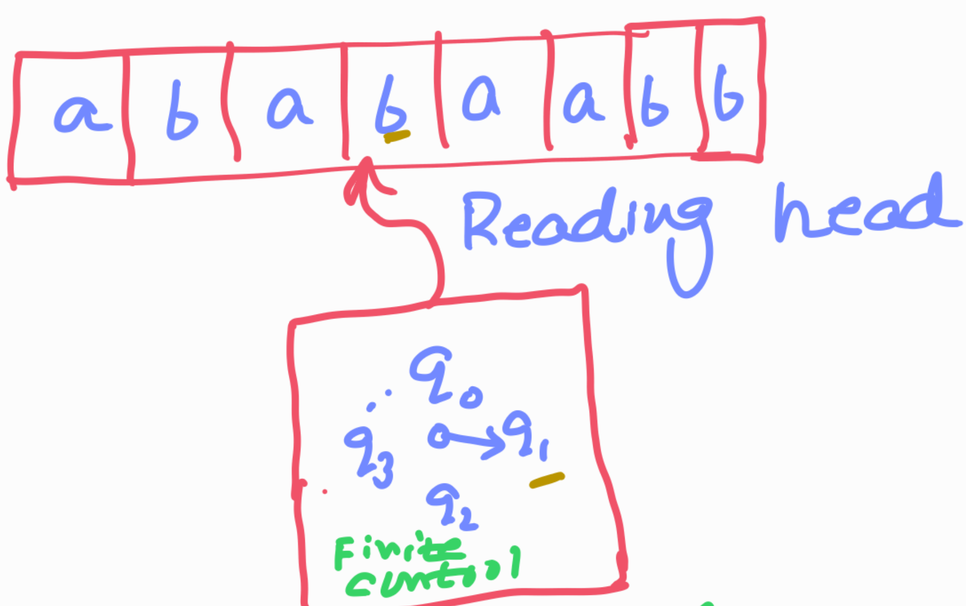
06-01-2021

Finite Automata

A finite Automation consists of set of states & a set of transitions from state to state that occur on input symbols choosen from an alphabet Σ.

Input Tape



Input string is stored in input tape, which is divided into cells.

Finite control can sense what symbol is written at any position on the input tape by means of a movable reading head

Initially the reading head is placed at the leftmost square & finite control is set at the designated initial state.

At regular intervals the automation reads one symbol from the input tape & then enters a new state that depends only on the current state & symbol just read (DFA)

Finite Automata

Deterministic Finite Automata (DFA)

Non-Deterministic Finite Automata (NFA)

In D.F.A the machine control will transform to a single state only, in NFA the machine control will go to one or more states.

Eventually the reading head reaches the end of the input string; the string is accepted if the machine halts at one of the set of final states.

Mathematically -

A DFA is a quintuple (5-tuple) system.

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

$Q \rightarrow$ Finite set of states

$\Sigma \rightarrow$ It is an alphabet

$q_0 \in Q$ is the initial state

$F \subseteq Q$ is the set of final states;

$\delta \rightarrow$ The transition function from $Q \times \Sigma \rightarrow Q$

If M is the machine in state $q \in Q$ and the symbol read from tape is $a \in \Sigma$, then $\delta(q, a) \rightarrow Q$ is the uniquely determined state to which Q passes. (DFA)

Configuration of a DFA $(Q, \Sigma, \delta, q_0, F)$ is any element of

$\boxed{Q \times \Sigma^*}$ $(q_0, abab)$

The binary relation $\vdash_M$ holds between two configurations of M iff the machine can pass from one to other as a result of single move.

If (q, w) & (q', w') are two configurations of M, then

$\boxed{(q, w) \vdash_M (q', w')}$ iff $w = aw'$ for symbol $a \in \Sigma$ & $\delta(q, a) = q'$.

In this case we say that (q, w) yields (q', w') in one step

A configuration of the form (q, ϵ) signifies that M has consumed all its input.

$$(q, w) \vdash_M^* (q', w')$$ read as (q, w) yields (q', w') after some number, possibly zero, of steps.

A string $w \in \Sigma^*$ is said to be accepted by M iff there is a state $q \in F$ such that

$$\boxed{(q_0, w) \vdash_M^* (q, \epsilon)}.$$

The language accepted by M, $L(M)$, is the set of all strings accepted by M.

The transition diagram is represented in two ways.

1. Transition diagram.
2. Transition table.

Example -

Let M be the DFA $(Q, \Sigma, \delta, q_0, F)$
where $Q = \{q_0, q_1\}$

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

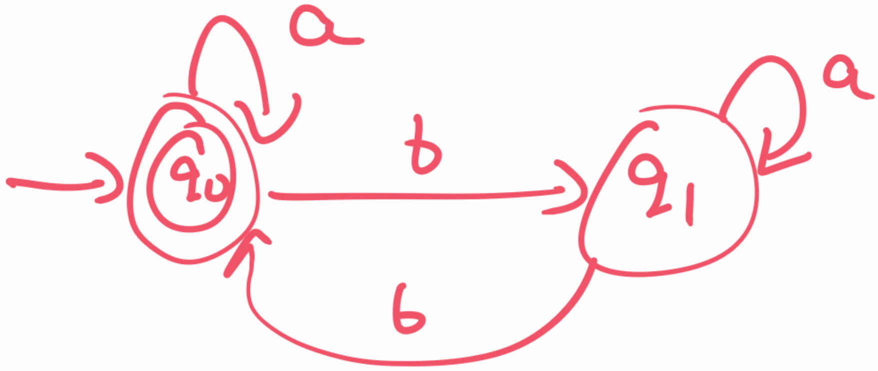
$q_0 = q_0$

$F = \{q_0\}$

δ: (Using transition table)

q	Σ	$\delta(q, \sigma)$
$\rightarrow q_0$	a	q_0
q_0	b	q_1
q_1	a	q_1
q_1	b	q_0

δ: (using transition diagram)



$L(M) = \{ \epsilon, a, aa, \ldots, abab, \ldots$

$\ldots, bb, \ldots \}$

$$(q_0, aabba) \underset{M}{\overset{1}{\vdash}} (q_0, abba)$$

$$\underset{M}{\overset{1}{\vdash}} (q_0, bba)$$

$$\underset{M}{\overset{1}{\vdash}} (q_0, ba)$$

$$\underset{M}{\overset{1}{\vdash}} (q_0, a)$$

$$\underset{M}{\overset{1}{\vdash}} (q_0, \epsilon)$$

Accepted

1) Construct DFA for the following language

$L = \{ w \in \Sigma^* \mid |w| \leq 3 \}$

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

$L = \{ \epsilon, a, b, aa, ab, b \ldots, aaa, \ldots \}$

$M_1 = (Q, \Sigma, \delta, q_0, F)$

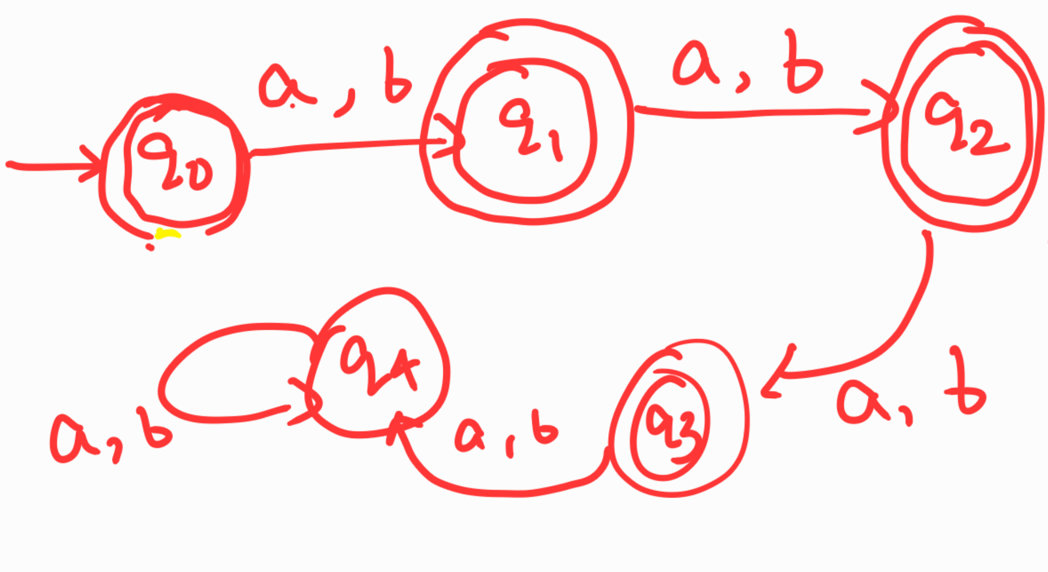
$\Sigma = \{a, b\}$ ✓

$q_0 = \underline{q_0}$ ✓

$Q = \{ q_0, q_1, q_2, q_3, q_4 \}$ ✓

$F = \{ q_0, q_1, q_2, q_3 \}$ ✓

δ:



✓

2) Design DFA for the following languages over the alphabet $\Sigma = \{0,1\}$

i) $L = \{ w : w$ starts with 1 & $|w| = 1 \pmod 3 \}$

ii) $L = \{ w : w$ starts with 01 & the length of the string is divisible by 4.

iii) $L = \{ w : w$ starts with 101 & $|w| = 2 \pmod 5)$

i) $L = \{ 1, 1x \mid x \in \Sigma^* \text{ \& } |x| = \cancel{0}\ 3, 6, 9, \ldots \}$

$L = \{ 1x \mid x \in \Sigma^* \text{ \& } |x| = 0, 3, 6, 9, \ldots \}$

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

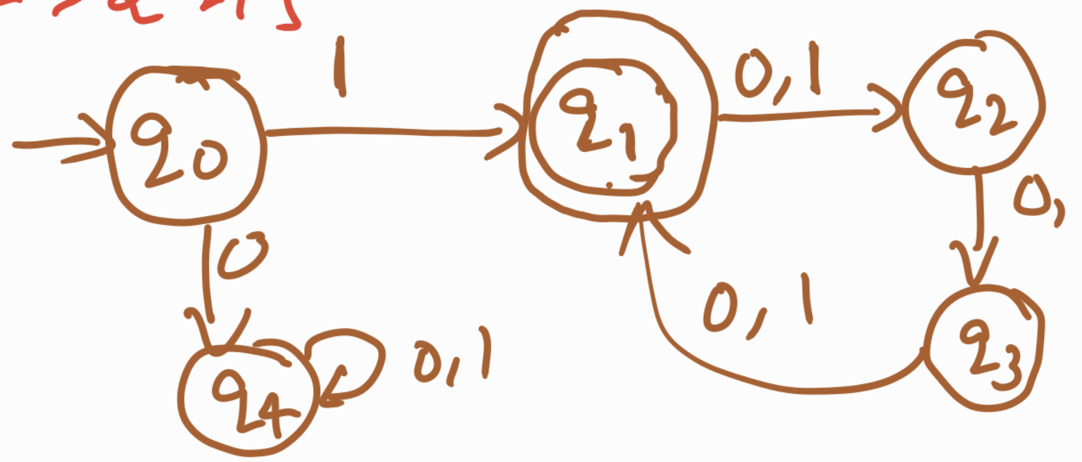
$\Sigma = \{0, 1\}$

$q_0 = q_0$

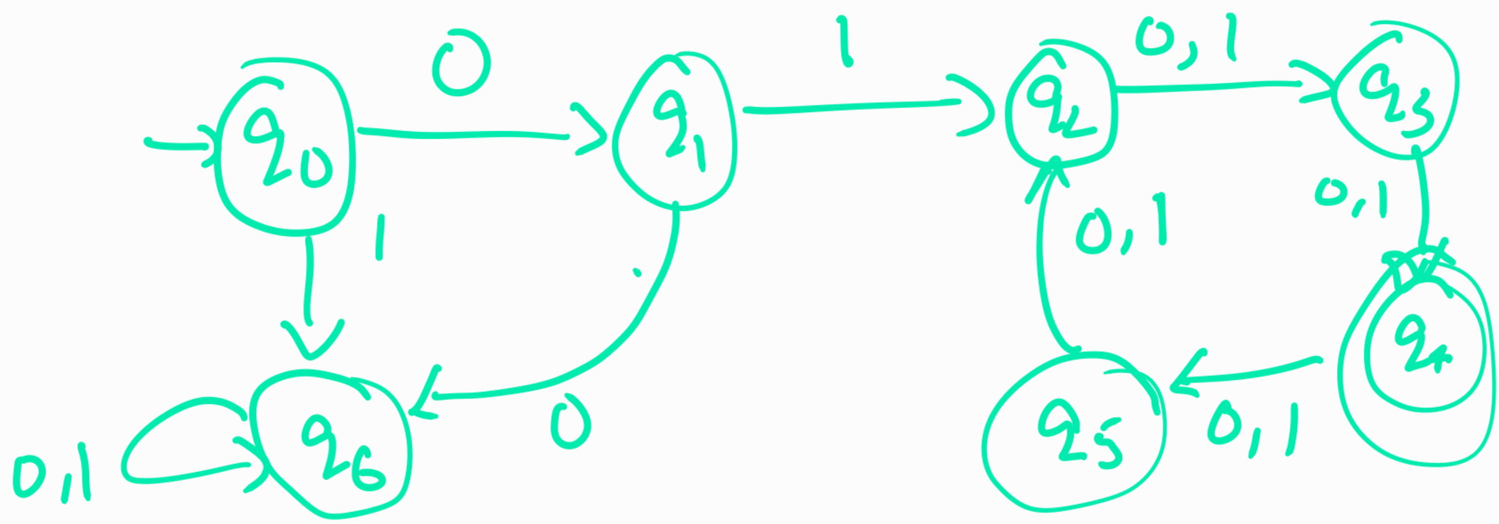
$Q = \{q_0, q_1, q_2, q_3, q_4\}$

$F = \{q_1\}$

δ :



ii) $w = \underline{01x}$ & $|w| \equiv 0 \pmod 4$



iv) $L = \{ \underline{w} \mid$ w starts with 1 & the number of 0's is divisible by 3 $\}$

$L = \{ \underline{1},\ 1\underline{x} \mid$ the no. of zeros in x is divisible by $3 \}$

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

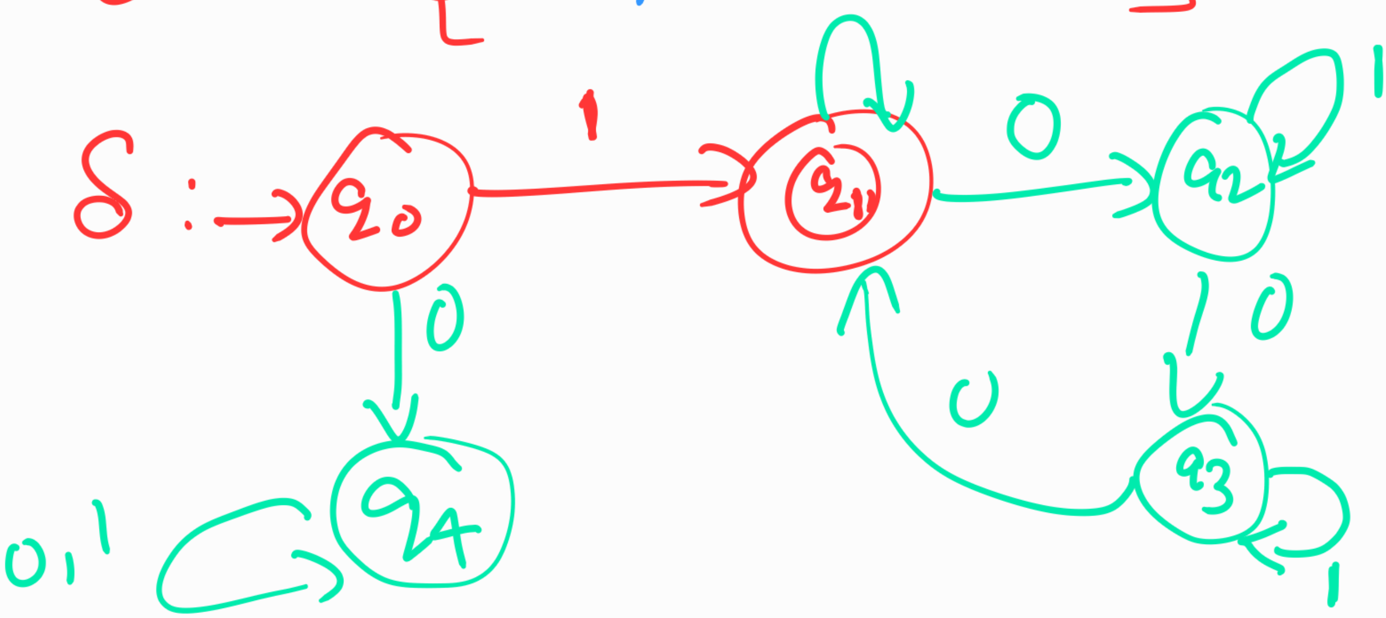
$\Sigma = \{0, 1\}$

$q_0 = q_0$

$F = \{q_1\}$

$Q = \{q_0, q_1, q_2, q_3, q_4\}$

δ :

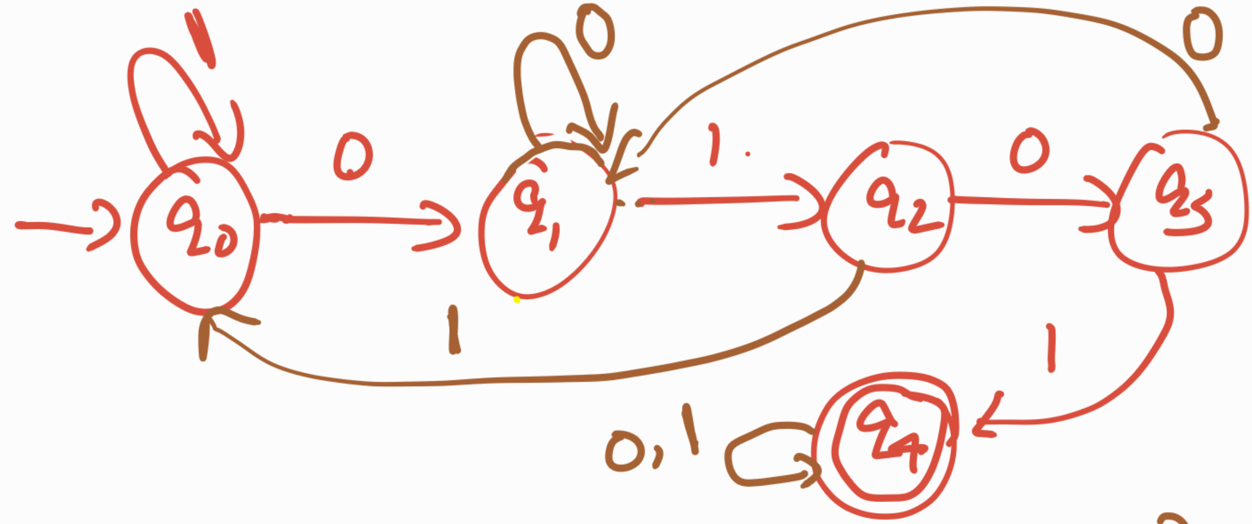


v) $L_4 = \{ w :$ w contains $\underline{0101}$ as a substring $\}$

$= X\,\underline{0101}\,Y$

$X, Y \in \underline{\Sigma}^*$

δ :



$Q = \{q_0, q_1, q_2, q_3, q_4\}$

$F = \{q_4\}$ $\quad q_0 = q_0$

vi) $L = \{ w \mid$ starts & ends with same symbol$\}$

$L = \{\epsilon, 0, 1, 0x0, 1x1 \mid x \in \Sigma^*\}$

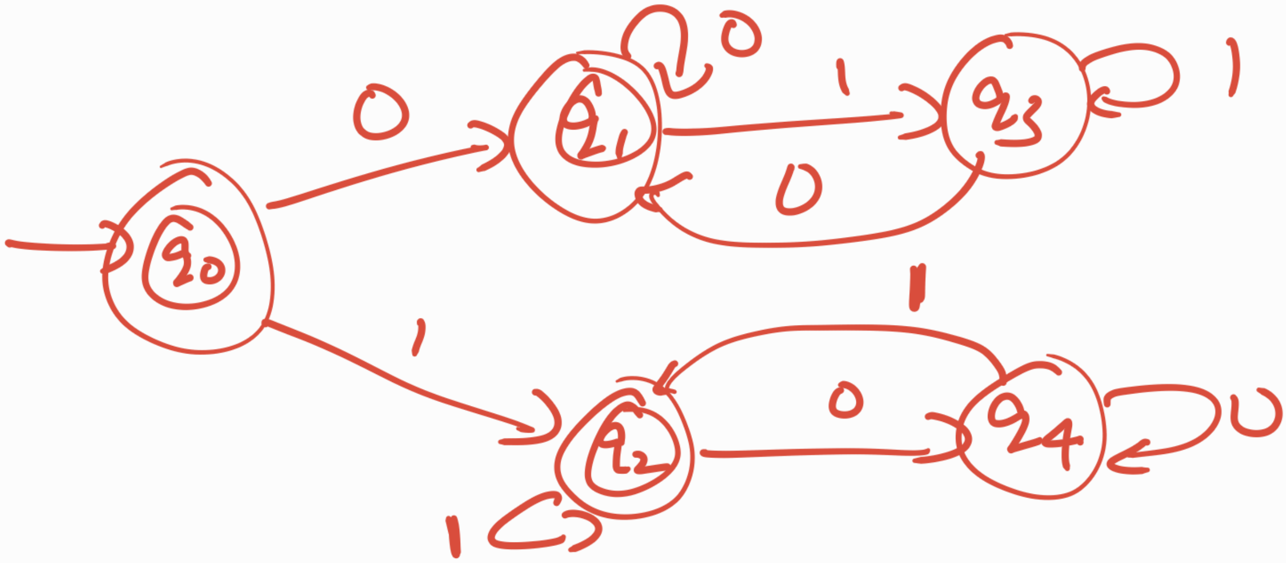
$M = (Q, \Sigma, \delta, q_0, F)$

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

$q_0 = q_0$

$Q = \{q_0, q_1, q_2, q_3, q_4\}$

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



H.W $\Sigma = \{0,1\}$

1) $L = \{ w :$ The fourth symbol from left end is always $\underline{1} \}$.

2) $L = \{ w :$ The third symbol from right end is always $\underline{0} \}$. $x0(0+1)(0+1)$

7-1-2021

1. Construct the DFA to accept all the strings over the alphabet $\Sigma = \{0, 1\}$ such that each string contain even number of 0's & odd number of 1's.

2. Construct the DFA that accepts all the strings of 0's & 1's, where its decimal equivalent is divisible by 3.

3. Design DFA's for the following Languages.

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

ii) $L = \{a^m b^n \mid m \geq 0, n \geq 1\}$
$\Sigma = \{a, b\}$

iii) $L = \{w : w$ does not contain 101 over the alphabet
$\Sigma = \{0, 1\}$

iv) $L = \{ w \in \{a,b\}^* :$ each a in w is immediately Preceded by b. $\}$

b, bb, ab,

v) $L = \{ w \in \{a,b\}^* :$ w has both ab and ba as substrings$\}$

aba
bab

✓ vi) $L = \{ w \in \Sigma^3 \mid |w|_a = 2 \cdot |w|_b \}$

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

vii) $L = \{ w \in \Sigma^3 \mid |w|_a < |w|_b \}$.

viii) $L = \{ w \in \Sigma^3 \mid w = w^R \}$

Non Deterministic Finite Automata (NFA)

The ability to change the states in a way that is only partially determined by the current state & input symbol - is called non-determinism.

That is we shall permit several possible next states for a given combination of current state & input symbol.

"Every NFA is equivalent to DFA".

NFA=DFA

An NFA is a quintuple $M=(Q, \Sigma, \delta, q_0, F)$, where

$Q \rightarrow$ It is a finite set of states.

$\Sigma \rightarrow$ An alphabet

$q_0 \rightarrow$ Initial state

$F \subseteq Q \rightarrow$ Set of final states

δ : Transition function :

$$\delta : Q \times \Sigma \longrightarrow 2^{Q}$$

The transition from a state can be multiple next states for each input symbol. Hence it is called non-deterministic.

A string is accepted by a NFA machine M, if atleast one of all possible transitions ends in a final state.

$$(q_0, \omega) \overset{*}{\underset{M}{\vdash}} (q, \epsilon)$$

$\hookrightarrow$ is a final state

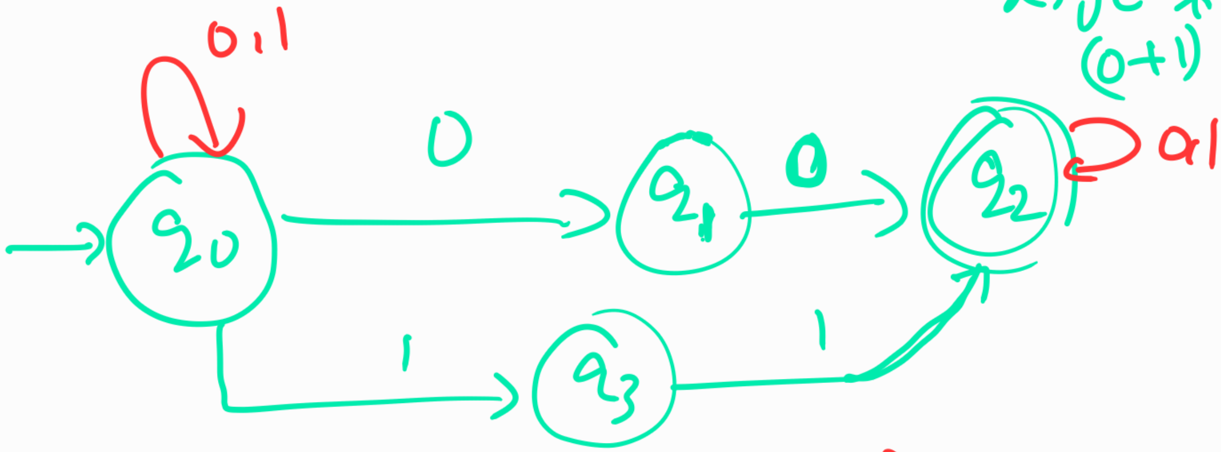
Ex - Design NFA to accept all ~~the~~ strings contain 00 or 11 over ~~the~~ alphabet $\Sigma = \{0, 1\}$.

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

$\Sigma = \{0, 1\}$ $\qquad q_0 = q_0$ $\qquad x00y$ or $x11y$

$x, y \in (0+1)^*$

δ:



$Q = \{q_0, q_1, q_2, q_3\}$

$F = \{q_2\}$

NFA to DFA

If there is an NFA $M = (Q, \Sigma, \delta, q_0, F)$ which recognizes L. Then there exist a DFA $M' = (Q', \Sigma, q_0, \delta', F')$ can be constructed for machine M as:

1. Initially $Q' = \phi$
2. Add q_0 to Q'.
3. For each state in Q' find the set of possible states for each input symbol using transition function of NFA. If this set of states not in Q' then add it to Q'.
4. Final state of DFA will be all states which contain F.

Ex: Convert the following NFA into DFA.

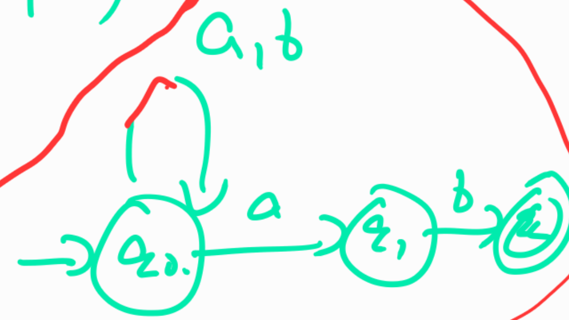
$M = (Q, \Sigma, \delta, q_0, F)$

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

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

$F = \{q_2\}$

δ:



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

$x\,ab$

$x\,(a+b)^*$

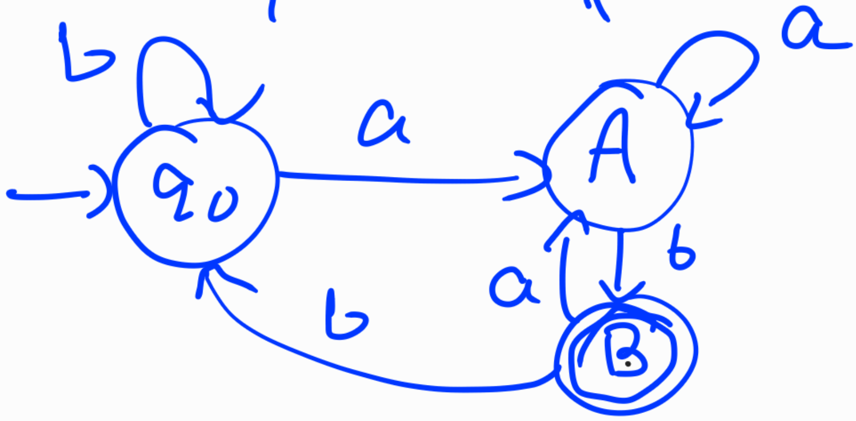
$\Sigma = \{a, b\}$ $\quad q_0 = q_0$

$Q' = \{q_0, A, B\}$

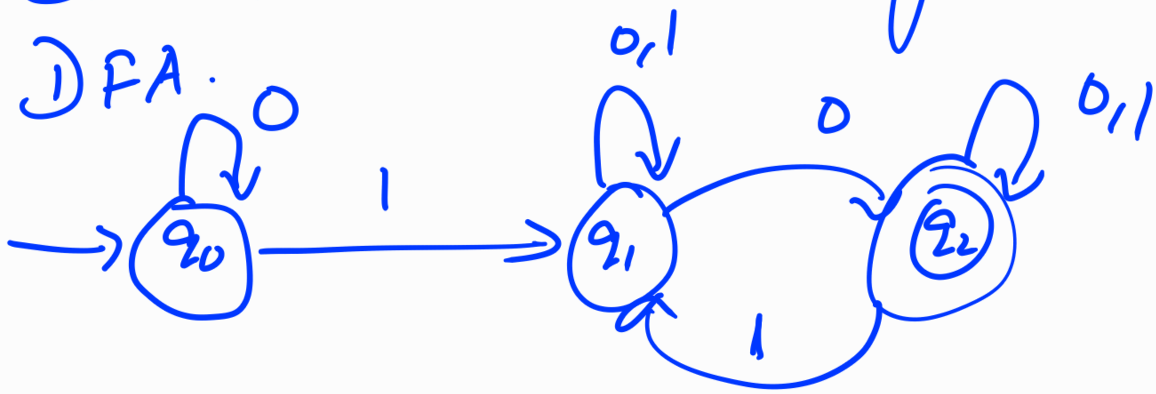
$F = \{B\}$

δ':

	a	b
$\rightarrow q_0$	$\underbrace{\{q_0, q_1\}}_{A}$	$\{q_0\}$
A	$\{q_0, q_1\}$	$\underbrace{\{q_0, q_2\}}_{B}$
B	$\{q_0, q_1\}$	q_0



2) Convert the following NFA into DFA.



FA with ϵ-moves

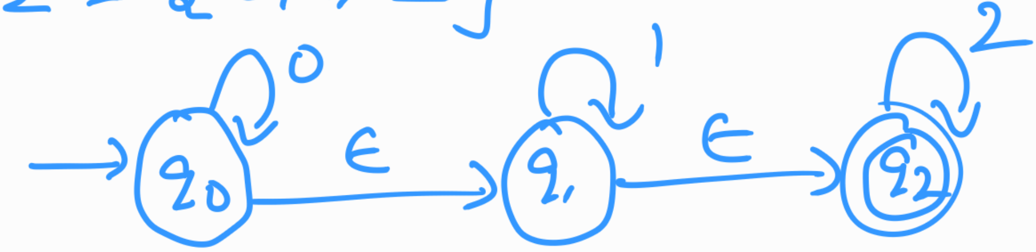
$M = (Q, \Sigma, \delta, q_0, F)$

$\delta : Q \times (\Sigma \cup \{\epsilon\}) \rightarrow 2^Q$

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

$\Sigma = \{0, 1, 2\}$

δ:



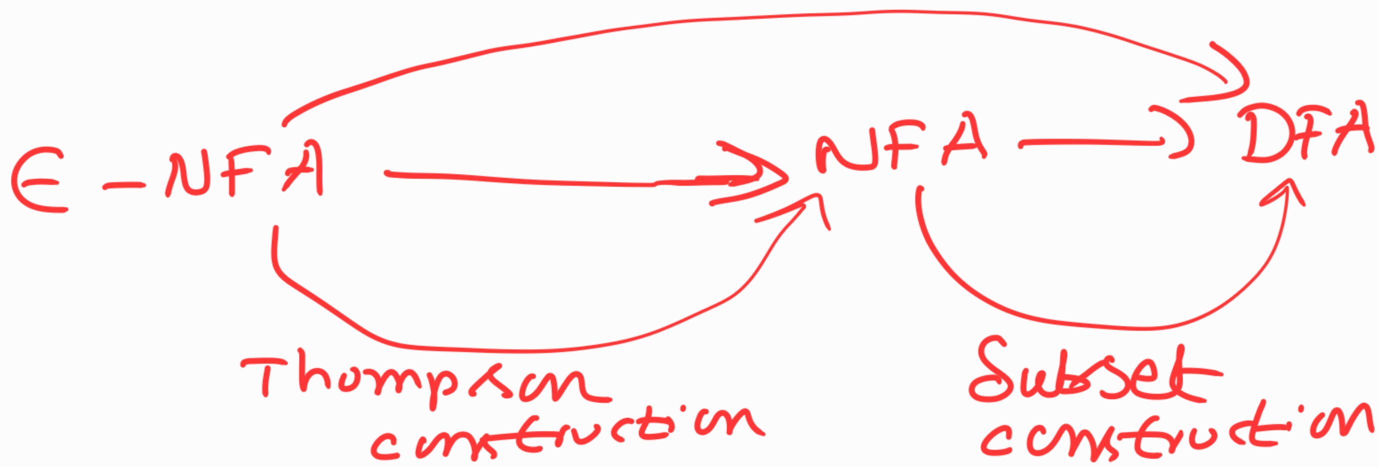
Construct the ϵ-NFA for the following languages.

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

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

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

* If L is accepted by an NFA with ϵ-transitions, then L is also accepted by an NFA without ϵ-transitions.

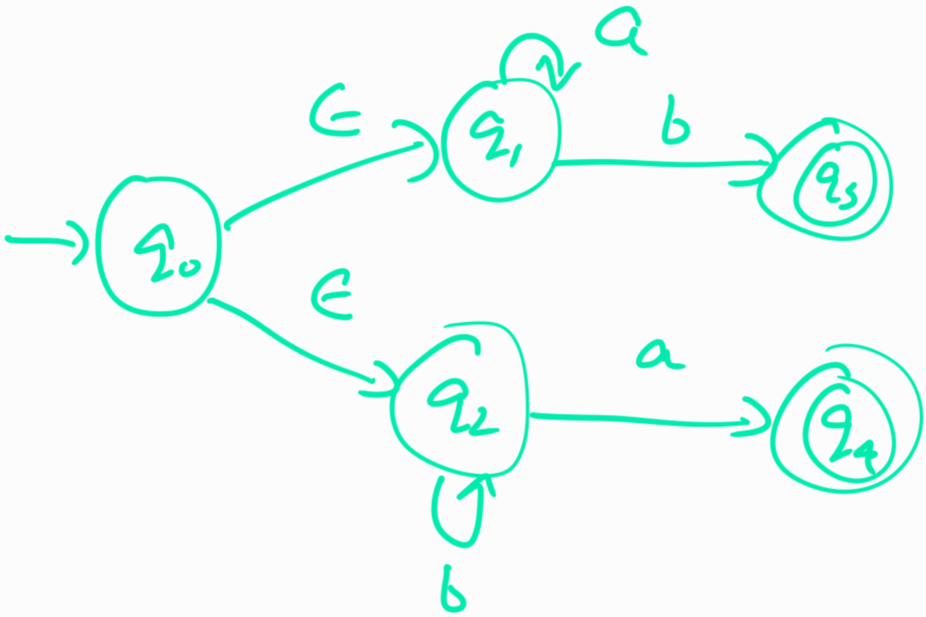


ϵ-closure(q) – If q is any state in ϵ-NFA then the set of all states which are at 0-distance from state q is known as ϵ-closure(q).

(or)

The "set of all states that can be reached from state q along ϵ-enabled transition path" is known as ϵ-closure(q).

Ex



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

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

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

$\epsilon\text{-closure}(q_3) = \{$

$\epsilon\text{-closure}(q_4) = \{$

Equivalence between ϵ-NFA & NFA (ϵ-NFA $\longrightarrow$ NFA)

Let $M = (Q, \Sigma, \delta, q_0, F)$ be ϵ-NFA

$M' = (Q', \Sigma, \delta', q_0', F')$ be an NFA

1) Initial state - No change in the initial state.

$$q_0' = q_0.$$

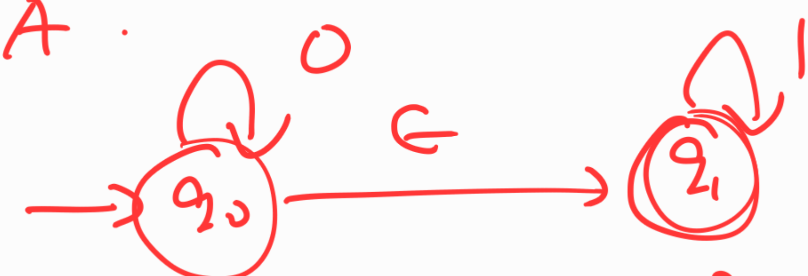
2) Construction of δ':

$$\delta'(q, x) = \epsilon\text{-closure}\{\delta(\epsilon\text{-closure}(q), x)\}$$

3) $Q' = Q$

4) "Every state whose ϵ-closure contain the final state of ϵ-NFA is a final state in NFA".

Ex -



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