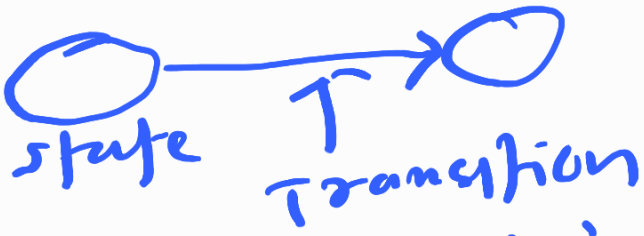
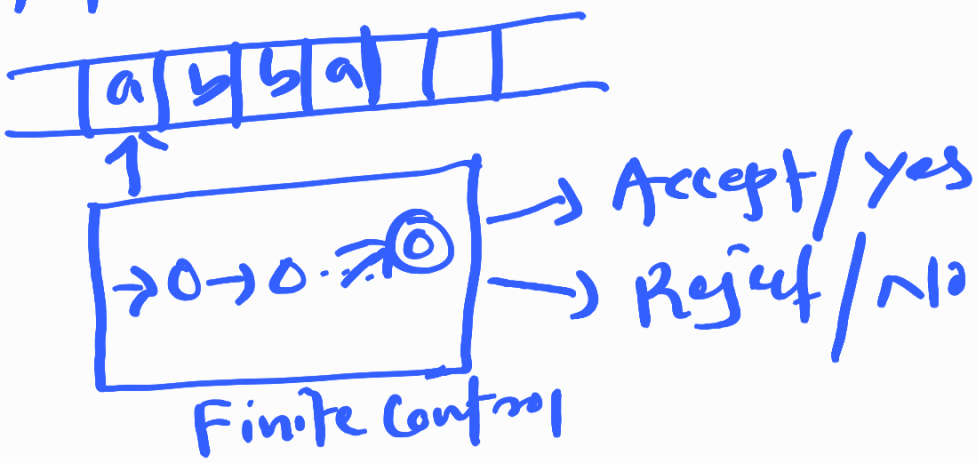
- FA
  - FA without o/p
    - DFA
    - NFA
    - ε-NFA
  - FA with o/p
    - Mealy Machine
    - Moore Machine

Finite Automata (FA) / Finite State Machine (FSM)

- FA is a mathematical/Abstract Model/ machine that consists of states and transitions.
- states is represented by circle and transition is represented by arrows.



- FA takes some string as input and this input goes through a finite number of states and may enter in to final state.



- In FA the no. of state is finite.
- FA is used to recognize strings/pattern.

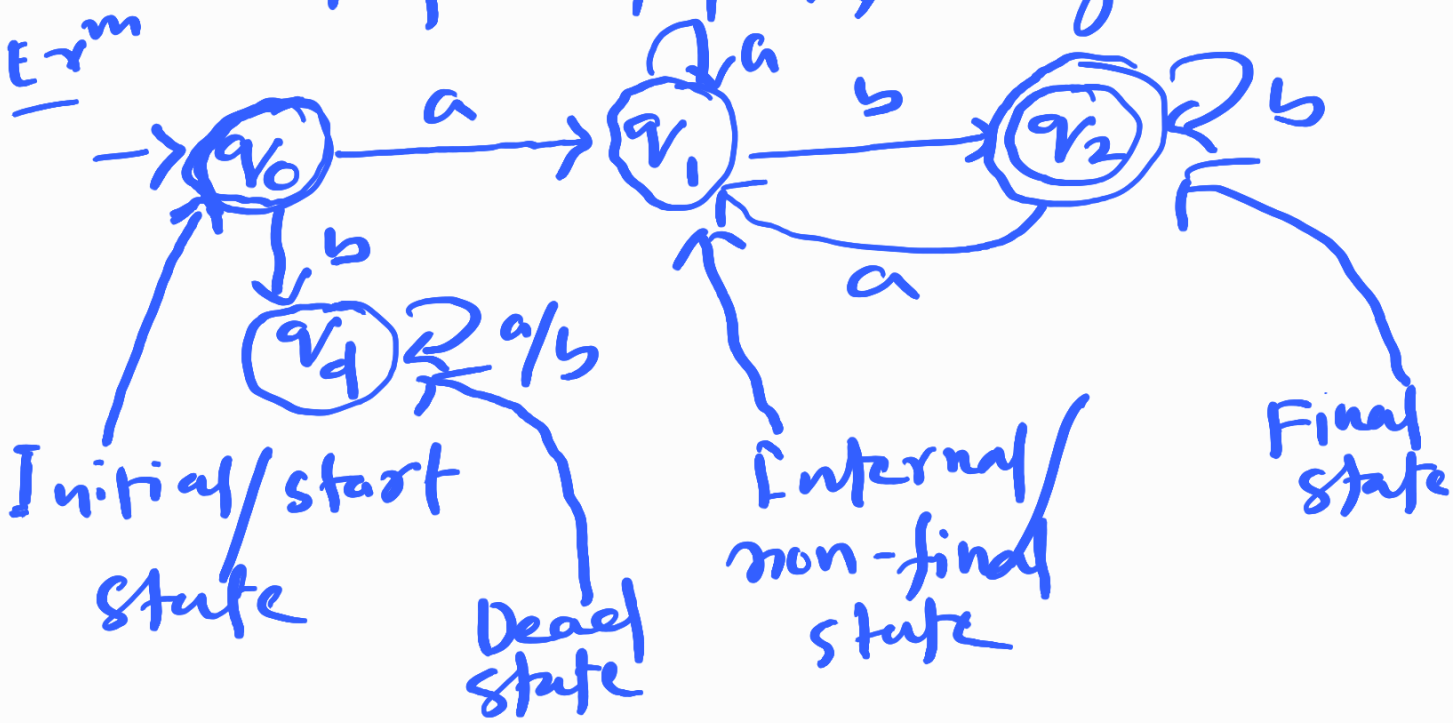
Description of FA

1. State transition Diagram
2. Transition table

1. State Transition Diagram (STD)

- STD is directed graph, can be constructed as follows.
  - There is a node for each state and is represented by circle.
  - There is a directed edge from one node to another node that define transition and is represented by using arrow.

Ex^m



② Transition Table (TT)

- TT consists of rows and column to represents a mathematical function in tabular form.
- Mathematical f^n / Transition f^n will take two argument i.e. state and input symbol and return a state.

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

state

state

input symbol

Input

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

states

state

Deterministics Finite Automata (DFA)

- Deterministics refers to the uniqueness of the computation.
- In DFA, for a particular input symbol machine may goes to one state only.

$$① \xrightarrow{a} ②$$

Formal Def^n of DFA

Mathematically, DFA can be defined by using 5-tuple $(Q, \Sigma, \delta, q_0, F)$ where

Q = finite set of states

Σ = finite set of inputs.

δ = Transition function

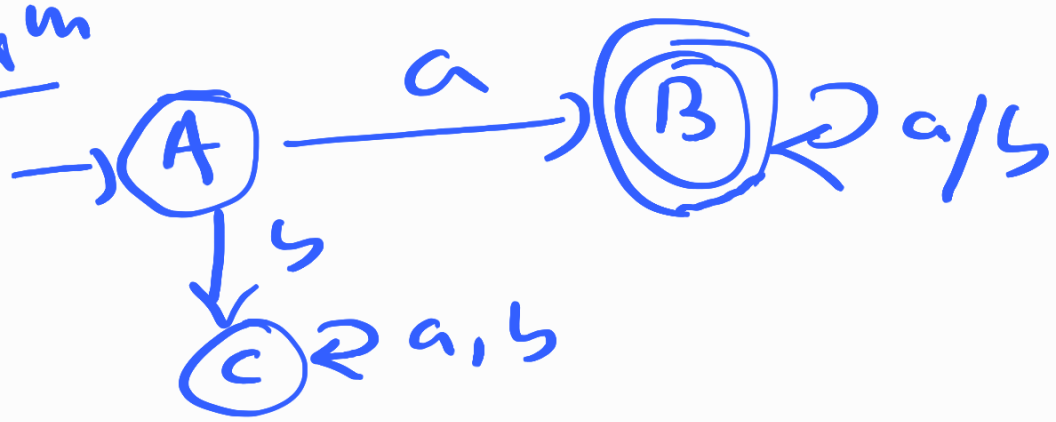
q_0 = Initial/start state

F = set of final state

Transition function (δ)

$$\delta : \underset{\text{state}}{Q} \times \underset{\text{input symbol}}{\Sigma} \rightarrow \underset{\text{state}}{Q}$$

Exm



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

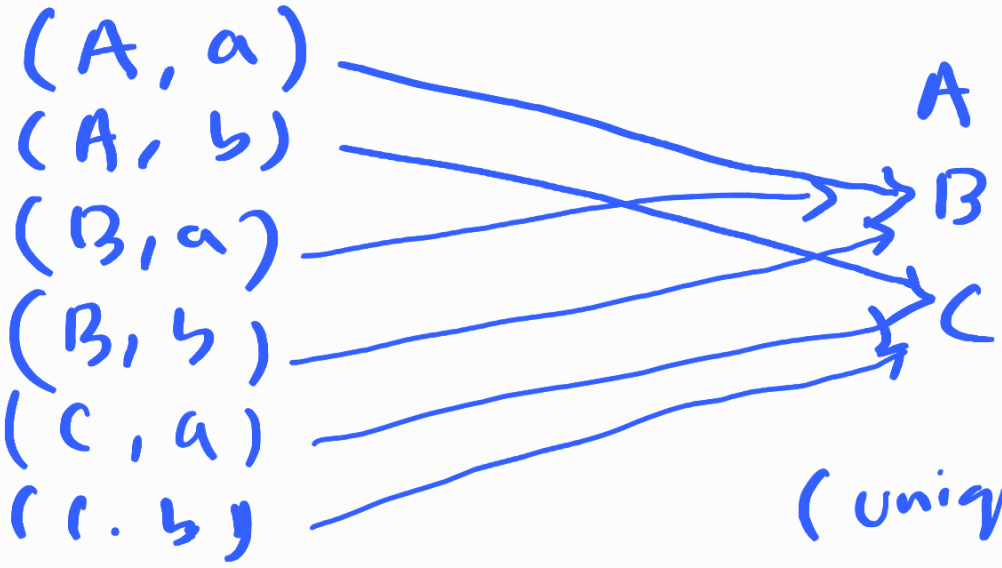
$\Sigma : \{a, b\}$

$q_0 : \{A\}$

$F : \{B\}$

$\delta : Q \times \Sigma \to Q$

$\{A, B, C\} \times \{a, b\} \to \{A, B, C\}$



(unique mapping)

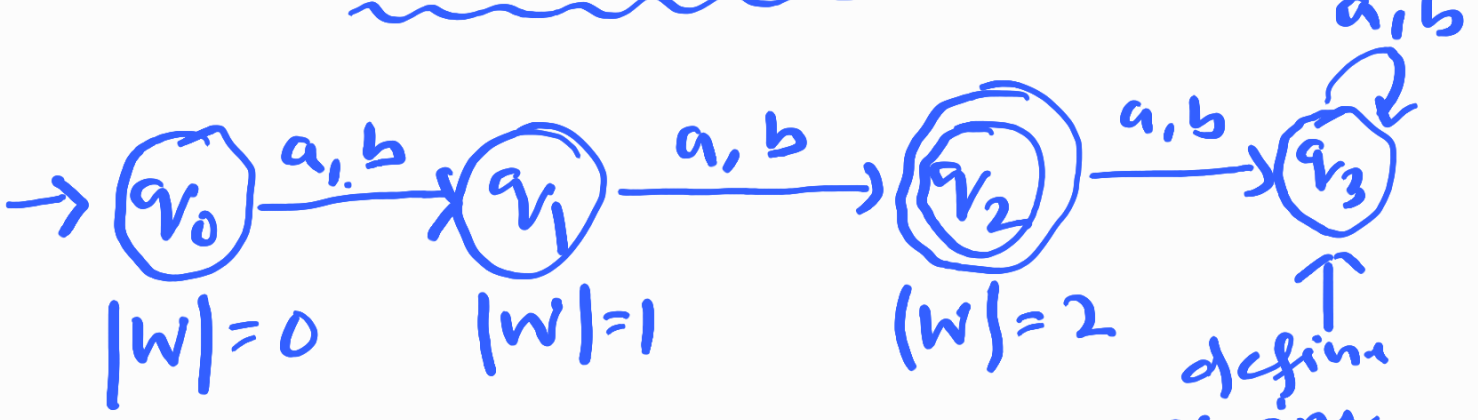
many to one

Design

Construct a DFA, that accept all string over $\Sigma = \{a, b\}$ of length 2.

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

$L = \{aa, ab, ba, bb\}$ $\boxed{w = \text{string}}$



define string length 3 and more.

String Accept

scan the entire string, if we reach to a final state from initial state.

$w = ab$

$\delta(q_0, ab) \Rightarrow \delta(q_1, b)$

$\Rightarrow q_2$ (final state) machine will accept "ab"

$w = bab$

$\delta(q_0, bab) \Rightarrow \delta(q_1, ab)$

$\Rightarrow \delta(q_2, b)$

$\Rightarrow q_3$ (non-final)

(Reject "bab")

Language Accept

- A DFA is said to accept a language if all the string in the language are accepted and all the string not in the language are rejected.

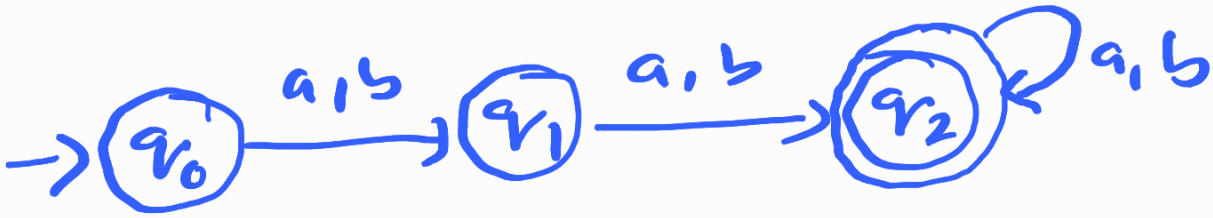
Assignment

① construct a DFA, that accept all string over $\Sigma = \{a, b\}$ having string length atleast '2'.

② At most '2'.

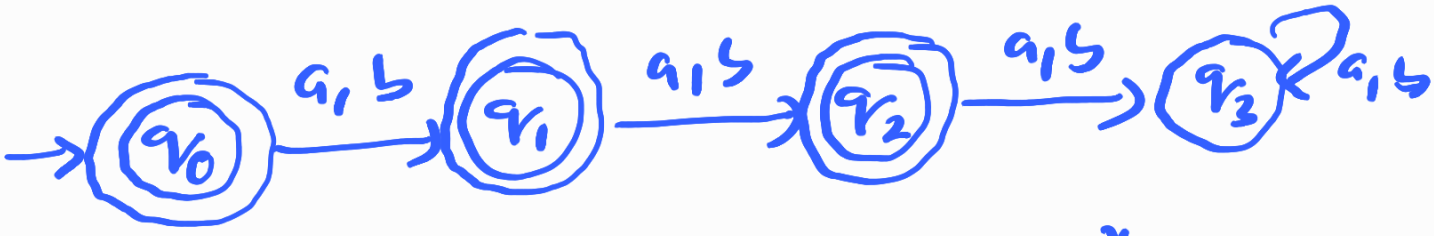
A1 $\Sigma = \{a, b\}$

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



A2

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



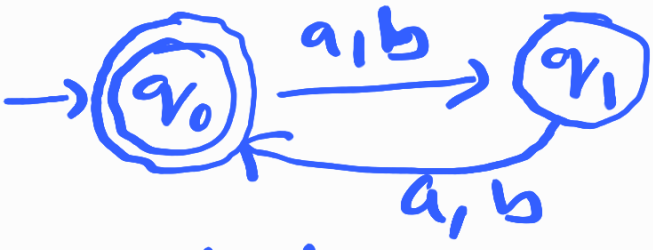
Construct a DFA, $w \in (a,b)^*$ and length of string is even i.e

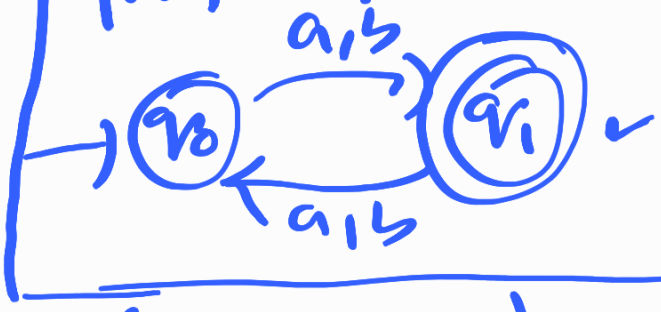
$|w| \bmod 2 = 0$

$\boxed{w \in (a,b)^*}$ $\Sigma = \{a,b\}$

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

$|w| \bmod 2 = 1$ (odd)





w: aabab

$\delta(q_0, aabab) \Rightarrow \delta(q_1, abab)$

$\Rightarrow \delta(q_0, bab)$

$\Rightarrow \delta(q_1, ab)$

$\Rightarrow \delta(q_0, b)$

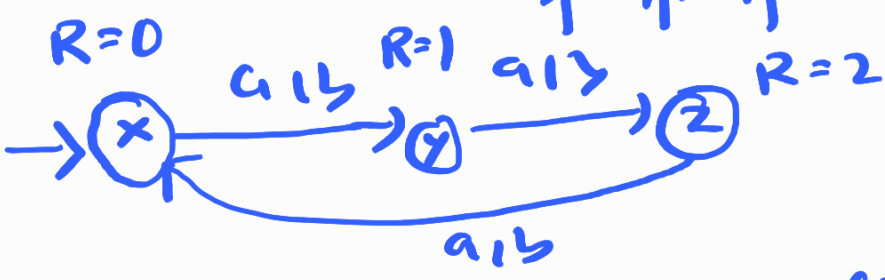
$\Rightarrow q_1$ (non final state)

(*) $|w| \bmod 3 = ?$

↳ modular arithmetic

possible remainder

$\{0, 1, 2\}$



If $|w| \bmod 3 = 0$ then final state = x

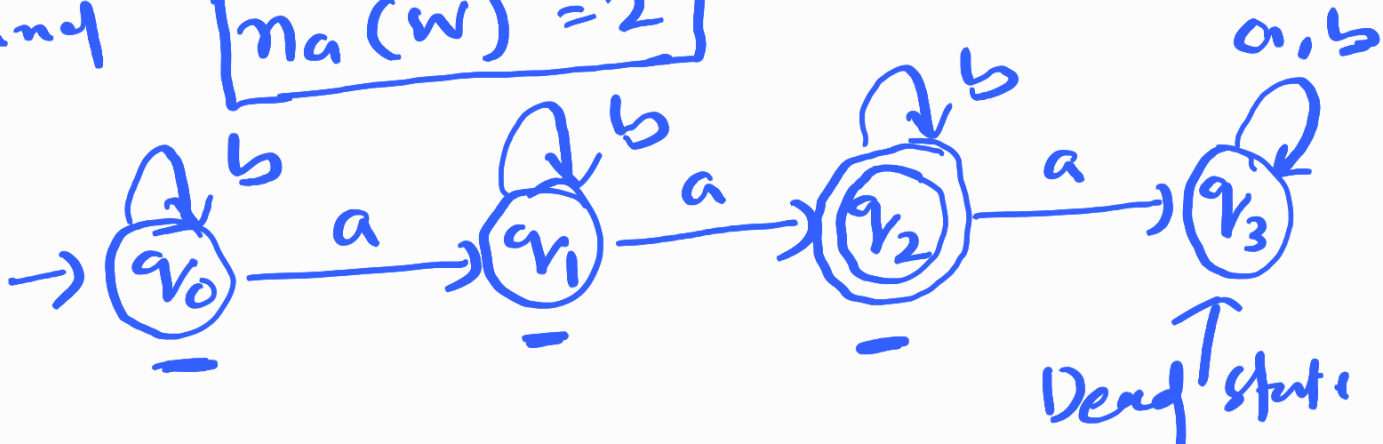
$|w| \bmod 3 = 1$ " " " = y

$|w| \bmod 3 = 2$ " " " = z

$\|w\|$	mod 3 =	Result
0		0
1		1
2		2
3	→	0
4	→	1
5	→	2
6	→	0

final		
x	y	z
0	1	2
3	4	5
6	7	8
9	10	11
12	⋮	⋮
⋮	⋮	⋮

(*) Design a DFA, where $w \in (a,b)^*$ and $\boxed{n_a(w) = 2}$



(*) Design a DFA, where $w \in (a,b)^*$

and $\begin{cases} n_a(w) = \underline{0 \bmod 2} \checkmark \\ n_b(w) = \underline{1 \bmod 3} \checkmark \end{cases}$

no. of a's = $\{0, 2, 4, 6, 8 \cdots\}$

no. of b's = $\{1, 4, 7, 10, 13, 16 \cdots\}$

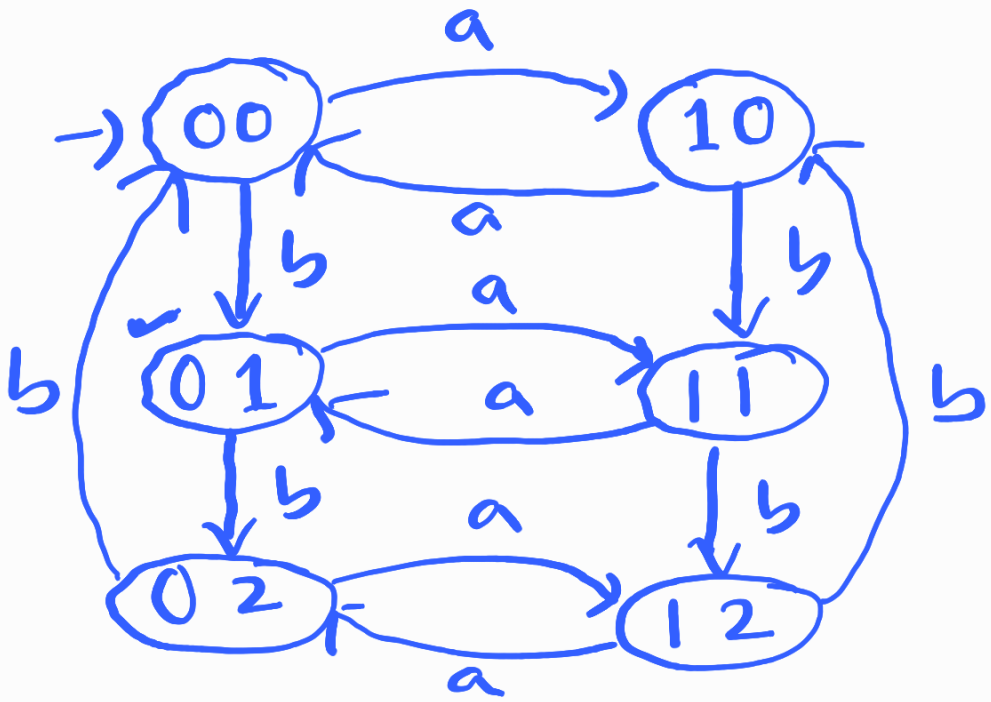
Mod 2 : $\{\underline{0}, 1\}$

mod 3 : $\{\underline{0}, 1, 2\}$

$n_a(w)$	$n_b(w)$
0	0
0	1
0	2
1	0
1	1
1	2

$\rightarrow$ no. of a is divisible by 2 and no. of 'b' is divisible by 3

$\rightarrow$ no. of 'a' mod 2 = 1 and no. of b mod 3 = 2

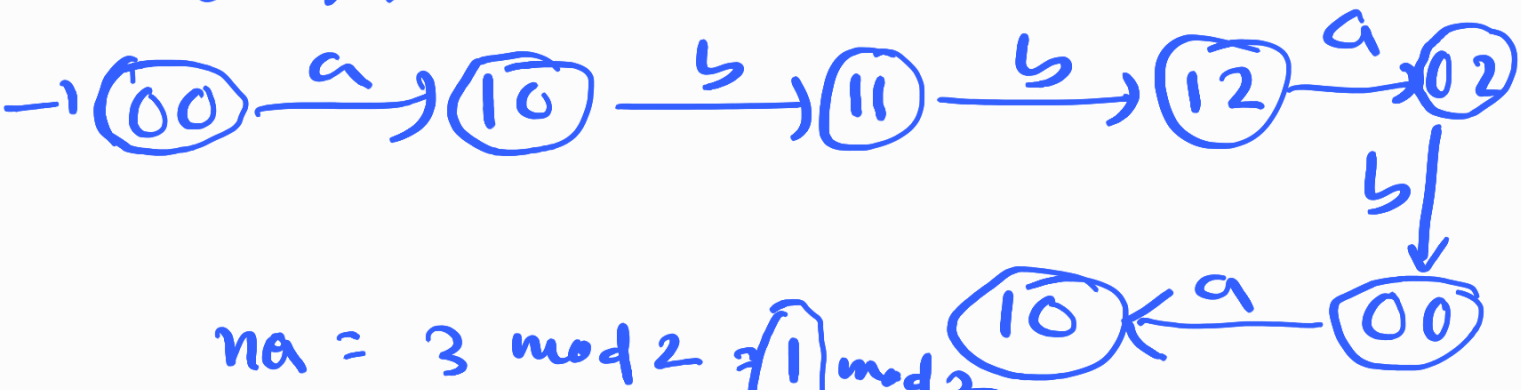


In horizontal design, we are counting no. of 'a' and vertical design, we are counting no. of 'b'.

$$\left.\begin{cases} n_a(w) = 0 \bmod 2 \\ n_b(w) = 0 \bmod 3 \end{cases}\right\} \text{final } (00)$$

$$\left.\begin{cases} n_a(w) = 1 \bmod 2 \\ n_b(w) = 2 \bmod 3 \end{cases}\right\} (12)$$

abbabba



$n_a = 3 \bmod 2 = 1 \bmod 2$

$n_b = 3 \bmod 3 = 0 \bmod 3$

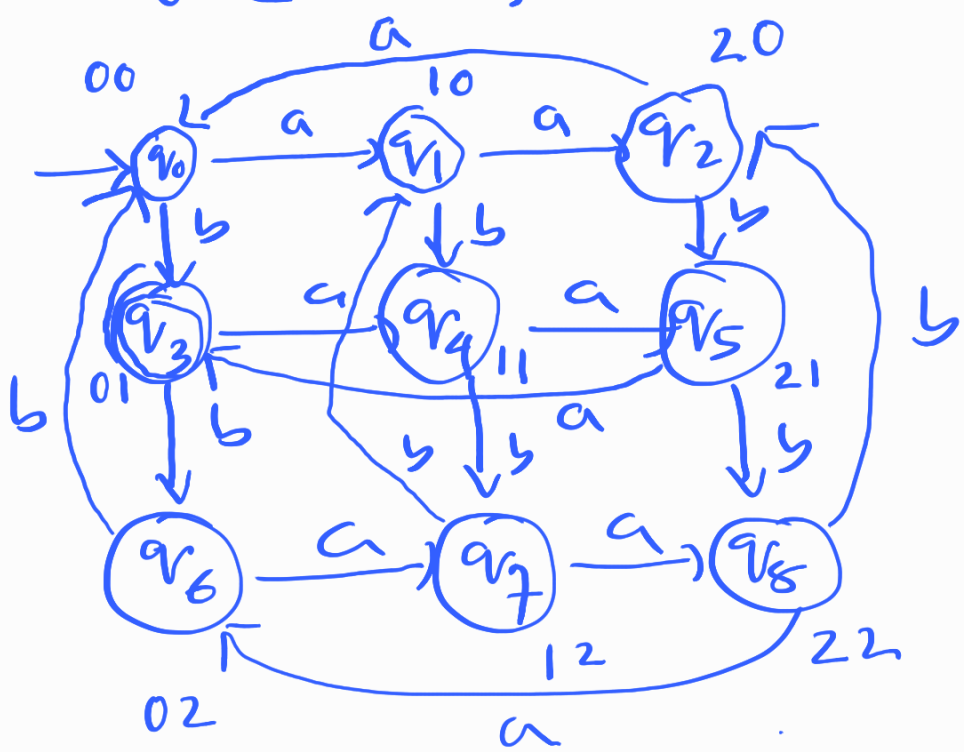
Assignment

(*) Design DFA, where $w \in (a,b)^*$ and

$$n_a(w) \cong 0 \bmod 3$$
$$n_b(w) \cong 1 \bmod 3$$

a $\boxed{Z_3 = \{\underline{0}, \underline{1}, \underline{2}\}}$ $b = Z_3 = \{0,1,2\}$

$w \in (a,b)^*$

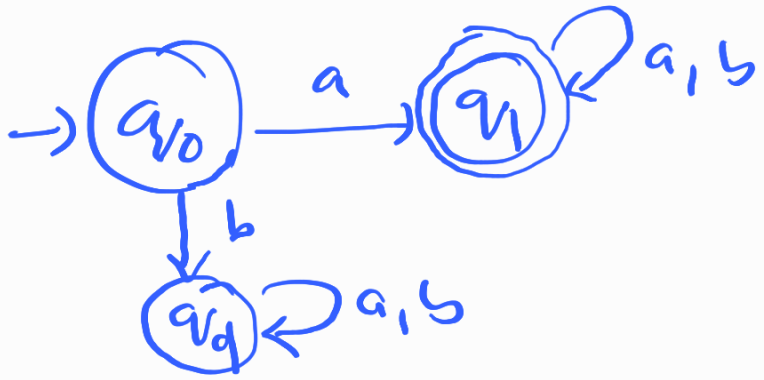


2/6/2021

(*) Design a DFA over $\Sigma = \{a,b\}$ that accepts set of all string start with 'a'.

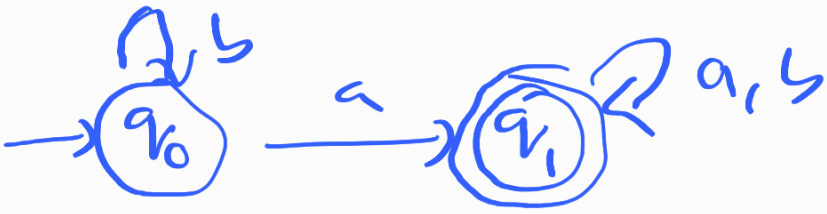
$\Sigma = \{a,b\}$

$L = \{\underline{a}, aa, ab, aaa, aba \ldots\}$



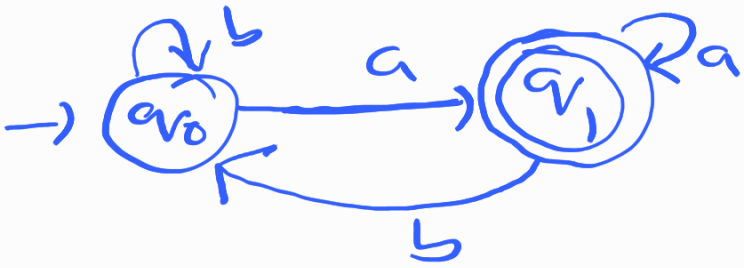
* DFA, $\Sigma = \{a, b\}$ that accepts set of all strings that contain 'a'.

$L = \{a, ab, ba, aa, aab \ldots\}$



* Ending with 'a' over $\Sigma = \{a, b\}$

$L = \{a, aa, ba, baa \ldots\}$



* $L = \{\epsilon, a, ba, bba, aa \ldots\}$

not ending with 'b'

previous Exm i.e. $\boxed{\text{ending with 'a'}}$

$\Downarrow$ operation (Complement)

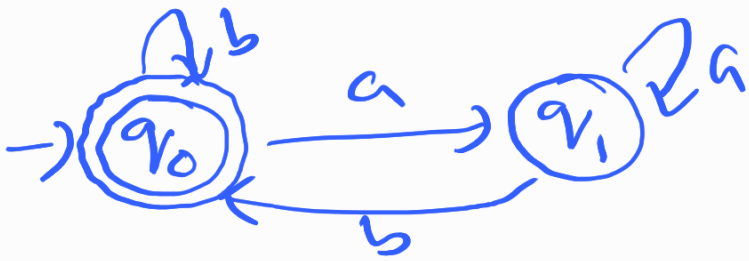
$\boxed{\text{not ending with 'a'}}$

<u>complement operation</u> we can apply with finite automata (FA), by changing final state to non-final state and non-final to final state.

<u>ending with 'a'</u>

$\Downarrow$ Complement

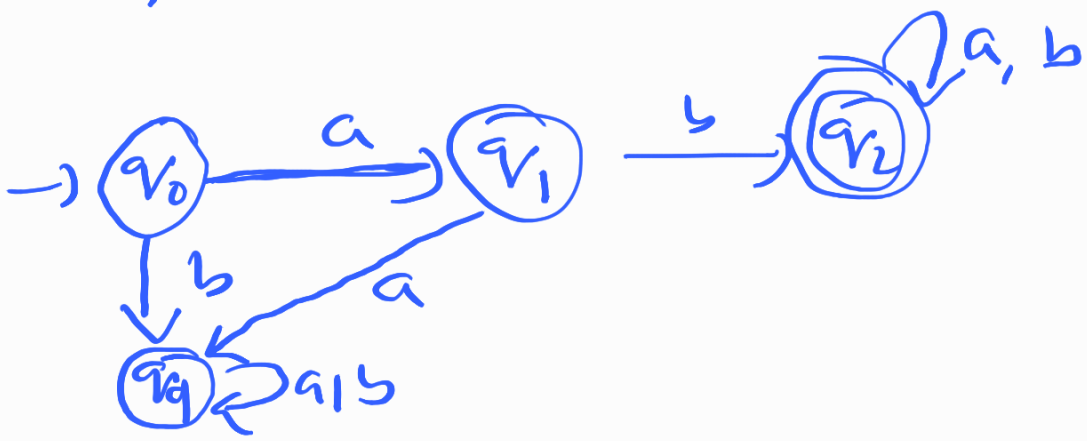
<u>not ending with 'a'</u>



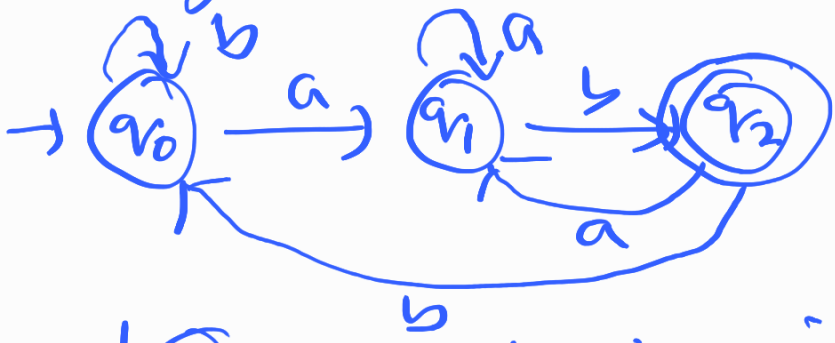
$L = \{\epsilon, b, bb, aab, aabb \ldots\}$

(*) start with 'ab' over $\Sigma = \{a, b\}$

$L = \{ ab, aba, abb, abaa \ldots \}$



(*) ending with 'ab' $\Sigma = \{a, b\}$



<u>Assignment</u> (1) – containing 'ab' as substring over $\Sigma = \{a, b\}$

(*) strings start with 'a' and end with 'b', over $\Sigma = \{a, b\}$

$L = \{ ab, aab, abb \ldots \}$

start with 'a'	end with 'b'
$L_1 = \{ a, aa, ab, abb \ldots \}$	$L_2 = \{ b, ab, abb \ldots \}$

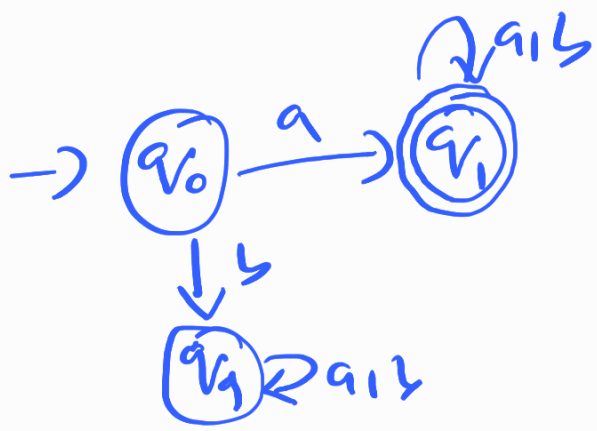
If we concatenate the languages of these two statements then we get language L.

$$L = L_1 \circ L_2 \quad \text{concatenation.}$$

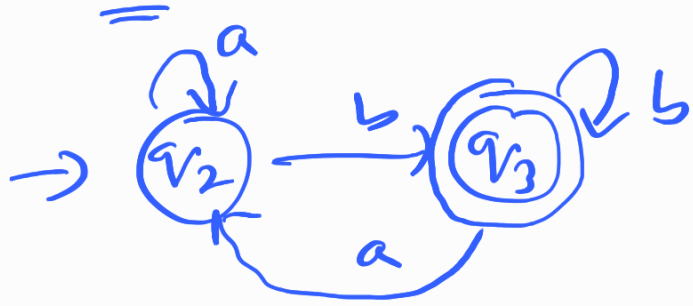
$$= \{a, aa, ab, \ldots\} \circ \{b, ab, abb \ldots\}$$

$$= \{ab, aab, aabb, aab, aaab \ldots\}$$

$\underline{\underline{L_1}} =$



$\underline{\underline{L_2}}$



Concatenation by merging final state of first m/c with initial state of second m/c.

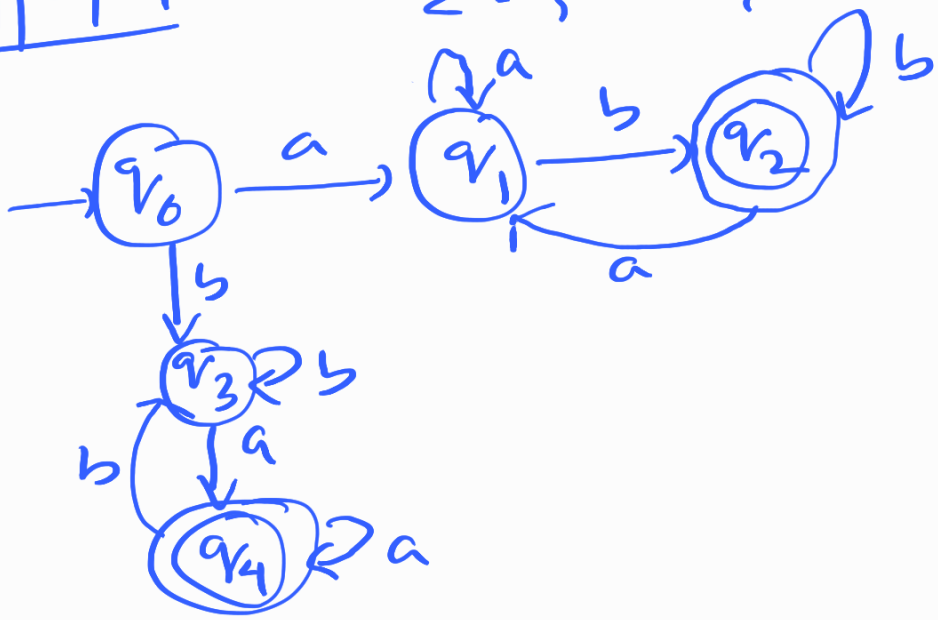
Assignment

② string start and end with different symbols.

③ string start and end with same symbols.

9/6/21

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



Union

② string start and end with different symbols.

(i) start with a and end with b

(ii) start with b and end with a

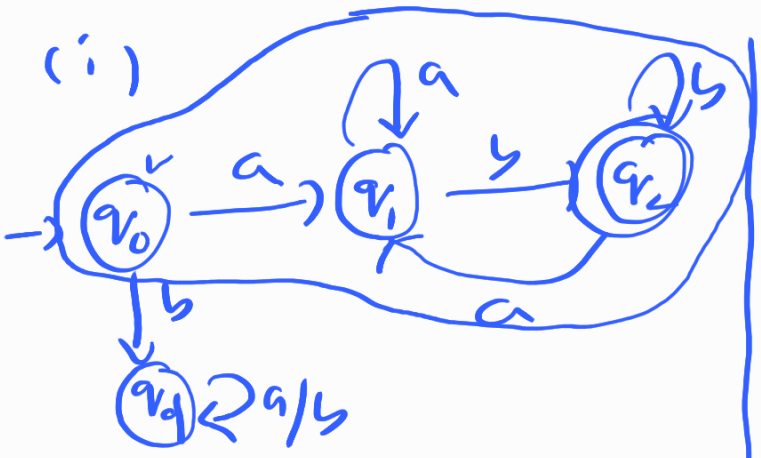
$L_1 = \{ ab, aab, abb \ldots \}$

$L_2 = \{ ba, bba, baa \ldots \}$

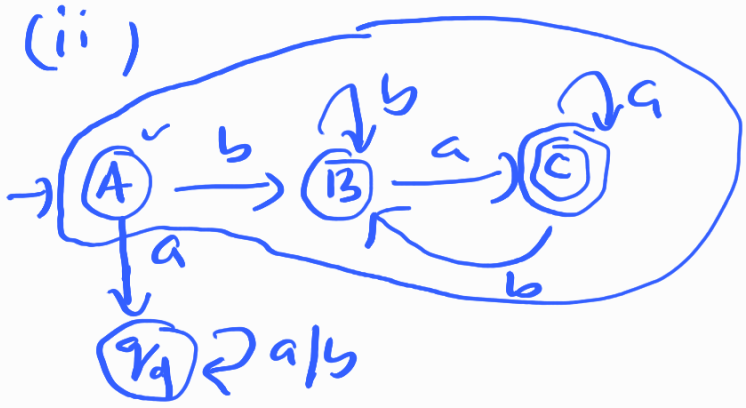
$L_1 \cdot L_2 = \{ abba, abbba, \ldots$

$L_1 \cup L_2 = \{ ab, ba, aab, bba, \ldots \}$

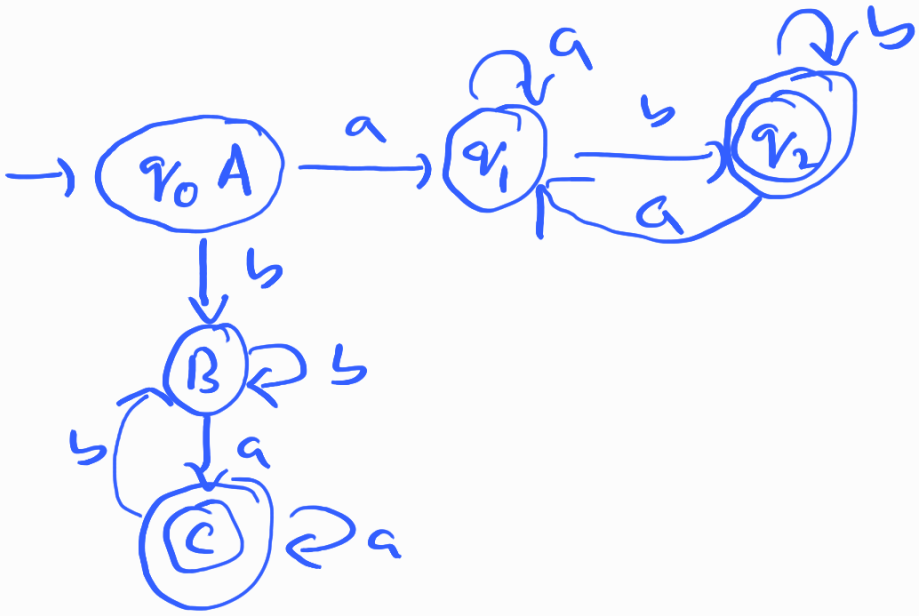
(i)



(ii)

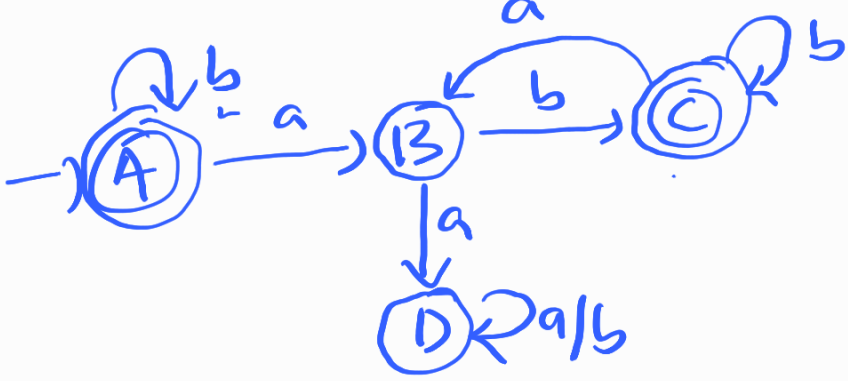


After union of above two machines the resultant machine is given below.



(*) Design a DFA in which $w \in (a,b)^*$. and every 'a' should be followed by 'b'.

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



(A1) Any 'a' should never be followed by 'b'.

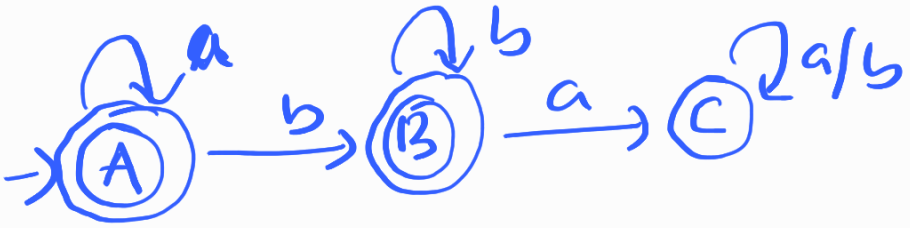
(A2) Every 'a' should be followed by 'bb'.

(*) Design a DFA that will accept the following language.

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

$$= \{ \epsilon, a, b, aa, bb, ab, aab \ldots \}$$

→ Any number of a followed by any no. of b.



(A3) Design a DFA for

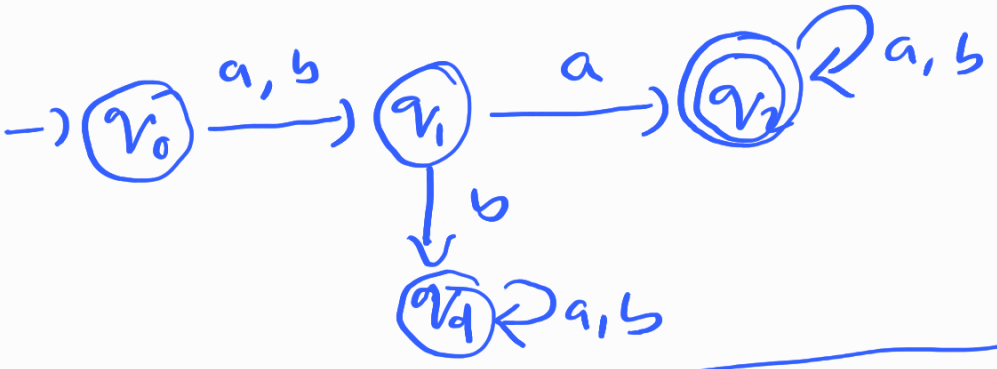
$$L = \{ a^n b^m \mid n, m \geq 1 \}$$

(A4) Design a DFA for

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

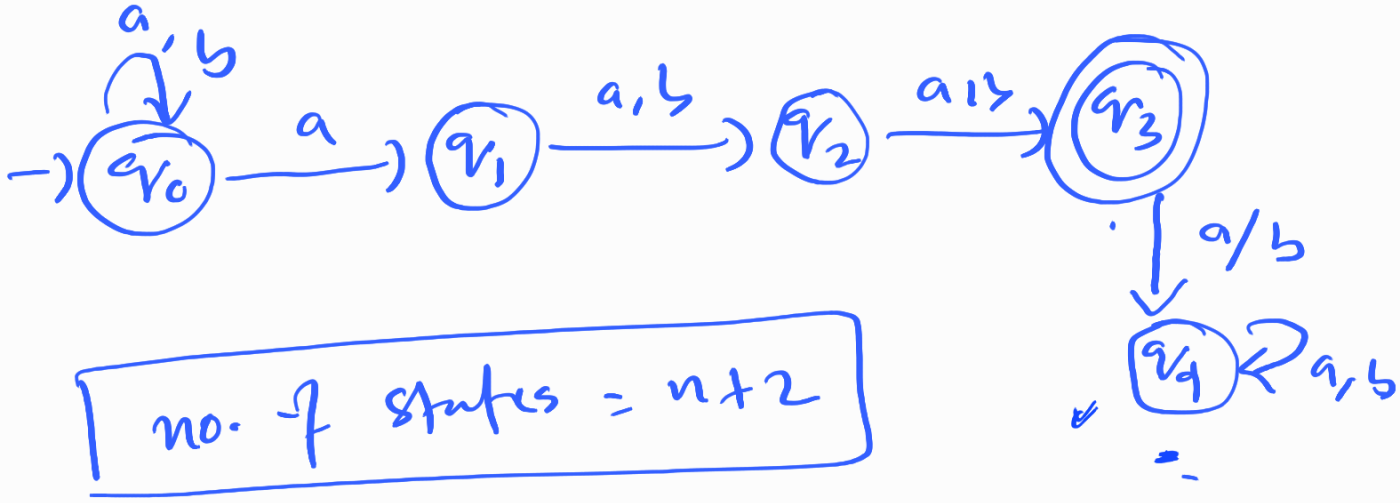
(*) Design a DFA over $\Sigma = \{a, b\}$ and $w \in (a,b)^*$ where 2nd symbol in 'w' from LHS should be 'a'.

$$L = \{ aa, ba, aab, aabb, baab \ldots \}$$



$\boxed{\text{no. of states} = \text{symbol position}(n^{th}) + 2}$

(*) 3rd symbol from RHS is 'a' (NFA)

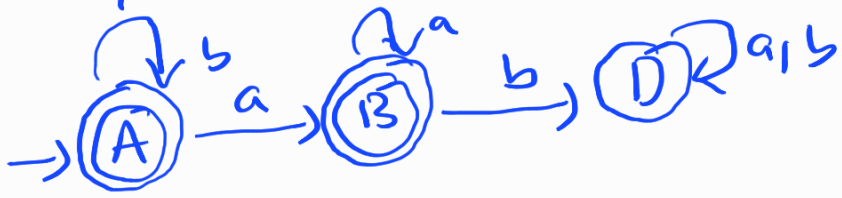


$\boxed{\text{no. of states} = n+2}$

7/6/21

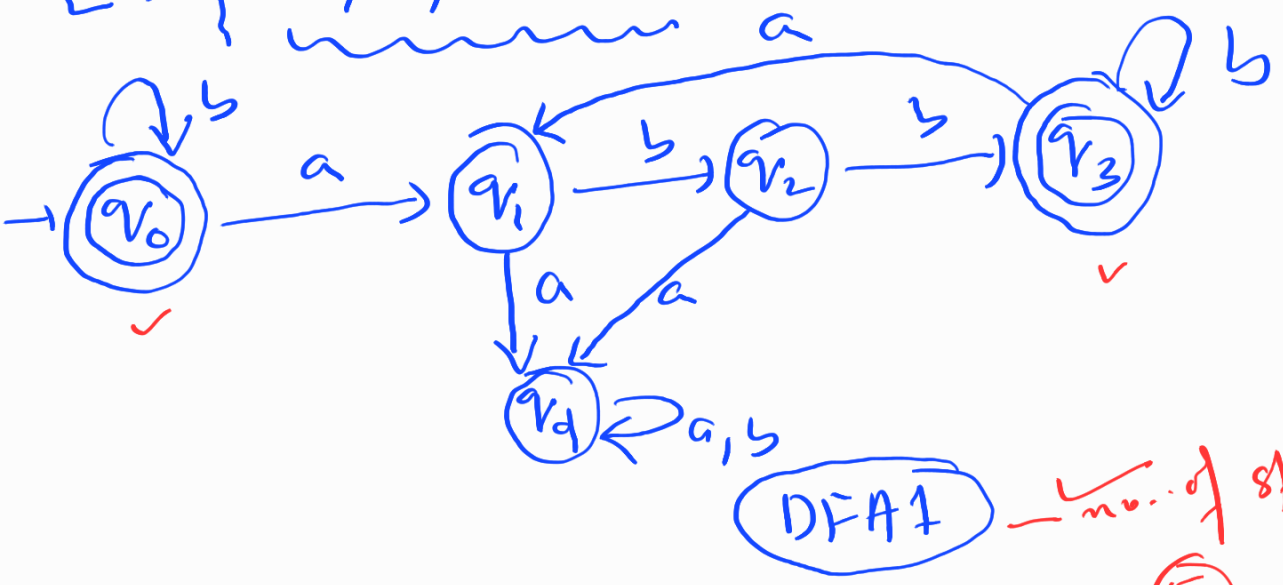
(A1) Any 'a' should never be followed by 'b'

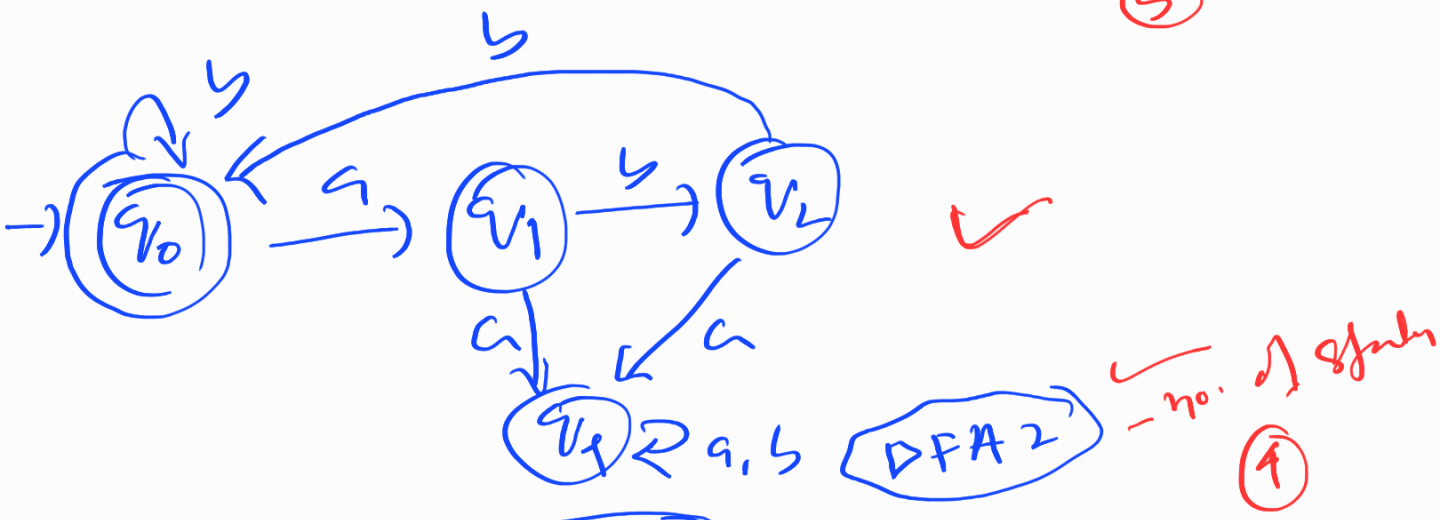
$L = \{ \epsilon, a, aa, aa, b, bb, ba \ldots \}$



(A2) Every 'a' should be followed by "bb".

$L = \{ \underbrace{\epsilon, b, bb \ldots, abb}_{a}, bbabb \ldots \}$





w = abba in DFA2

$\delta(q_0, abba) \Rightarrow \delta(q_1, bba)$

$\Rightarrow \delta(q_2, ba)$

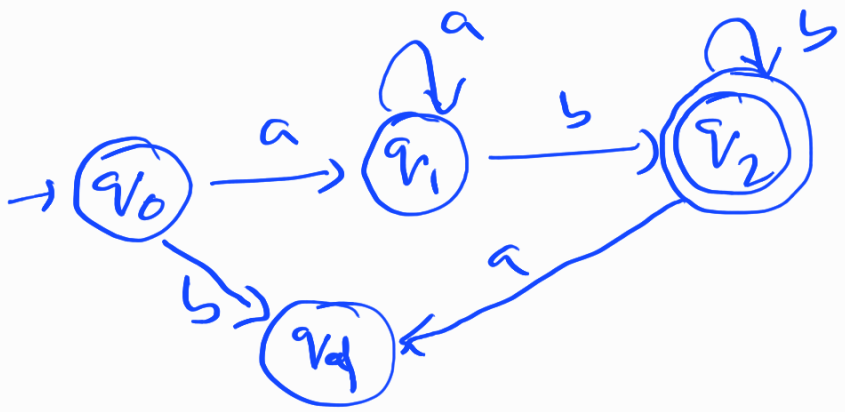
$\Rightarrow \delta(q_0, a)$

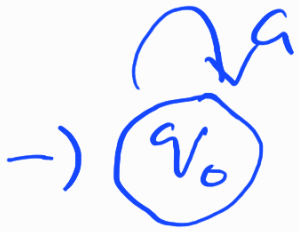
$\Rightarrow q_1 \notin q_f$

so, this string 'w' not accept by DFA2.

(A3) $L = \{a^n b^m \mid n, m \geq 1\}$

$L = \{ab, aab, aabb \ldots\}$





any no. of 'a' including zero no.

(A4) $L = \{a^n b^m c^l \mid n, m, l \geq 1\}$

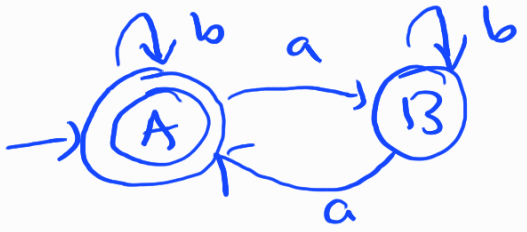
<u>operations</u>

- → Complement
- → Concatenation
- → Union
- → cross product
- → Reversal

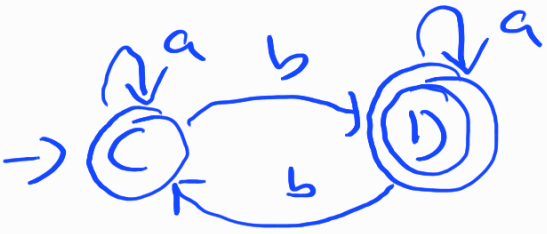
These operations we can apply in DFA

Cross product

DFA1: no. of a's are even $\Sigma = \{a, b\}$



DFA2: no. of b's are odd $\Sigma = \{a, b\}$



DFA1 X DFA2

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

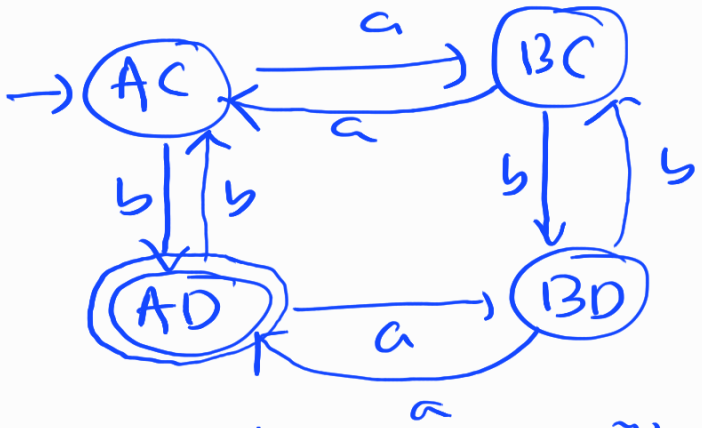
$= \{\underset{\uparrow}{AC}, \underset{\uparrow}{AD}, \underset{\uparrow}{BC}, \underset{\uparrow}{BD}\}$

$\begin{matrix}A\\C\end{matrix} \xrightarrow{a} \begin{matrix}B\\C\end{matrix}$ $\qquad$ $\begin{matrix}A\\C\end{matrix} \xrightarrow{b} \begin{matrix}A\\D\end{matrix}$

$\begin{matrix}A\\D\end{matrix} \xrightarrow{a} \begin{matrix}B\\D\end{matrix}$ $\qquad$ $\begin{matrix}A\\D\end{matrix} \xrightarrow{b} \begin{matrix}A\\C\end{matrix}$

$\begin{matrix}B\\C\end{matrix} \xrightarrow{a} \begin{matrix}A\\C\end{matrix}$ $\qquad$ $\begin{matrix}B\\C\end{matrix} \xrightarrow{b} \begin{matrix}B\\D\end{matrix}$

$\begin{matrix}B\\D\end{matrix} \xrightarrow{a} \begin{matrix}A\\D\end{matrix}$ $\qquad$ $\begin{matrix}B\\D\end{matrix} \xrightarrow{b} \begin{matrix}B\\C\end{matrix}$



Resultant DFA will accept set of all string that has no. of a's are even and no. of b's are odd.

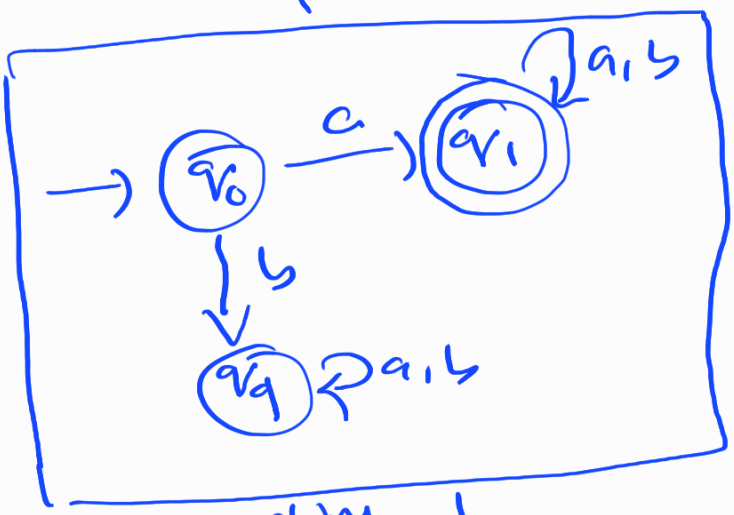
<u>Reversal</u>

$L = \{$ set of string start with a $\}$

$= \{ a, aa, ab, aaa, aab \ldots \}$

$L^R = \{ a, aa, ba, aaa, baa \ldots \}$

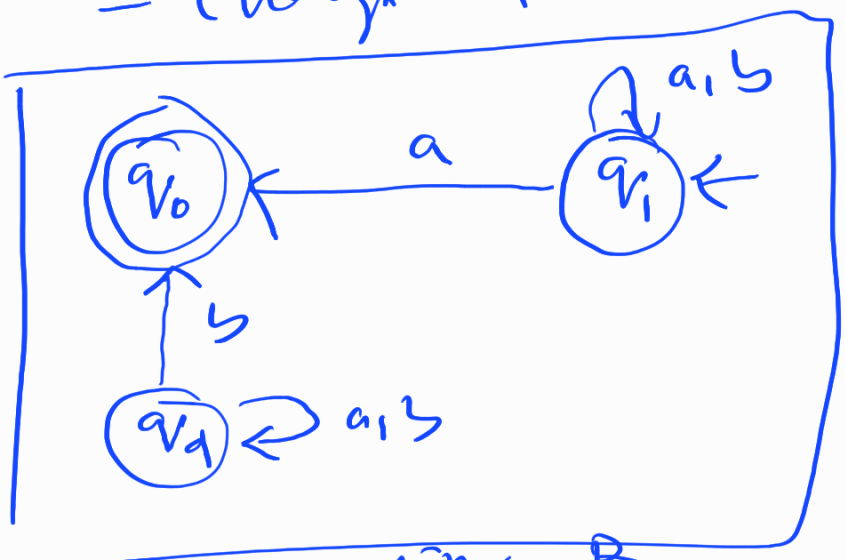
$= \{$ end with a $\}$



Accepting L

procedure

- change initial to final state.
- change final to initial state.
- change direction of transitions.



Accepting L^R

In case of Reversal, the resultant machine may be a DFA or NFA.