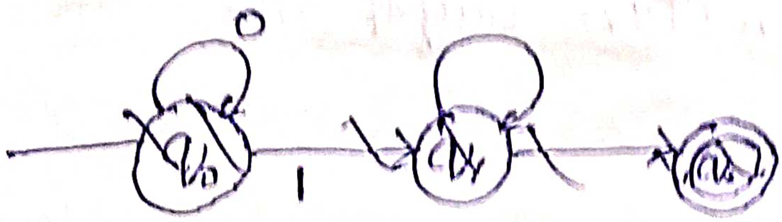
In a DFA from each state of transition graph there must be exactly one transition for each input symbol of the alphabet.

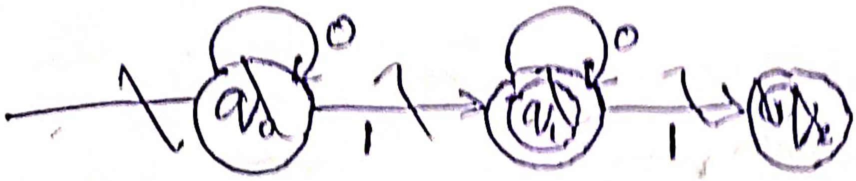
Represent the following DFA using transition graph.

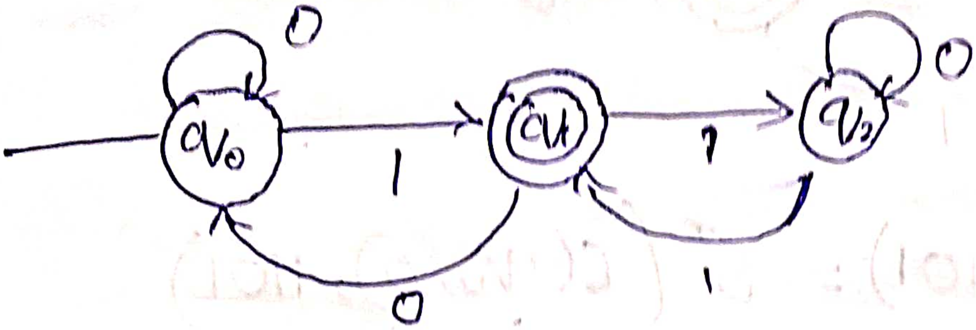
$$M = (\{q_0, q_1, q_2\}, \{0,1\}, \delta, q_0, \{q_1\})$$

where δ is

	0	1
$\rightarrow q_0$	q_0	q_1
$+ q_1$	q_0	q_2
q_2	q_2	q_1







Extended transition function, δ^*

δ^* is defined recursively as $\delta^*(p, x) = q$

eg:- $\delta(q_0, a) = q_1$

$\delta(q_1, b) = q_2$.

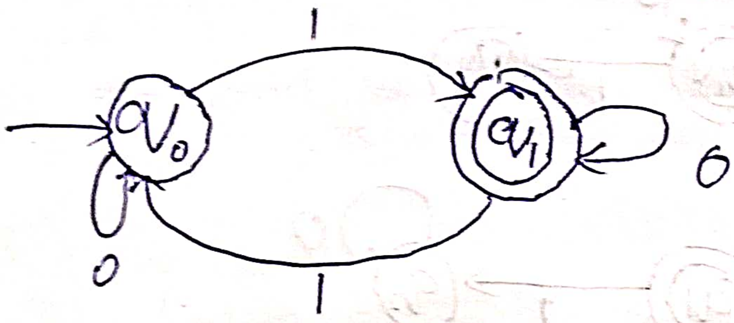
$\delta^*(q_0, ab) = \delta^*(\delta(q_0, a), b)$

$= \delta^*(q_1, b)$

$= \delta(q_1, b)$

$= q_2$.

Check whether the given DFA accepts the input string 01101



$$\delta^*(q_0, 01101) = \delta^*(\delta(q_0, 0), 1101)$$

$$= \delta^*(q_0, 1101)$$

$$= \delta^*(\delta(q_0, 1), 101)$$

$$= \delta^*(q_1, 101)$$

$$= \delta^*(\delta(q_1, 1), 01)$$

$$= \delta^*(q_0, 01)$$

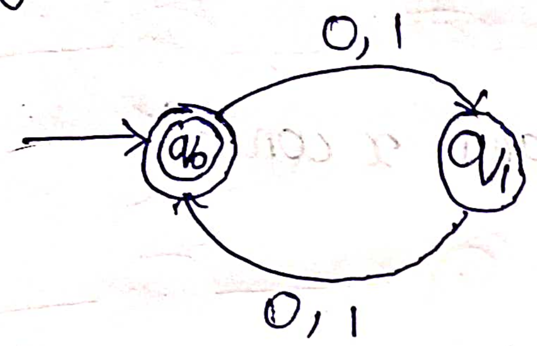
$$= \delta^*(\delta(q_0, 0), 1)$$

$$= \delta^*(q_0, 1)$$

$= \cancel{\delta^*}\, q_1$, final state

Hence the given DFA accepts the string 01101

Check whether the given DFA accepts the string 100



$$\delta^*(q_0, 100) = \delta^*(\delta(q_0, 1), 00)$$

$$= \delta^*(q_1, 00)$$

$$= \delta^*(\delta(q_1, 0), 0))$$

$$= \delta^*(\delta(q_0, 0))$$

$$= \delta^*(q_1$$, not final state.

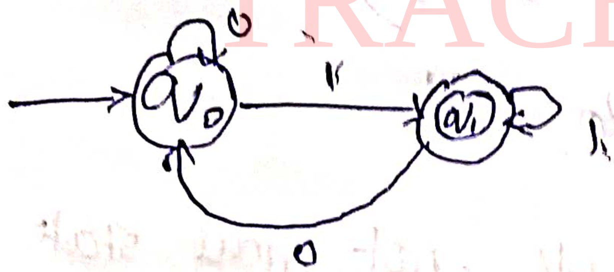
Hence the given DFA does not accept the string 100.

Q) find the language of given DFA.

$L(D) = \{x \mid x \in \{0,1\}^*$ and x contains the sub-string $10\}$

$= \{10, 110, 1110, 100111 \ldots\}$

Que. Find the language of given DFA



01011

$L(D) = \{x \mid x \in (0,1)^*$ and x which contain the ~~Subset 01~~ Substring Ending in 1$\}$

~~$\{01, 001011, 1011100, 101100 \ldots\}$~~

Language $L = \{x \mid x \in \{0,1\}^* \text{ and } x \text{ Ends in } 1\}$

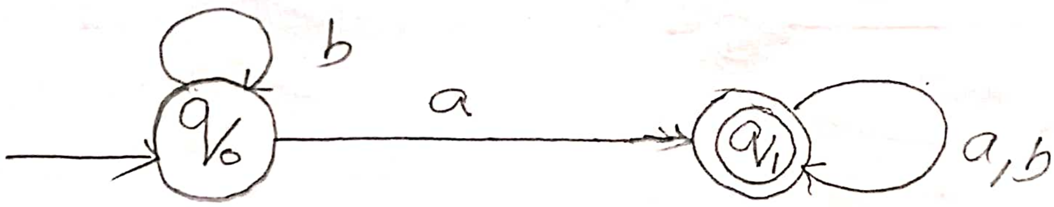
$= \{01, 010101, 01101, 00110101 \ldots\}$

Draw DFA to accept strings of a's and b's having atleast 1 a

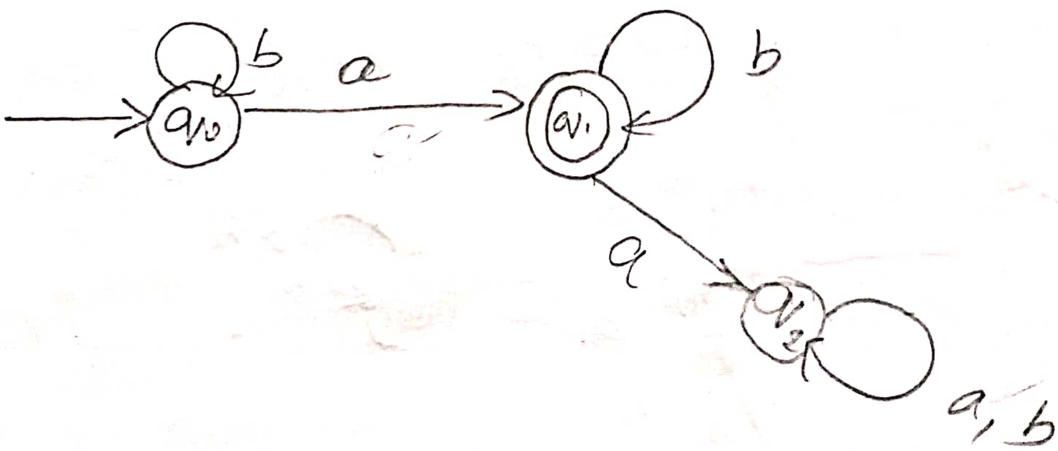
$\Sigma = \{a, b\}$

A'B

D.



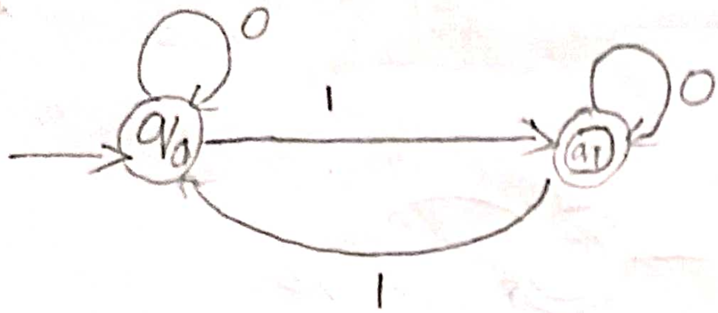
Design a DFA to accept strings of a & b having exactly 1a



Construct DFA for the following language

$L = \{x \mid x \in \{0,1\}^* \text{ and } x \text{ contains odd of 1's}\}$

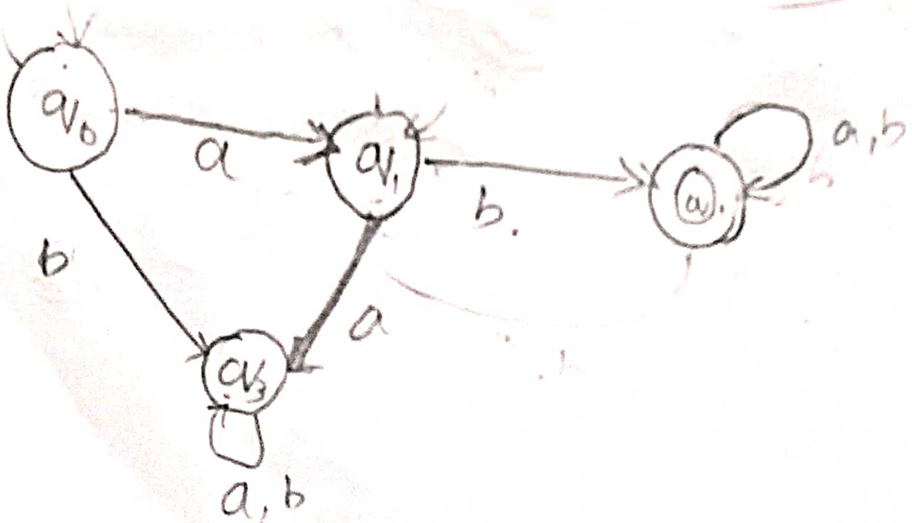
$\{1, 01, \cancel{011}$ 100, $\cancel{101}$, 100111, ~~100~~, 10111, . . . $\}$



Q) Draw transition graph for a given language.

$L = \{abx \mid x \in \{a,b\}^*\}$

$\{ab, aba, abbba, ababab, \ldots\}$



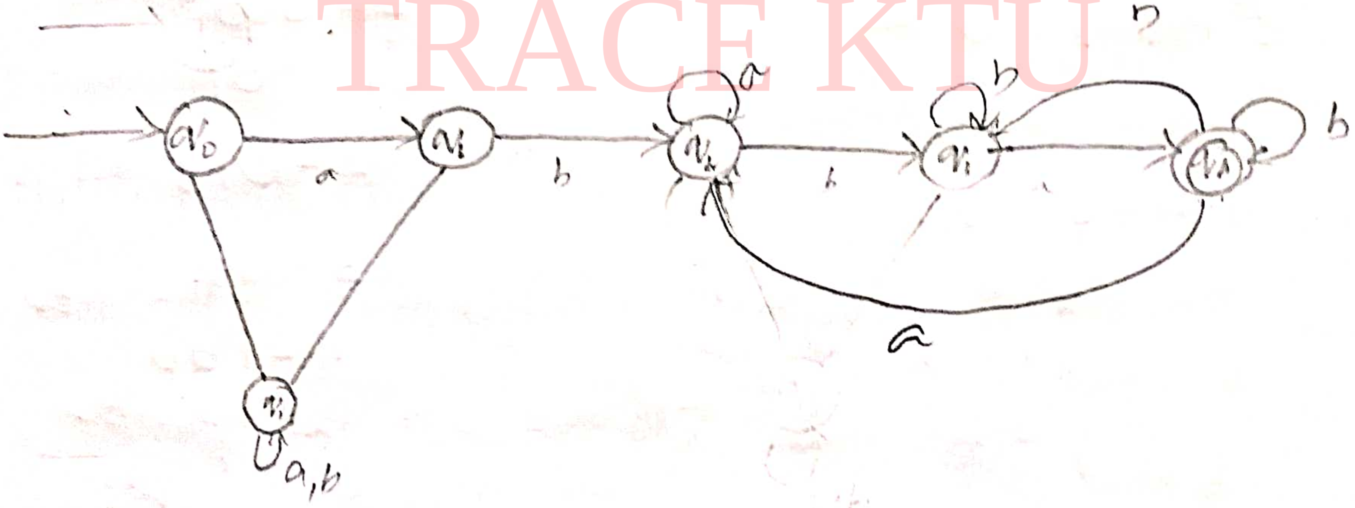
construct DFA for the following languages

I) ~~Set~~ Set of strings such that every string ends in 00 over an alphabet $\Sigma = \{0,1\}$

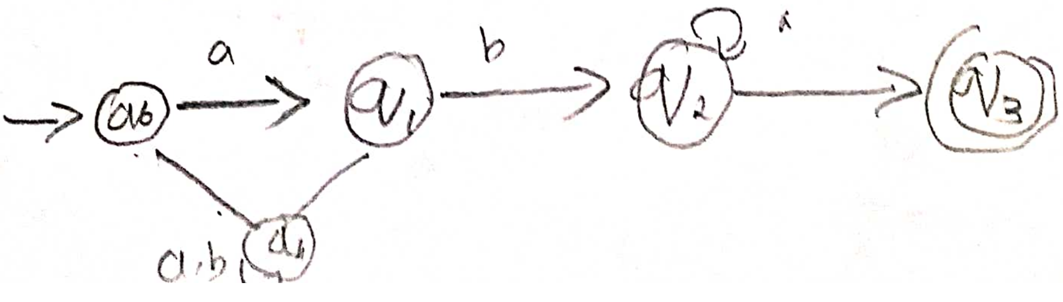
II) Set of all strings Ending with ba over an alphabet $\Sigma = \{a,b\}$

Q) $L = \{abxba \mid x \in \{a,b\}^*\}$

$\{abba \quad ababa \quad ababba \quad abbaabba\}$

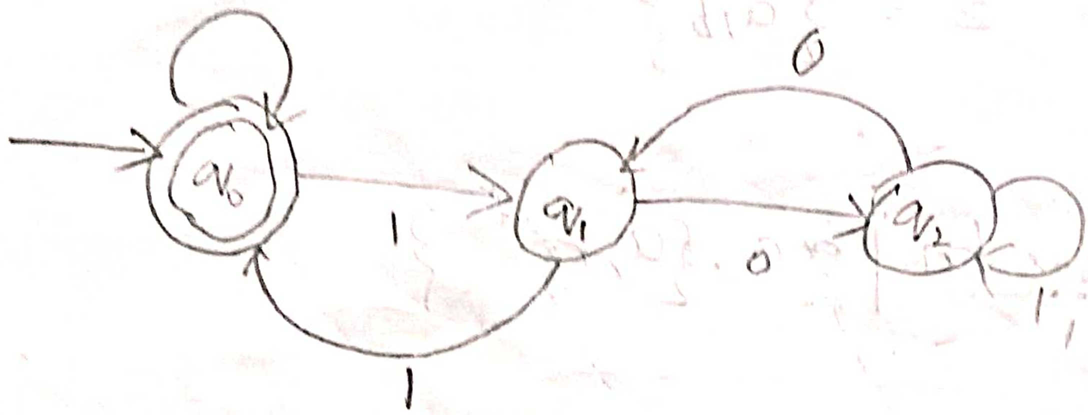


Design a DFA for the language $L = \{x \mid x \in \{a,b\}^*$ and x starts with ab and ends with ba $\}$



construct a DFA for the language

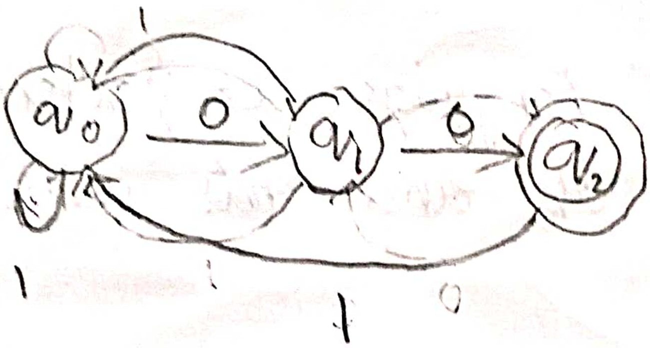
$L = \{ x \in \{0,1\}^* / x$ represents a multiple of 3 in binary$\}$



0 mod3 = 0
1 mod3 = 1
2 mod3 = 2
3 mod3 = 0
4 mod3 = 0
5 mod3 = 1

0 - 000
1 - 001
2 - 010
3 - 011
4 - 100
5 - 101
6 - 110
7 - 111

i)

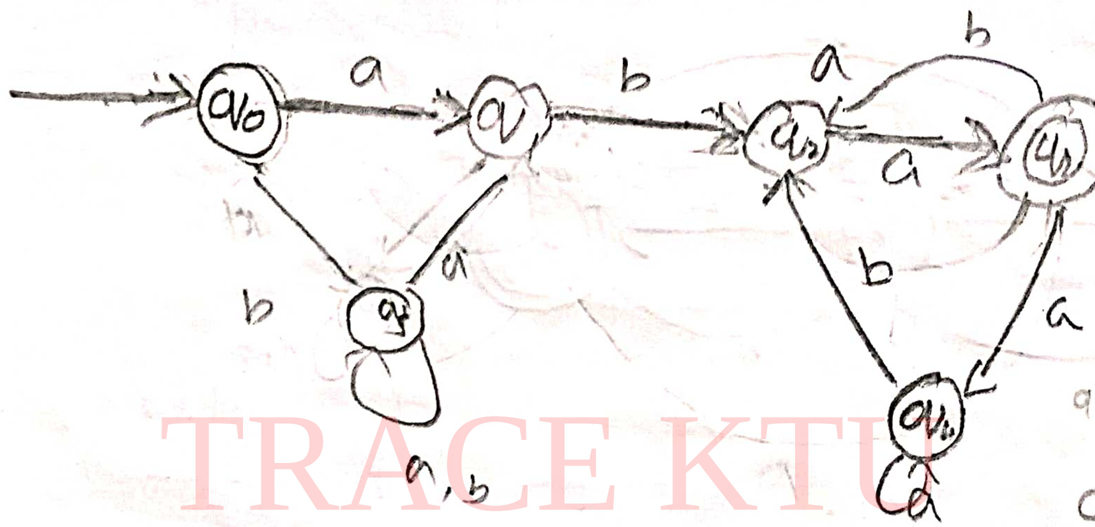


ii)

Multiple of 5

decimal	binary	%	State
0	000	0 % 5 = 0	q_0
~~2~~ 1	001	1	q_1
2	010	2	q_2
3	011	3	q_3
4	100	4	q_4
5	101	0	q_0
6	110	1	q_1
7	111	2	q_2

Q) Design a DFA for the Language $L = \{x \mid x \in \{a,b\}^*$ and x starts with ab and Ends with ba $\}$



abba, abbbba,

abaaba

abbbba,

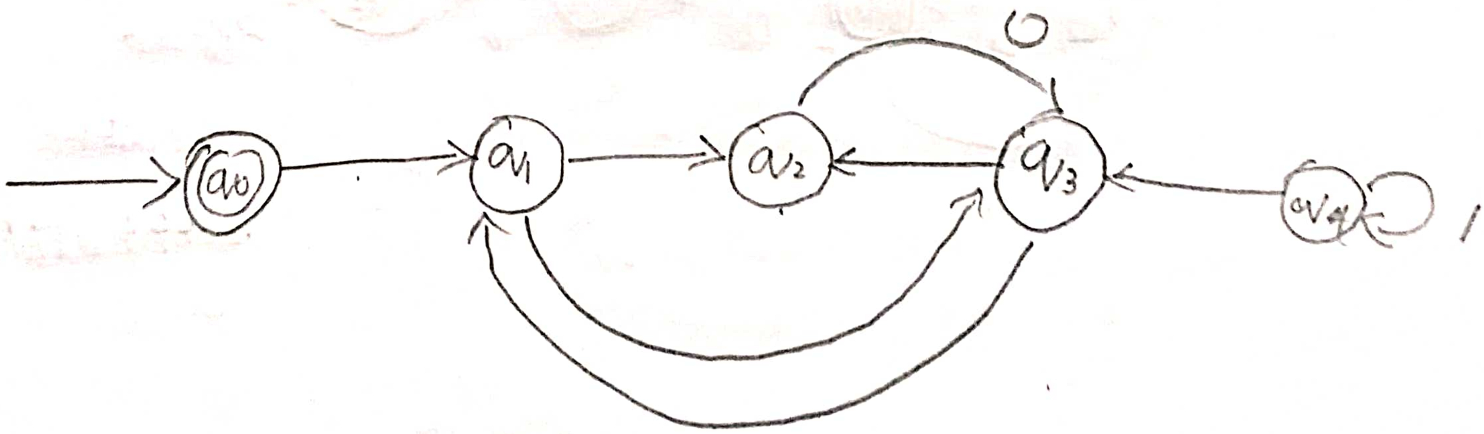
ababba,

abaaaba,

abaaba

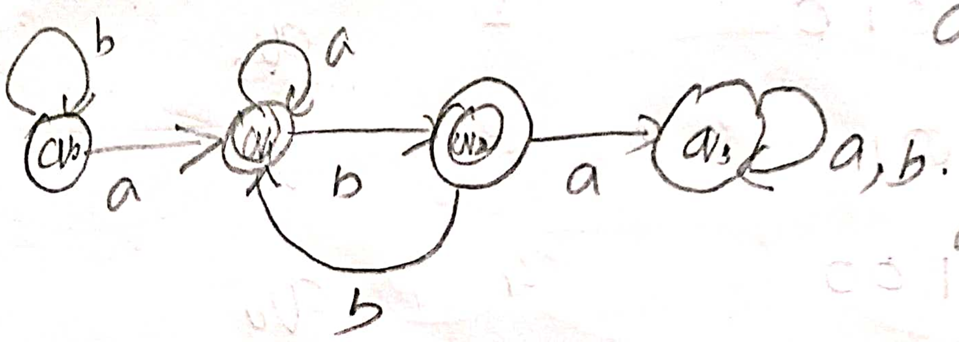
Q) Multiple of 5

decimal	binary	%	state
0	000	0 % 5 = 0	q_0
1	001	1	q_1
2	010	2	q_2
3	011	3	q_3
4	100	4	q_4
5	101	0	q_0
6	110	1	q_1
7	111	2	q_2



Design a DFA for the language ~~L =~~ {

$L = \{ x \in \{a, b\}^*$, abba is not a sub string of x }



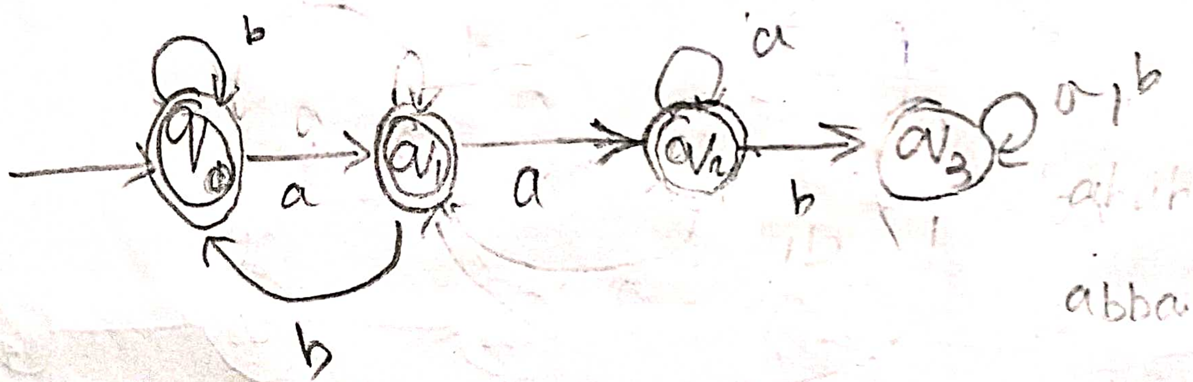
aa, abba,

abbaba

should not come

Qn. Construct DFA to accept strings of a & b except ~~to~~ those having aab

L =



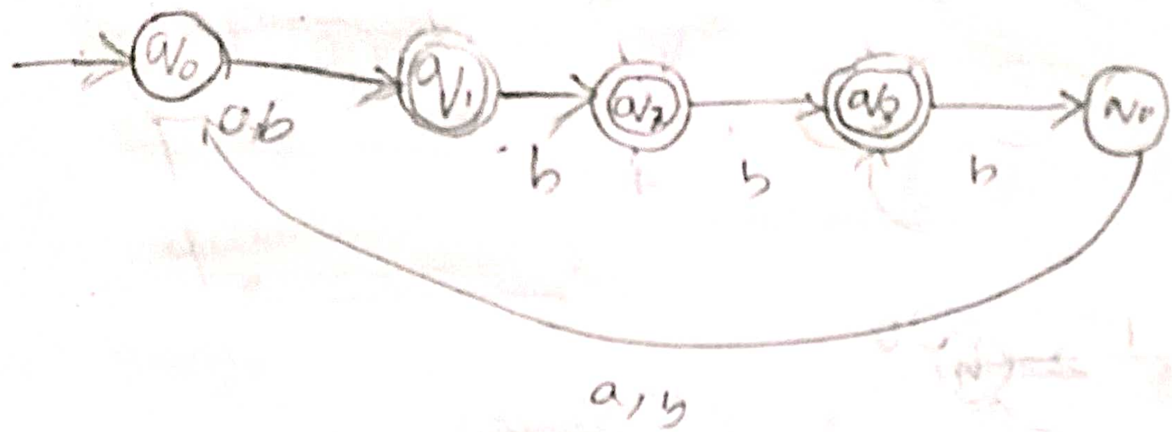
abbabba

abbbabbb

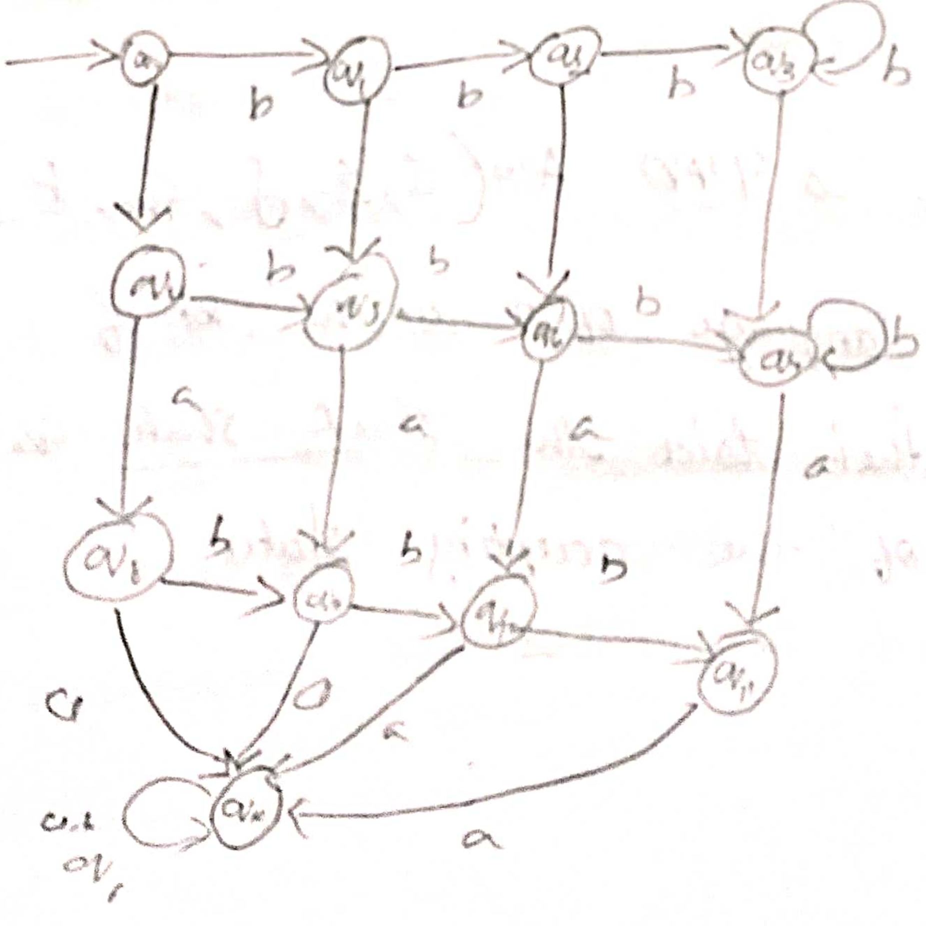
a

Q) Design DFA to accept the language

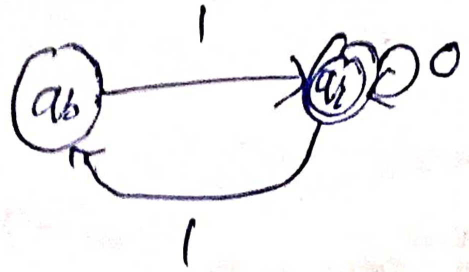
$L = \{w \mid w \in \{a,b\}^* \text{ and } |w| \bmod 5 \neq 0\}$



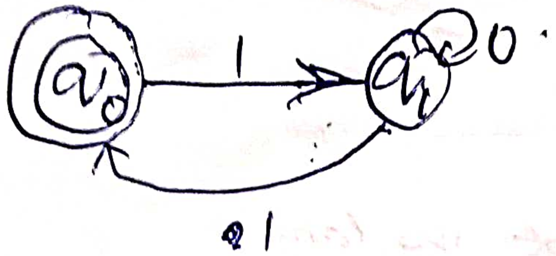
Q) construct a DFA to accept the language set of all strings that accept Exactly 2a's and more than 2b's



Odd no



Even no's



Regular language

~~Language~~

Let A be a DFA $A = (Q, \Sigma, \delta, q_0, F)$

then the language of A is the set of strings w that take the start state q_0 to one of the accepting states

$L(A)\{w \mid \delta^*(q_0, w)$ is in $F\}$

if there is such a DFA for the Language L then we can say L is a Regular Language.

Non deterministic finite automata

NFA : a non deterministic finite automata is a 5 tuple $(Q, q_0, \Sigma, \delta, F)$. where

Q – Set of finite set of states

$\Sigma \Rightarrow$ finite set of Input ~~signals~~ Symbols

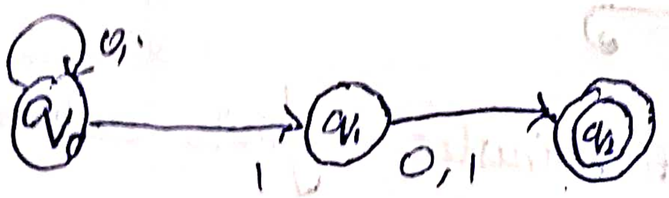
$q_0 \Rightarrow$ start state, $q_0 \in Q$.

$F \Rightarrow$ Set of final states, $F \subset Q$.

$\delta \Rightarrow$ The transition function which takes a state in Q and an input symbol in Σ as arguments and returns a subset of Q.

$\delta : Q \times \Sigma \rightarrow 2^Q$:- power set of Q.

NFA that accepts exactly those strings that have symbol one in the second last position



$\delta(q_0, 1) = \{q_0, q_1\}$

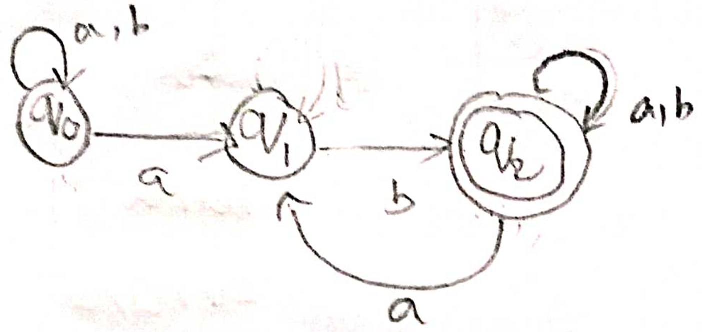
$\delta(q_2, 0) = \phi$

$\delta(q_2, 1) = \phi$

$(\{q_0, q_1, q_2\}, \{0,1\}, \delta, q_0, \{q_2\})$

	0	1
q_0	q_0	q_0, q_1
q_1	q_2	q_2
q_2	ϕ	ϕ

construct NFA that accepts strings having Substring ~~AB~~ ab



abab

abba

abbbaab

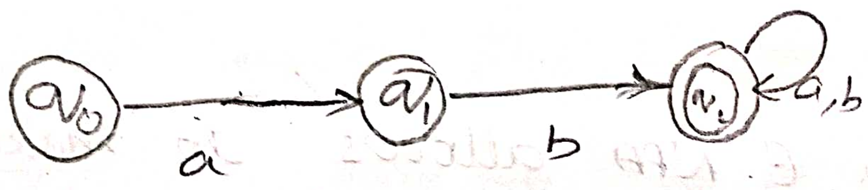
Construct NFA that accept all strings starting with ~~AB~~ ab

ababab

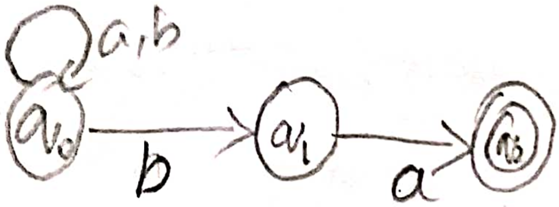
abbbabab

abaabba

abaaabai



Construct NFA that accept all string Ending with ~~BA~~ ba



baba

abbaba

aaabba

NFA with ε transition

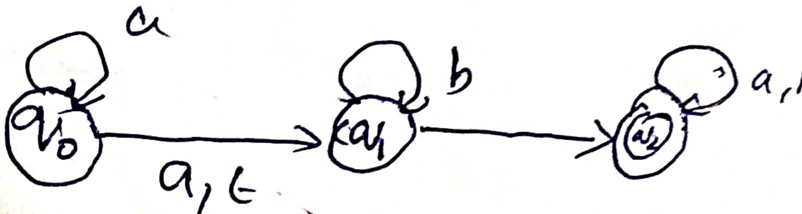
NFA with ε transition or ε NFA is defined as

$$N_\epsilon = (Q, \Sigma, \delta, q_0, F)$$

where all components have their same interpretation as for an NFA, Except δ which is defined as $\delta : Q \times (\Sigma \cup \{\epsilon\}) \longrightarrow 2^Q$

ie, an ε NFA allows to make a transition without recieving an input symbol

Eg:-



	a	b	ε
q_0	$q_0\ q_1$	ϕ	q_1
q_1	ϕ	q_1	q_2
q_2	q_2	q_2	ϕ

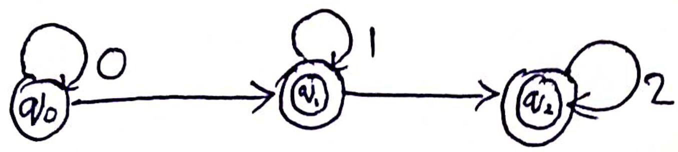
$(\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, \{q_2\})$

ε ~~quantize~~ -closure.

ε-closure of a state q is defined recursively as follows

I) State q is in ε-closure of q

II) If state p is in ε-closure of q and there is a transition from state p to state r label ε then r is in ε-closure of q.

eg:- 

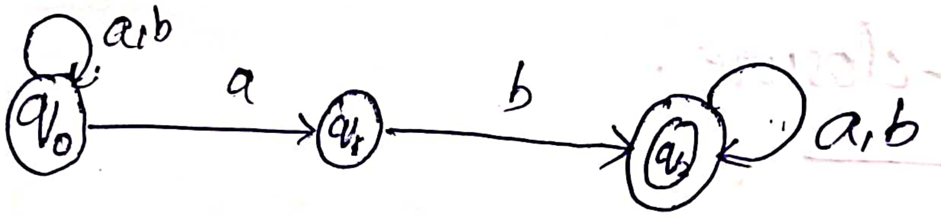
ε Closure $(q_0) = \{q_0, q_1, q_2\}$

ε closure $(q_1) = \{q_1, q_2\}$

ε-closure $(q_2) = \{q_2\}$

Equivalence of DFA and NFA

Construct DFA Equivalent to given NFA



We have NFA as $(Q_N, \Sigma, \delta_N, q_0, F_N)$

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

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

$q_0 = \{q_0\}$

$F_N = \{q_2\}$.

DFA:-

By Subset construction method we define the ~~sea~~ Equivalent DFA, D as $D = (Q_D, \Sigma, \delta_D, q_D, F_D)$

δ_D is as followed

Q_D	a	b
$\{q_0\}$ P_1	$\{q_0, q_1\}$ P_2	$\{q_0\}$ P_1
$\{q_0, q_1\}$ P_2	$\{q_0, q_1\}$ P_2	$\{q_0, q_2\}$ P_3
$\{q_0, q_2\}$ P_3	$\{q_0, q_1, q_2\}$ P_3	$\{q_0, q_2\}$ P_3
$\{q_0, q_1, q_2\}$ P_4	$\{q_0, q_1, q_2\}$	$\{q_0, q_2\}$ P_3

$Q_D = 2^{Q_N}$

$q_D = \{q_0\}$

$F_D = \{S \mid S \in Q_D$ and $S \cap F_N \neq \phi\}$

$\delta_D(S, a) = \bigcup_{P \in S} \delta_N(P, a)$

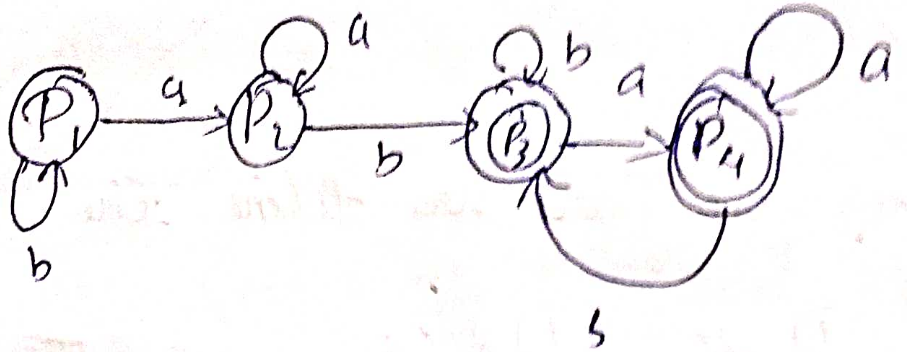
for each Subset of Q_N

$Q_D = \{P_1, P_2, P_3, P_4\}$

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

$F_D = \{P_3, P_4\}$

DFA.



Theorem :-

Corresponding to an NFA N accepting a language L there exist a DFA D which also accepts L.

OR

If $D = (Q_D, \Sigma, \delta_D, q_D, F_D)$ is the DFA constructed from the NFA $N = (Q_N, \Sigma, \delta_N, q_0,$ using subset construction method, $L(D) = L(N)$

To prove the above theorem we have to show that

$$\delta_D^*(q_0, x) = \delta_N^*(q_0, x) \text{ for all } x \in \Sigma^*$$

<u>Basis</u>.

~~Let~~ length of $x = 0$

Let $|x| = 0$

ie, $x = \epsilon$

$\cancel{LHS = \delta_D^*(q_0, \epsilon) = \delta_D(}$

$LHS = \delta_D^*(q_0, \epsilon) = q_0 = \{q_0\}$

$RHS = \delta_N^*(q_0, \epsilon) = \{q_0\}$

Hence $LHS = RHS$; Hence it is true for $x = \epsilon$

INDUCTION

Assume that the claim is true for all strings while in δ^* with length = k.

$$|y| = k.$$

$$\delta_D^*(q_0, y) = \delta_N^*(q_0, y)$$

~~Take an~~

Now let us prove the same is true for a string of length k+1

$$|x| = k+1$$

$$x = ya.$$

$$LHS = \delta_D^*(q_D, x)$$

$$\delta_D^*(q_0, ya)$$

$$= \delta_D(\delta_D^*(q_0, y), a)$$

$$= \delta_D(\{P_1, P_2, \cdots\}, a)$$

$$= \bigcup_{i=1}^{k} \delta_N(P_i, a)$$

$$RHS = \delta_N^*(q_0, x)$$

$$= \delta_N^*(q_0, ya)$$

$$= \delta_N(\delta_N^*(q_0, y), a)$$

$$= \delta_N(\{P_1, P_2, \cdots P_k\}, a)$$

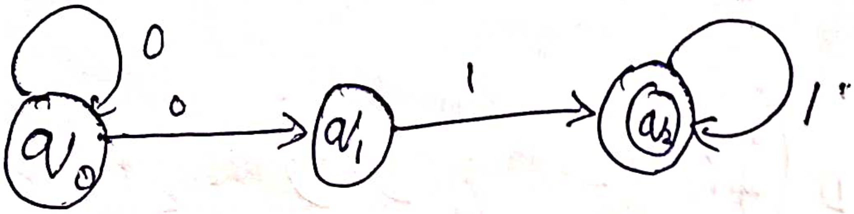
$$= \delta_N(P_1, a) \cup \delta_N(P_2, a) \cup \cdots$$

$$= \bigcup_{i=1}^{k} \delta_N(P_i, a) \longrightarrow (2)$$

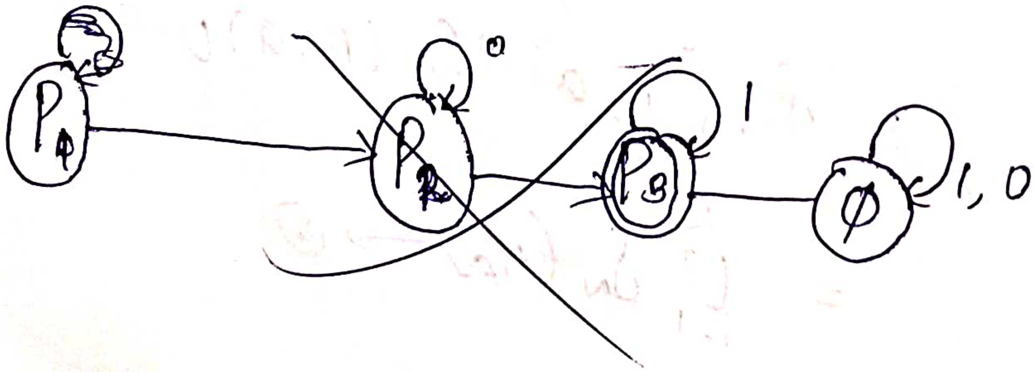
$$LHS = RHS$$

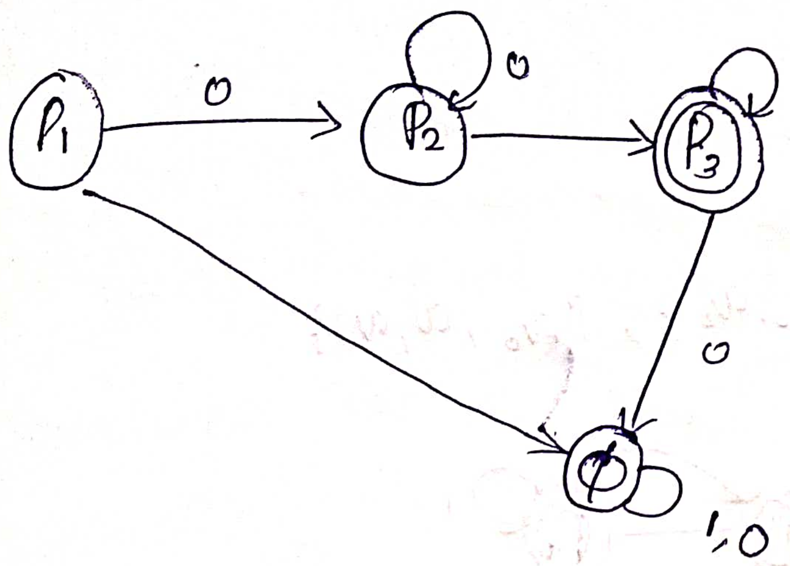
Hence the above theorem is proved by Induction principle

Q) construct a DFA for the following NFA using subset construction method.



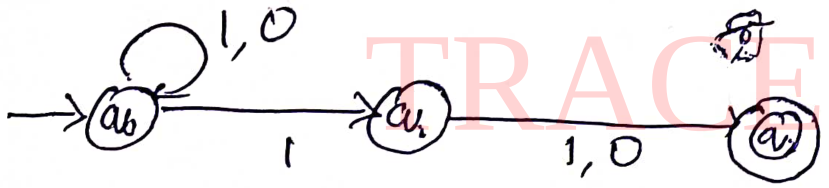
Q_0	0	1
$\{q_0\}$ P_1	$\{q_0, q_1\}$ P_2	ϕ
$\{q_0, q_1\}$ P_2	$\{q_0, q_1\}$ P_2	$\{q_2\}$ P_3
$\{q_2\}$ P_3	ϕ	$\{q_2\}$ P_3
ϕ	ϕ	ϕ





Hw.

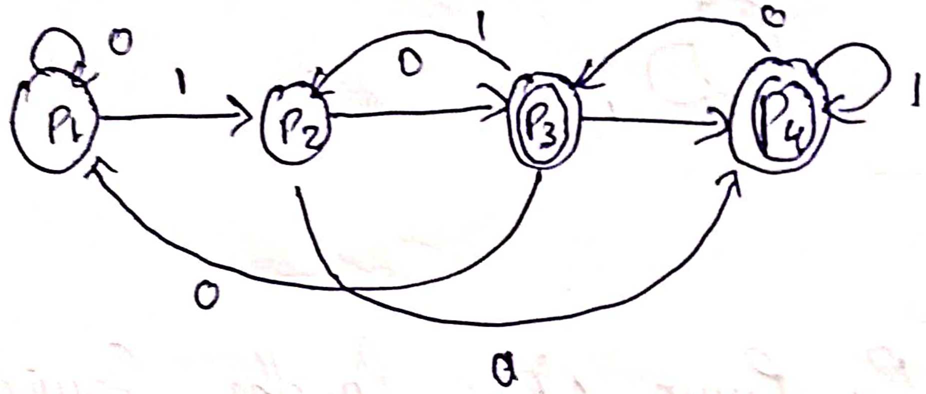
Convert the Qy Given NFA to the Equivalent DFA



Q_D	0	1
P_1 $\{q_0\}$	$\{q_0\}$ P_1	$\{q_0, q_1\}$ P_2
P_2 $\{q_0, q_1\}$	$\{q_0, q_2\}$ P_3	$\{q_0, q_1, q_2\}$ P_4
P_3 $\{q_0, q_2\}$	$\{q_0\}$ P_1	$\{q_0, q_1\}$ P_2
$\{q_0, q_1, q_2\}$ P_4	$\{q_0, q_2\}$ P_3	$\{q_0, q_1, q_2\}$ P_4

$Q_D = \{P_1, P_2, P_3, P_4\}$

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



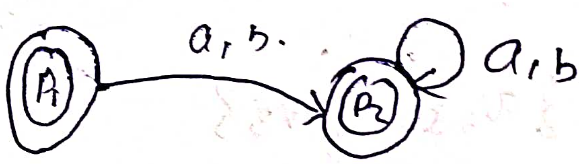
Equivalence of DFA and ε NFA

Q) Convert the following ε NFA to equivalent DFA using subset construction method.

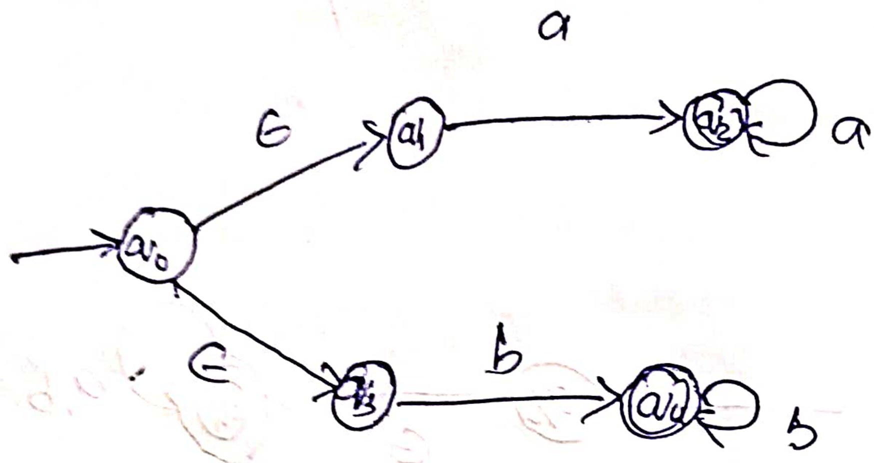
ϵ-closure $\{q_0\} = \{q_0, q_1\}$.

ϵ-closure $\{q_1\} = \{q_1\}$

Q_D	a	b
$\{q_0, q_1\}^{P_1}$	$\{q_1\}^{P_2}$	$\{q_1\}^{P_2}$
$\{q_1\}^{P_2}$	$\{q_1\}^{P_2}$	$\{q_1\}^{P_2}$



Obtain the DFA Equivalent to Given ENFA by applying the subset construction method



~~q_0~~	~~a~~	~~b~~
~~q_1, q_3~~		

ε- closure $\{q_0\} = \{q_0, q_1, q_3\}$

ε - closure $\{q_1\} = \{q_1\}$

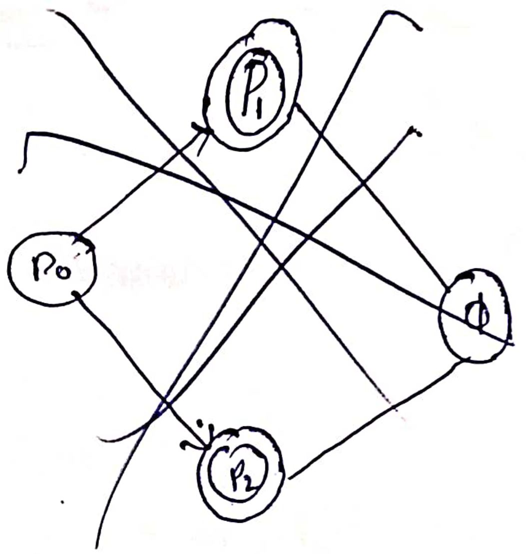
ε - closure $\{q_2\} = \{q_2\}$

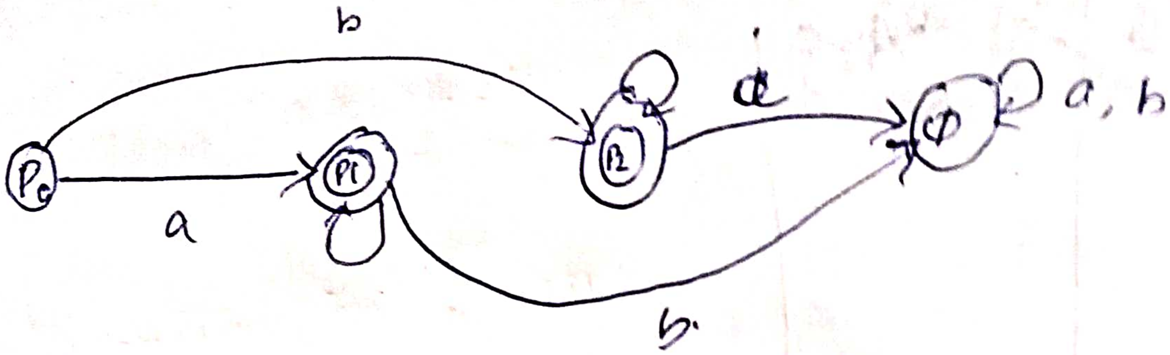
ε - closure $\{q_3\} = \{q_3\}$

ε - closure $\{q_4\} = \{q_4\}$

Q_P	a	b
~~$\{q_0, q_1\}$~~	~~$\{q_1\}$~~	~~$\{q_4, q_2\}$~~

Q_P	a	b
$\{q_0, q_1, q_3\}^{P_1}$	$\{q_2\}^{P_2}$	$\{q_4\}^{P_3}$
$\{q_2\}^{P_2}$	$\{q_2\}^{P_2}$	ϕ
$\{q_4\}^{P_3}$	ϕ	$\{q_4\}^{P_3}$




b
a
b
a, b
a
b