Non empty and finite

$\Sigma_{eng} = \{A, B, C, \ldots Z; a, b, c, \ldots z\}$

$\Sigma_{bin} = \{0, 1\}$

$\Sigma_{dec} = \{0\text{-}9, .\}$

Words / Strings.

→ Finite Sequence of alphabate

→ Denoted by (w)

$w = 0101$

It is a string over alphable of (0,1)

$|w|$ = length of String = No. of charater present in String

$\boxed{\epsilon = \text{Empty string. (Epsilon)}}$

Σ^* = Set of all string (including Empty string) over Σ

Language (L)

→ Set of String over any Σ
→ It is denoted by (L)
→ Subset of Σ^*

Grammar (G)

→ set of rules to produce a valid string in a language
→ dented by (G)

Sub String

→ Given Σ u is a substring of z if $\exists\ z = xuy$ where $x, u, y, z \in \Sigma^*$

$z = \underbrace{\epsilon}_{x}\underbrace{010}_{u}\underbrace{\epsilon}_{y}$ $\quad z = \underbrace{1}_{x}\underbrace{01}_{u}\underbrace{0}_{y}$ $\quad \boxed{S = n\left(\frac{n+1}{2}\right) + 1}$

No. of Substring

Prefix

→ Given Σ u is prefix of z if $z = ux$ where $u, x, z \in \Sigma^*$

$z = \underbrace{01}_{u}\underbrace{01}_{x}$; $z = \underbrace{\epsilon}_{u}\underbrace{01}_{x}$

Postfix

→ Given Σ u is postfix of z if $z = xu$ where $u, x, z \in \Sigma^*$

$z = \underbrace{01}_{x}\underbrace{11}_{u}$

Operation on alphabate, string & language

→ power of an alphabate.

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

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

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

$\Sigma^2 = \{00, 01, 10, 11\}$

$\Sigma^3 = \{ \ldots \}$

$\vdots$

$\Sigma^K = \{ w \mid |w| = K \}$

→ Kleen closure

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

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

$$\boxed{\Sigma^* = \bigcup_{i \geq 0}^{\infty} \Sigma^i}$$

$\Sigma^* = \{ w \mid |w| \geq 0 \}$

→ Positive closure (Σ^+)

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

$$\Sigma^+ = \bigcup_{i \geq 1}^{\infty} \Sigma^i$$

$\Sigma^+ = \{ w \mid |w| \geq 0 \}$

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

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

$\Sigma^* = \{\epsilon, a, b, aa, ab, bb, ba, aaa, aab, abb, bbb, bba, baa \ldots\}$

$\Sigma^+ = \{a, b, aa, ab, bb, ba, aaa, bbb, aab, abb, bba, baa \ldots\}$

Basic properties of Σ^* & Σ^+

→ $\Sigma^* = \Sigma^+ \cup \{\epsilon\}$

→ $\Sigma^* \cup \Sigma^+ = \Sigma^*$

→ $\Sigma^* \cap \Sigma^+ = \Sigma^+$

→ $\Sigma^* \Sigma^* = \Sigma^*$

→ $\Sigma^* \Sigma^+ = \Sigma^+ \Sigma^* = \Sigma^+$

1) $L = \Sigma^*$ (universal lang)

2) $L = \Sigma^+ \leq \Sigma^*$

3) $L = \{0, 00, 111, 010, 1010\}$

$L = \{00, 01, 10, 11\}$

$= \{w \mid |w| = 2\}$

4) $L = \{\ \} = \phi$ (does not even contain ϵ)

5) $L =$ (contain finite string) = finite language

$|L| =$ finite

Q(1) $L = \{0^n \mid n \geq 1\}$

$= \{0, 00, 000 \ldots\ldots\}$

(2) $L = \{0^n 1^n \mid n \geq 0\}$

$= \{\epsilon^\circ, 01, 0011, 000111 \ldots\ldots\}$

(3) $L = \{0^n 1^m \mid n = m ; 1 \leq n < 4\}$;

$= \{01, 0011, 000111, 00001111\}$

6) L = Contain infinte string = infinte language

Q given $\Sigma = \{$क, ख$\}$ find power an Σ^3 Kleen positive clouser and justfy if $\Sigma^+ - \epsilon^\circ = \Sigma^*$ or not...?

$\Sigma^+ = \{$ककक, ककख, कखक, खकक, खकख, कखख, खखक, खखख $\}$

$\Sigma^* = \{\epsilon^\circ$, ककक, ककख, कखक, खकक, खकख, कखख, खखक, खखख $\}$

Finite state Machine / Finite Automata (FA

→ It is a simplest machine which is used to recognize pattern

Application • FA to determine given no. is even or odd

• FA for simple mail verification.

•

→ It is an abstract machine which contain five elements or tuples $<Q, q_0, F, \Sigma, \delta>$

Q :- Finite set of states

q_0 :- Initial state

F :- set of final states

Σ :- Input Symbol

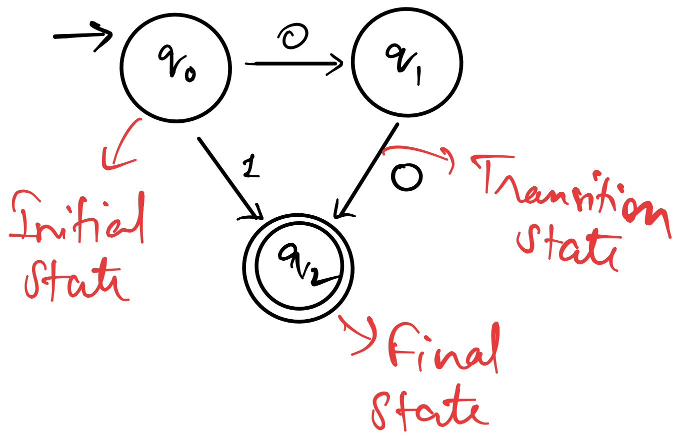
δ :- Transition Functions.

FA

- without output
  - DFA (Deterministic Finite Automata)
  - NFA (E-NFA) (Non-Deterministic Finite Automata)
- with output
  - Moore Machine
  - Mealy Machine

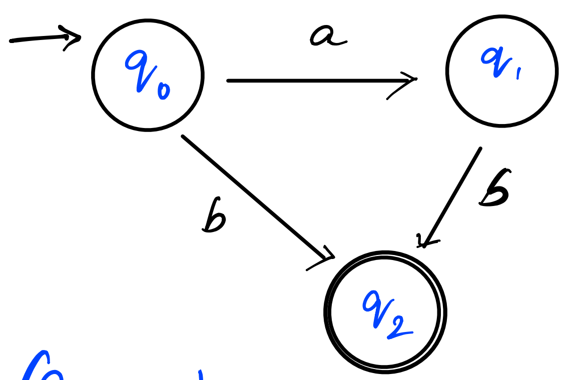
★ Representation of an FA

- Transition Diagram
- Transition Table



δ	0	1
$\rightarrow q_0$	q_1	q_2
q_1	q_2	–
q_2	–	–

δ = (Current State , Input Symbol) = Next State



$\delta(q_0, a) = q_1$

$\delta(q_0, b) = q_2$

$\delta(q_1, b) = q_2$

$Q(q_0, q_1, q_2)$

Mathematical – DFA

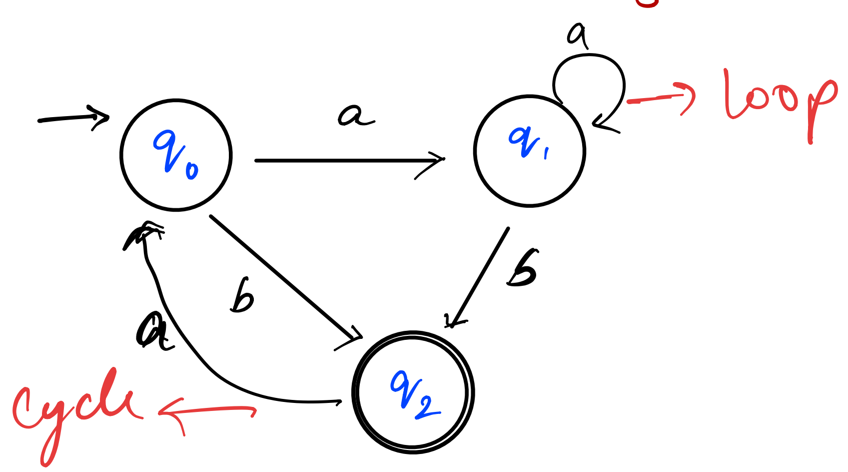
$M(Q, q_0, \Sigma, F, \delta)$

→ It is quin tuple as above

For DFA

$\delta : Q \times \Sigma = Q$

" There should be one and only one transition state to move from one state to another state "



→ Cycle & loop is present (Infinite language)

→ If Initial state is final state then it is an empty string (ϵ 0)

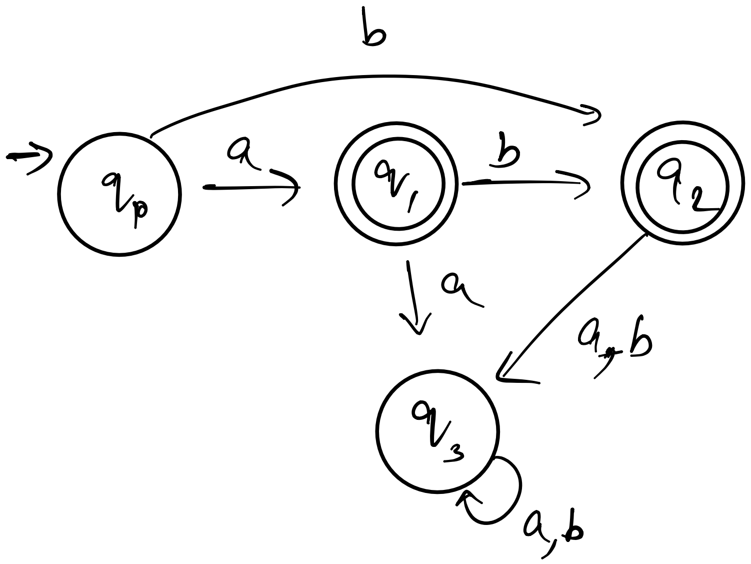
language of DFA

→ Set of all string acepted by DFA

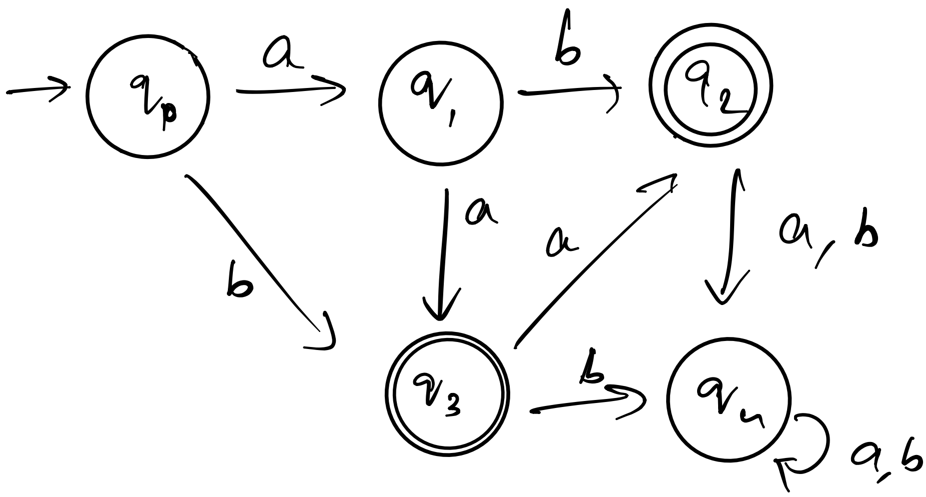
$$L(M) = \{ w \in \Sigma^* \mid w \text{ is acceptable by } M \}$$

$$L(M) = \{ w \in \Sigma^* \mid \delta(q_0, w) = \text{Final state} \}$$

Ques

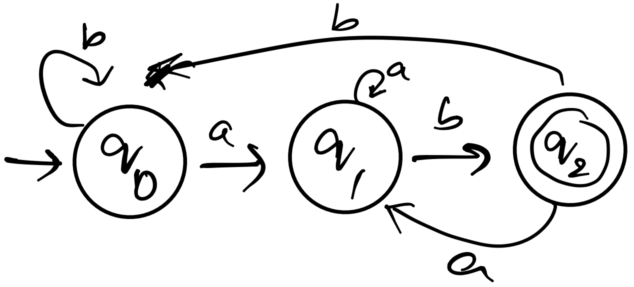


$L = \{a, ab, b\}$



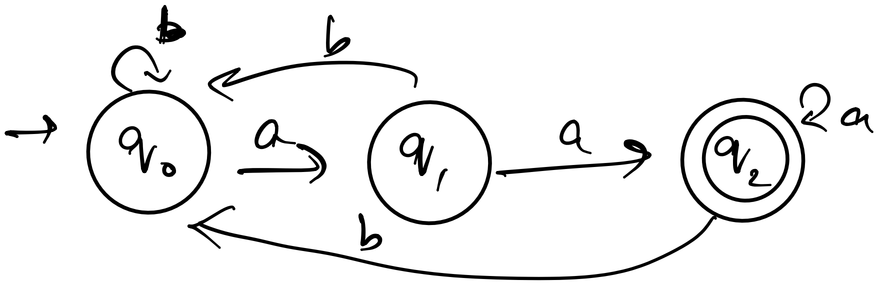
$L = \{ab, aa, b, ba, aaa\}$

Ques



$L = \{ w \mid w \text{ end with } ab \}$

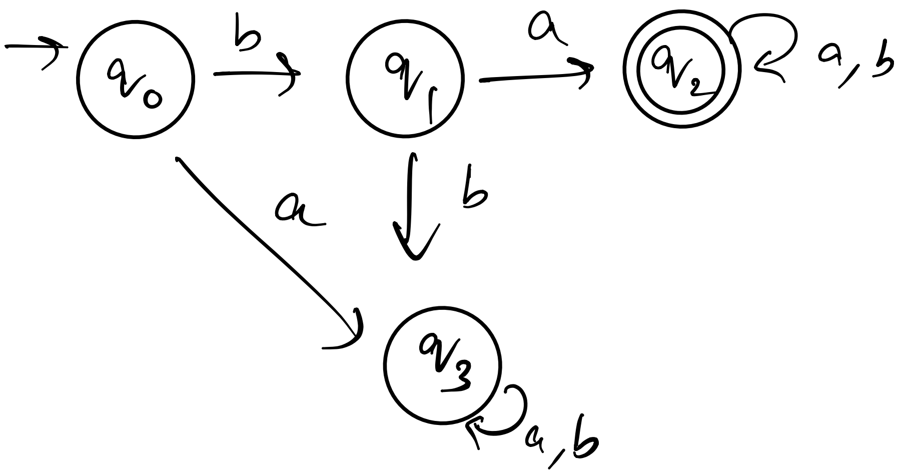
$L = \{ wab \mid w \in \Sigma^* \}$



$L = \{ aa,\ aaa^*,\ b^*aa,\ b^*aaa^*,\ aaa^*baa \}$

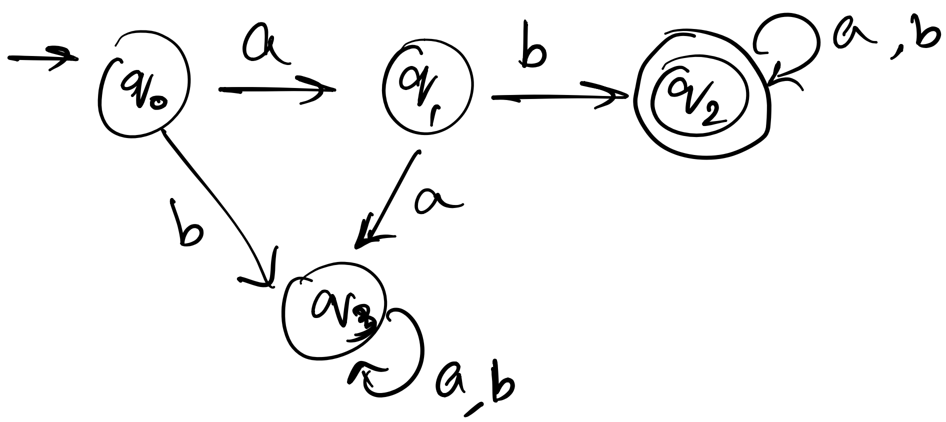
$L = \{ w \in \Sigma^* \mid w \text{ ends with } aa \}$

$L = \{ waa \mid w \in \Sigma^*$

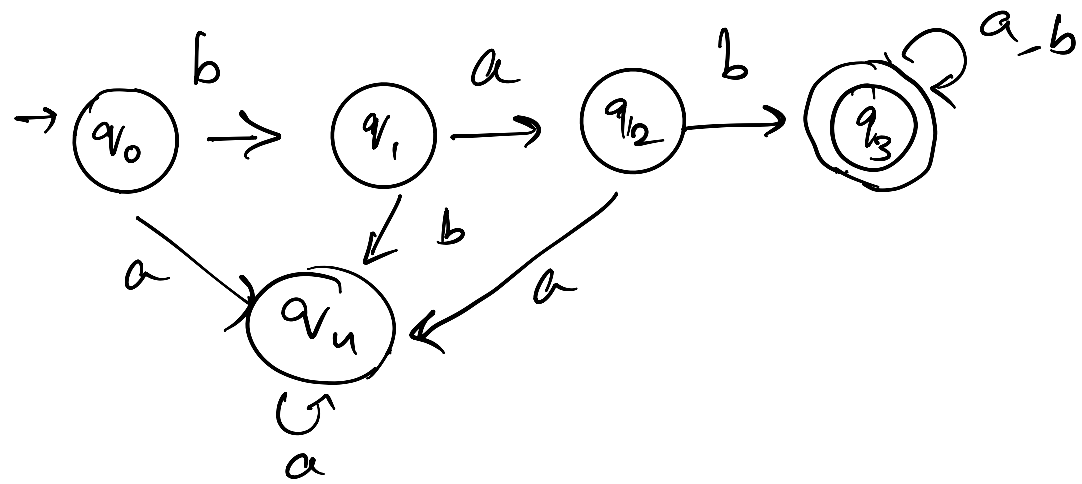


$L = \{ w \in \Sigma^* \mid w \text{ starts with } ba \}$

$L = \{ baw \mid w \in \Sigma^* \}$

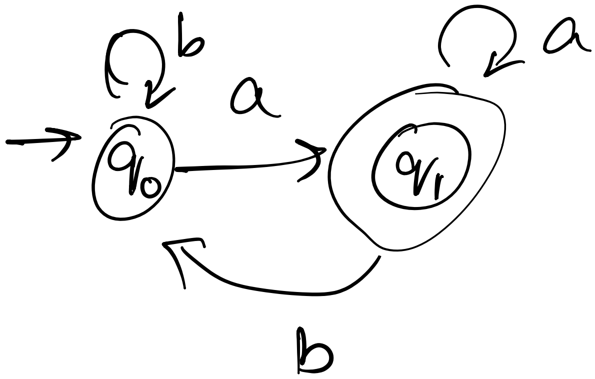


Q starts with bab :-

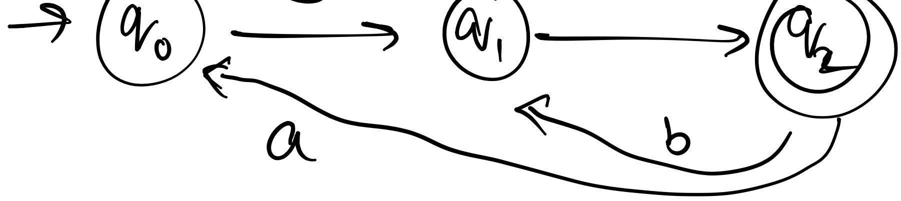


Q Ends with a

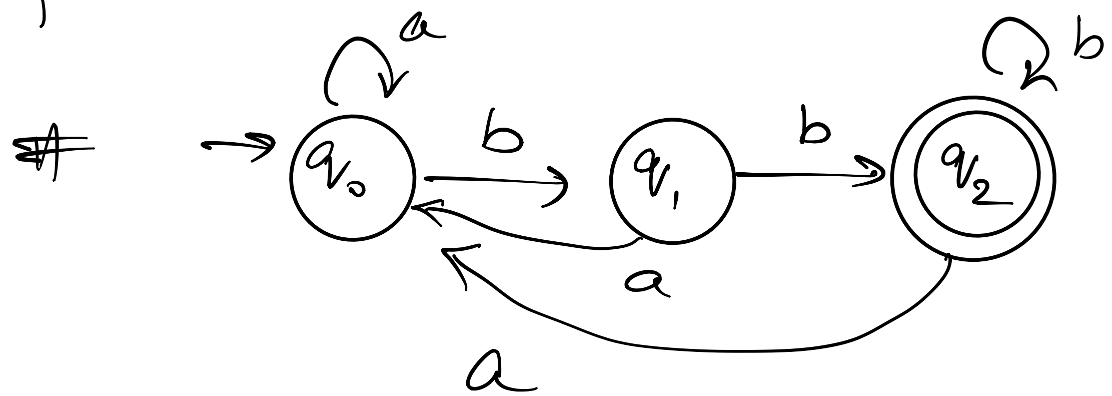
#



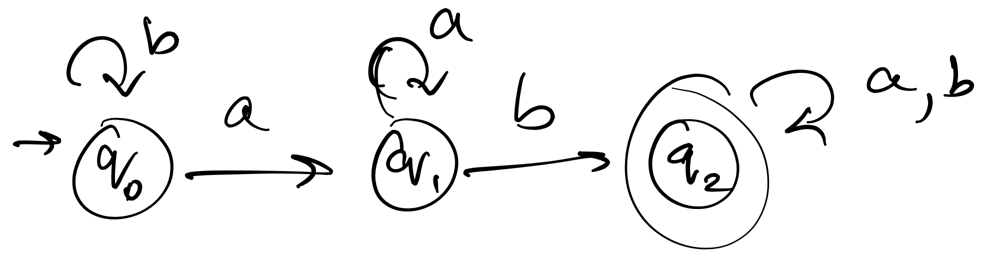
#



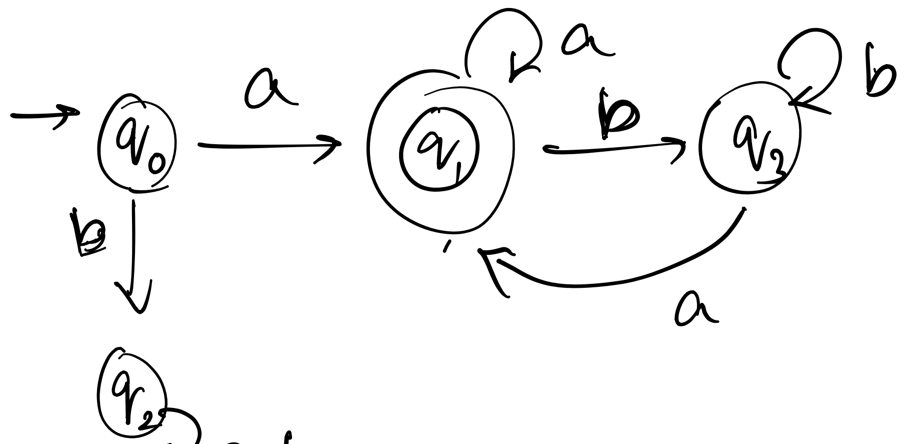
Q Endes with bb



Q Contain Substring ab.



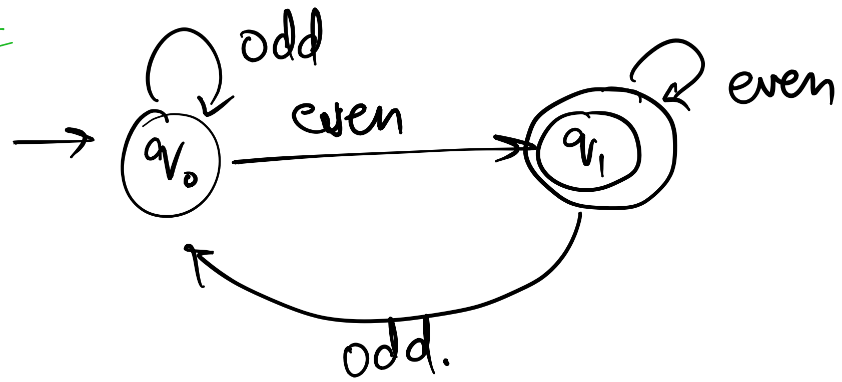
Q Starts and Ends with a



more Question left

Q) Construct a DFA over whole no. which only accept even number-?

$\Sigma = \{\underline{0}, 1, \underline{2}, 3, \underline{4}, 5, \underline{6}, 7, \underline{8}, 9\}$



NFA → Non-Deterministic Finite Automata.

→ A is a quin tuple (contain 5 elements) as below :

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

Q :- Finite set of states

q_0 :- Initial state

F :- set of final states

Σ :- Input Symbol

δ :- Transition Function

For NFA

$$\delta : Q \times \Sigma = 2^Q$$ (Subset of Q)

Acceptance by NFA

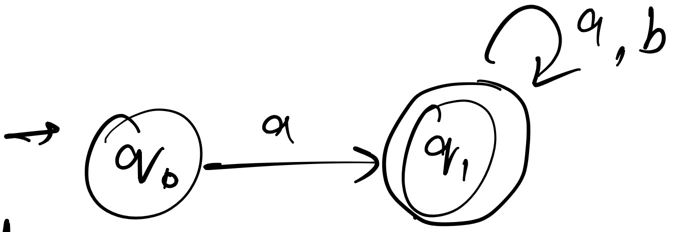
→ If any transition path lead us to a final State then we accept the string.

→ check for each and every transition path.

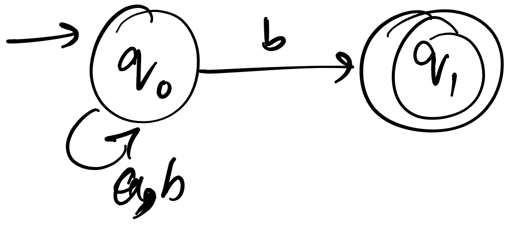
"In NFA for any input symbol there can be one or more then one transition or no transition"

Q $\Sigma = (a, b)$

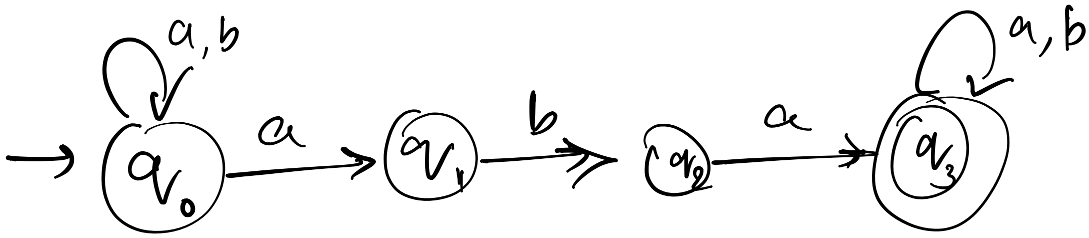
(i) Starts with a



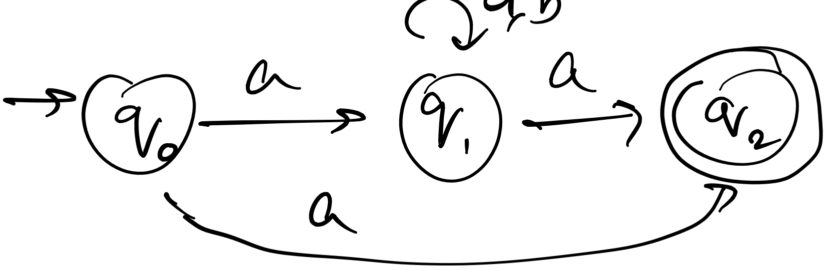
(ii) Ends with b



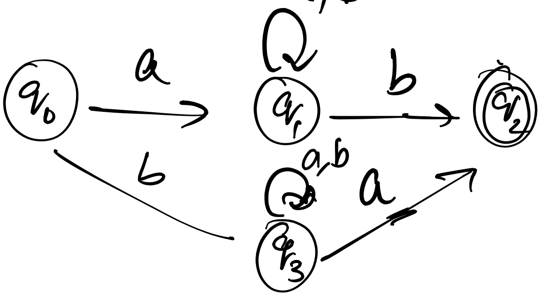
(iii) Contain substring aba



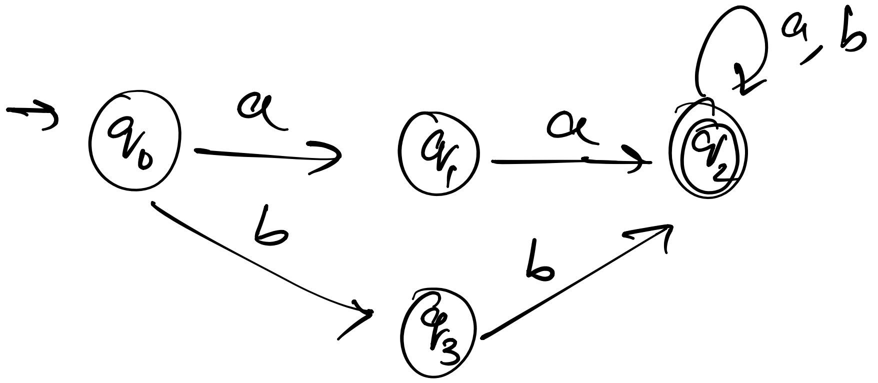
(iv) Starts and Ends with a



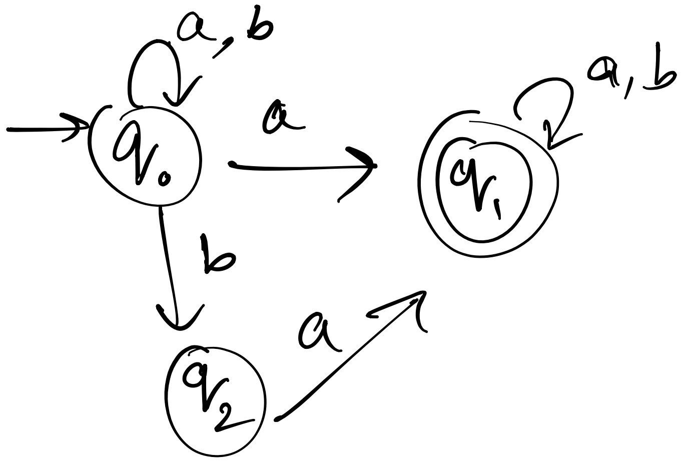
(v) Start and ends with different symbol



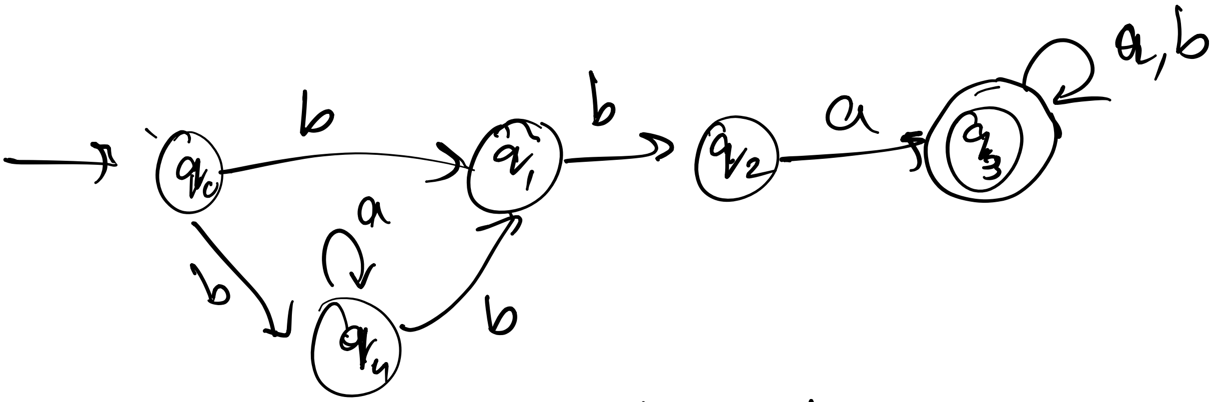
(vi) Start with aa or bb



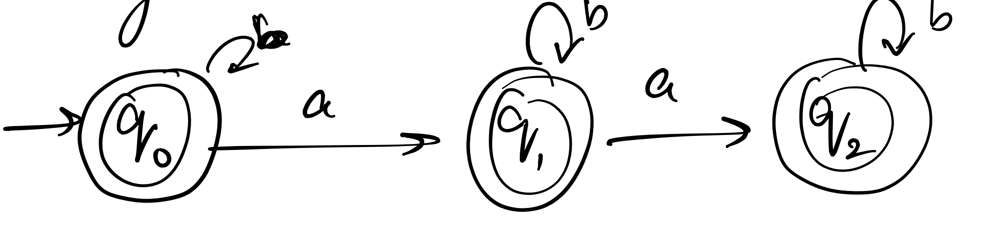
(vii) Starts with a or contain substring ba



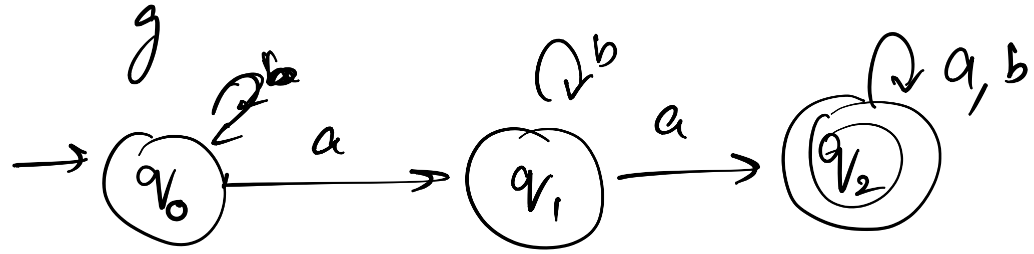
(viii) Starts with b and contain substring bba



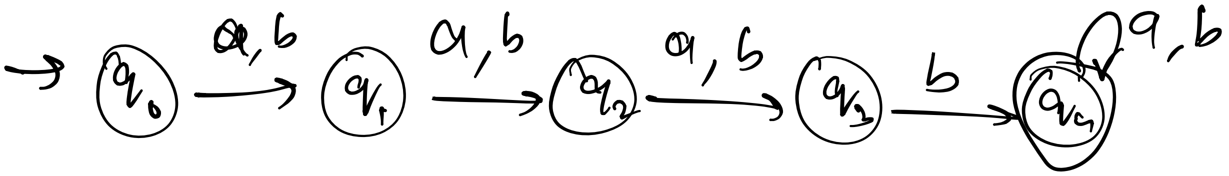
(ix) String that contain at most 2 a



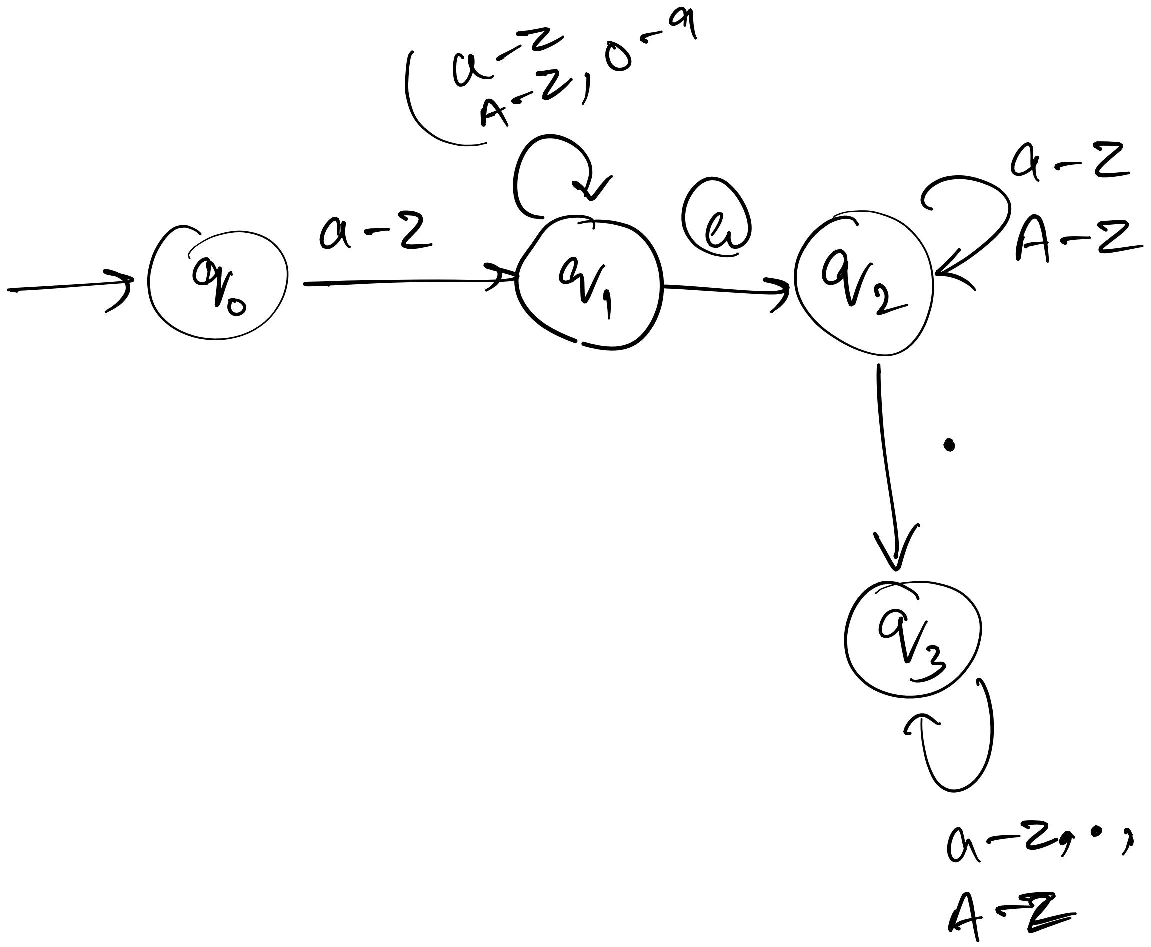
(x) String contain atleast 2 a



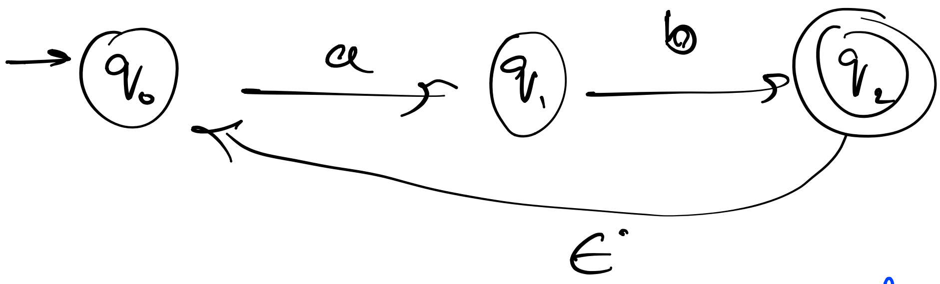
(xi) 4 th symbol should be b



(xii)



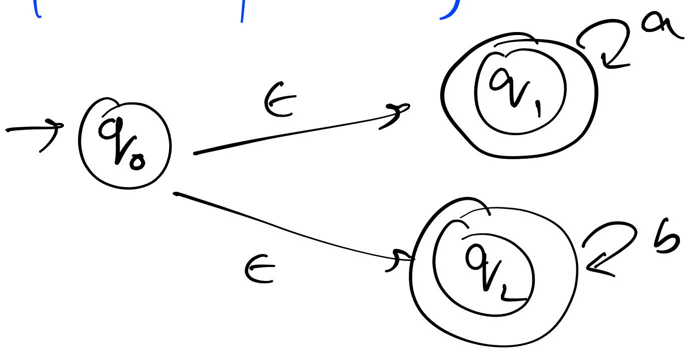
ϵ-NFA $\delta : Q \times \Sigma \cup \{\epsilon\} = 2^Q$



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

$L = \{(ab)^n \mid n \geq 1\}$

δ



$L = \{(a)^n, (b)^n \mid n \geq 0\}$

$L = \{(a)^n \cup (b)^n \mid n \geq 0\}$

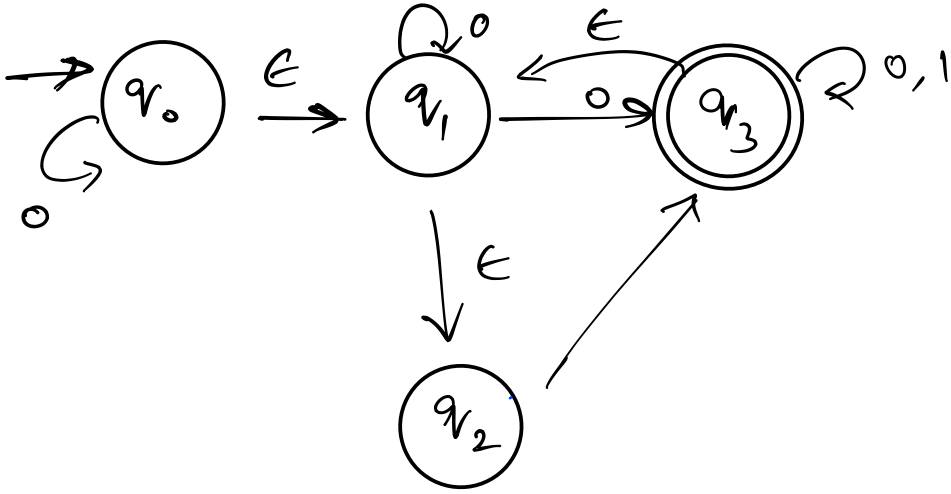
or

$L = \{(a)^n \mid n \geq 0\} \cup \{(b)^n \mid n \geq 0\}$

ϵ - closure

→ ϵ-closure(q_i)

All States that can be reached using ϵ-transition from q_i



$$\epsilon\text{-closure}(q_0) = \{q_0, q_1, q_2\}$$

$$\text{"} \quad (q_1) = \{q_1, q_2\}$$

$$\text{"} \quad (q_2) = \{q_2\}$$

$$\text{"} \quad (q_3) = \{q_3, q_1, q_2\}$$

Power : ε-NFA ~ NFA ~ DFA

$$\epsilon\text{-NFA} \xrightarrow{\text{(Thomson construcn)}} \text{NFA} \xrightarrow{\text{(Subset constructn)}} \text{DFA}$$

ϵ-NFA to NFA

\# Universally $\delta(q_i, \epsilon) = q_i$

ϵ-closure $(\phi) = \phi$

→ All states will be as it is just transition with change (Particularly ϵ transition will be removed)

δ → Transition fn^n of ϵ-NFA

δ' → Transition fn^n of NFA

\# General Formula (Notationally incorrect)

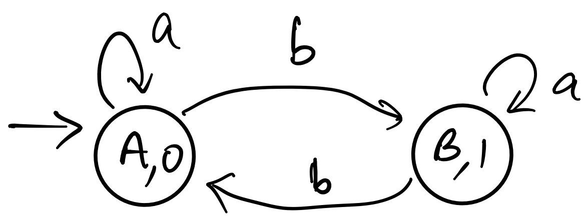
$$\boxed{\begin{aligned}&\delta'(A, 0)\\&= \epsilon\text{-closure}\left(\delta\left(\epsilon\text{ closure }(A), 0\right)\right)\end{aligned}}$$

\# FA with output

- Moore Machine
- Mealy Machine

Moore M. → • Outputs are generated based on states

• No final state



$\Sigma = a, b \mid \Delta = \{0, 1\} \mid Q = \{A, B\}$

δ : Trans^n Fun^n

λ : Output Fun^n

Ex w = bab
output = 0110

$M(Q, q_0, \Sigma, \Delta, \delta, \lambda)$

Q = Set of States

q_0 = Initial State

Σ = Input Symbols

Δ = Set of output Symbols

δ = Transition function

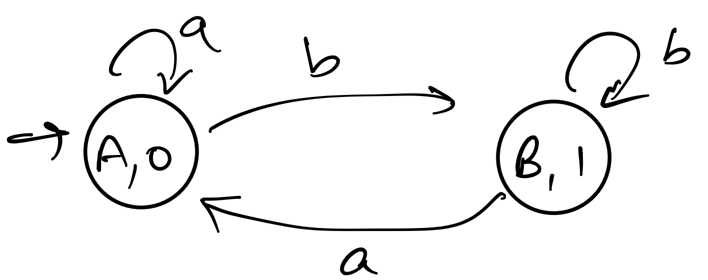
λ = output function.

$\delta : Q \times \Sigma = Q$ {Same as DFA}

$\lambda : Q \rightarrow \Delta$

Transition Table for moore machine

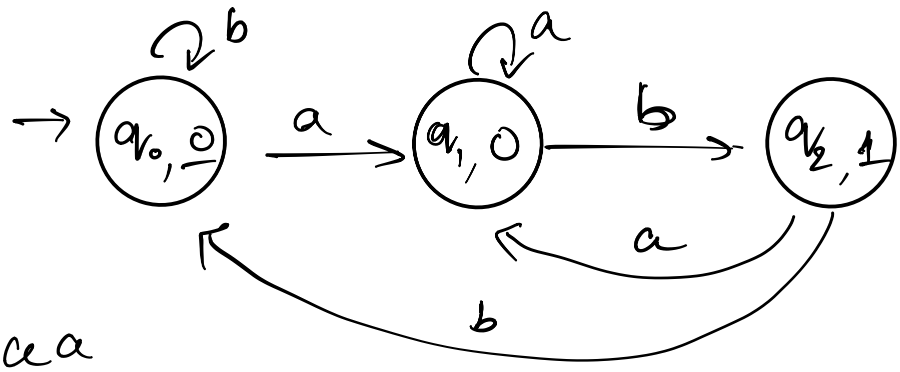
δ	a	b	λ
A	A	B	0
B	B	A	1



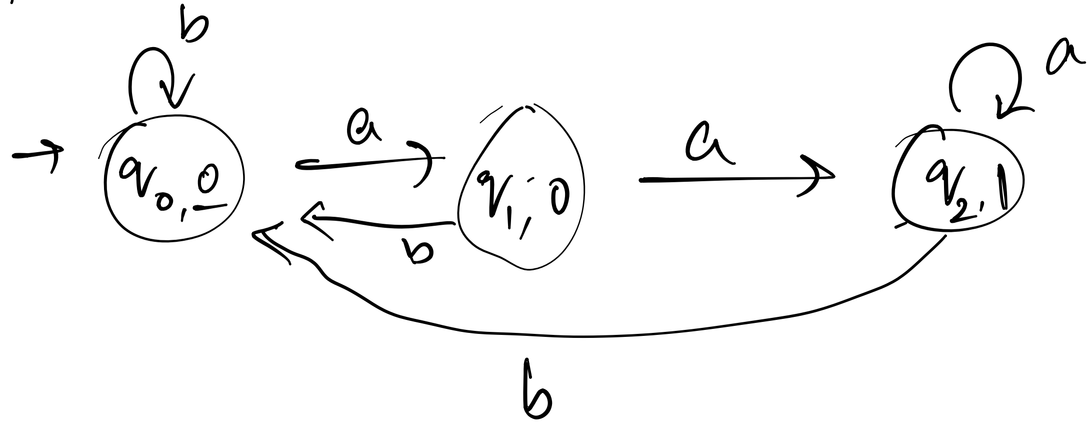
w = abab
output = 00101

w = bbab
= 01101

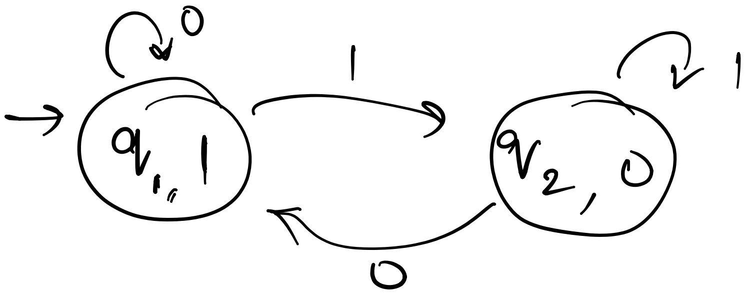
Q Construct a moore machine to expect ab substring



Q aa



Q Binary input and give binary output which gives us 1's compliment



Q Tacking input as binary string and output
=(input) mod 3
By taking assumption last letter of character