Formal Language

A Language is a collection of sentences; a sentence is a sequence of words; and a word is a combination of syllables.

It is observed that, a formal Learning of Language has the following three steps:

1. Learning its alphabet
   - the symbols that are used in the language.
2. Its words - as various sequences of symbols of its alphabet.
3. formation of sentences:-
   sequence of various words that follow certain rules of the language.

Symbol : It is a user defined entity. An atomic unit, such as digit, character, lower-case letter, etc. Sometimes a word.

* formal language doesn't deal with the "meaning" of the symbols.

Alphabet: A finite set of Symbols, usually denoted by 'Σ' in automata.

Ex: $\Sigma = \{0, 1\}$ $\Sigma = \{0, 9, a, 4\}$

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

String: A finite Length sequence of symbols, presumably from some alphabet. A String is denoted by 'w' in automata.

Ex: $w = 0110$ $w = 0aa$ $w = aabcaa$

Length of a String: It is the no. of symbols present in a string w which is denoted by '$|w|$'.

Ex: If $w = aabcaa$, then $|w| = 6$.

If $w = 0$, it is called an empty string i.e., ϵ

(It is denoted by λ).

Concatenation: If w & x are two string, then wx is the concatenation of w & x.

Ex: if $w = ab$ & $x = cd$,

then concatenation of w & x is

$R = abcd$.

Language

Definition: A Language is a ~~set~~ subset of Σ^* for some alphabet Σ.
It can be finite or infinite.

Ex:

1. $L1 = \{$ Set of string of length 2 $\}$
$= \{ aa, bb, ba, bb \}$

2. $L2 = \{$ set of string ~~of leng~~ starts with 'a' $\}$
$= \{ a, aa, aaa, abb, abbb, ababb, \ldots \}$

* When ε is concatenated with any string it returns the string only.

$\omega_1 = abc$
$\omega_2 = \varepsilon$

$\omega_1 \omega_2 = abc.$

* $\omega_1 = \varepsilon, \; \omega_2 = \varepsilon$
$\omega_1 \omega_2 = \varepsilon$

Kleene Star / Kleene closure (Σ^*)

It is a unary operator on a set of symbols or strings, Σ, that gives the infinite set of all the possible strings of all possible lengths over Σ including ε.

Representation:

$\Sigma^* = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2 \cup \ldots$ where Σ_p is the set of all possible strings of length p.

Ex: If $\Sigma = \{a, b\}$,
$\Sigma^* = \{\varepsilon, a, b, aa, ab, ba, bb, \ldots\}$

Positive closure / Kleene Plus (Σ^+)

The infinite set of all possible strings of all possible lengths over Σ excluding ε.

Representation:

$\Sigma^+ = \Sigma_1 \cup \Sigma_2 \cup \Sigma_3 \cup \ldots$

$\boxed{\Sigma^+ = \Sigma^* - \{\varepsilon\}}$

Ex: If $\Sigma = \{a, b\}$
$\Sigma^+ = \{a, b, ab, ba, bb, \ldots\}$

Regular Expression

The patterns corresponding to a token are generally specified using a compact notation.

A language are created by combining its alphabets.

Operation on Language:

1. Union: Let L_1 & L_2 are two language then union of two language denoted as $L_1 \cup L_2$. or $L_1 + L_2$ or $L_1 | L_2$ defined by $L_1 \cup L_2 = \{x : x \in L_1 \text{ or } x \in L_2\}$

   Ex: $L_1 = \{a, ab, cd\}$
   $L_2 = \{a, bc\}$
   $L_1 \cup L_2 = \{a, ab, bc, cd\}$

2. Intersection: Let L_1 & L_2 are two Language then intersection of two Language denoted as $L_1 \cap L_2$ defined by $L_1 \cap L_2 = \{x : x \in L_1 \text{ \& } x \in L_2\}$

   Ex: $L_1 = \{a, ab, cd\}$, $L_2 = \{a, bc\}$
   $L_1 \cap L_2 = \{a\}$.

3. Complement: Let L be the language then complement of L denoted as 'L^c' or '$\bar{L}$' defined by
   $L^c = \{x : x \in \Sigma^* - L\}$

   Ex: Let $\Sigma = (0,1)$ & $L = \{0, 1, 00, 11\}$
   then $L^c = \{\varepsilon, 0, 000, 001, 011..\}$

4. Reverse: Let L be the Language then reverse of L denoted as L^R defined reverse of L.

Scanned with CamScanner

Ex: L = abc
L^R = cba.

5. Concatenation: Let L_1 & L_2 are two Language then concatenation of two Language denoted by L_1L_2 or $L_1 . L_2$ defined by $L_1L_2 = \{x : x \in L_1 \times L_2\}$.

* Every string of L_1 concatenate with every string of L_2.

Ex: $L_1 = \{\varepsilon, a\}$, $L_2 = \{a, ab\}$
$L_1L_2 = \{a, ab, aa, aab\}$

$L_1 = \{\varepsilon, a\}$, $L_2 = \{\varepsilon, b\}$
$L_1L_2 = \{\varepsilon, b, a, ab\}$.

Dt - 20/01/21

Automata: It is defined as a system where information is transmitted & used for performing some task without direct participating of human being.

- The output depends only on input is called Automata without memory.

Scanned with CamScanner

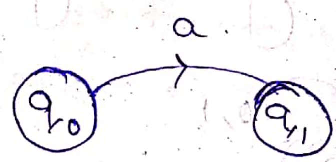
- The output depends on input & state is called Automata with Memory.

- The output depends ~~on~~ only on state is called Moore Machine.

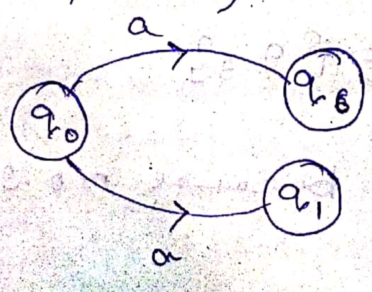
Finite State Machine / finite Automata (FSM/FA)

- It is a machine or mathematical model of a machine, which can only reach finite no. of state & transition.
  - It is called finite as it uses finite state & alphabet.

It is of two types:
- Deterministic finite Automata (DFA).



- Non-Determenistic finite Automata (NDFA/NFA)



DFA

It is a model of machine that has finite set of state & transition function where the ~~format~~ machine in a current state takes i/p symbol & uniquely determine the next state.

formally,

It is defined as

$$\boxed{M = (Q, \Sigma, \delta, q_0, f)}$$

where Q : finite set of states

Σ : finite Alphabet

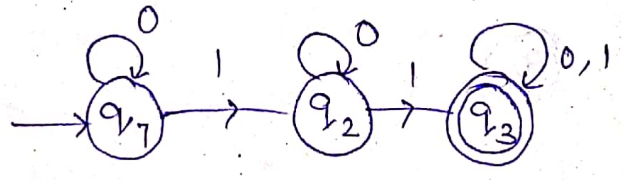
δ : Transition function from $Q \times \Sigma \rightarrow Q$.

q_0 : Initial / Start state

f : Set of final / accepting state.

$\delta(q, d) = q_1$

You are q looking to d going to q_1.



$Q = \{q_7, q_2, q_3\}$

$\Sigma = (0, 1)$

$q_0 = \{q_7\}$ $\quad f = \{q_3\}$

* q_7 is start because no state to state.

$\delta(q_7, 0) = q_7$

$\delta(q_7, 1) = q_2$

$\delta(q_2, 0) = q_2$

$\delta(q_2, 1) = q_3$

$\delta(q_3, 0) = q_3$

$\delta(q_3, 1) = q_3$

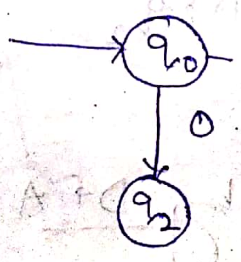
- State is represented by circle.

state.

- final state is represented by double circle.

- To a state from a not state represent initial state.

arc ---→ from a state to other state or itself represent transition.

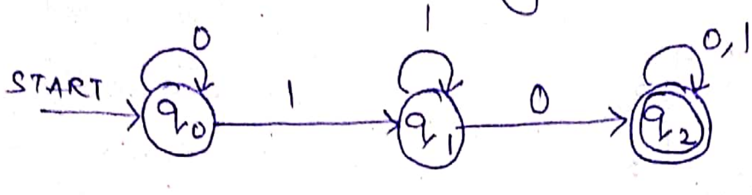


Transitions show the current state, input, & next state

form: $\delta(q_0, 0) = q_2$, $Q \times \Sigma \rightarrow Q$

Transition can be represented in three ways:

1. Transition Diagram:



2. Transition Table:

Present State	Next State for Input 0	Next State for Input 1
$\rightarrow q_0$	q_0	q_1
q_1	q_2	q_1
$* q_2$	q_2	q_2

3. Transition function:

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

Acceptance of string by DFA

A string w accepted by DFA 'M' if $\delta(q_0, w) = p$ for some p in final state F. i.e., we can reach to the final state after processing to all symbols of the string.

Properties:

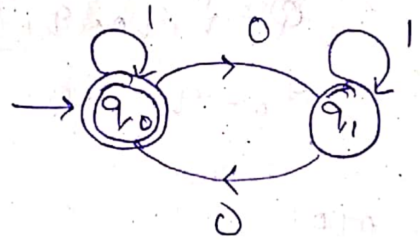
1. $\delta(q, \varepsilon) = q$
2. $\delta(q, aw) = \delta(\delta(q, a), w)$.

Numericals

DFA Design over $\Sigma = (0, 1)$.

1. Even number of 0.

$L = \{00, 0000, 100, 001, 11, 0101, \ldots\}$



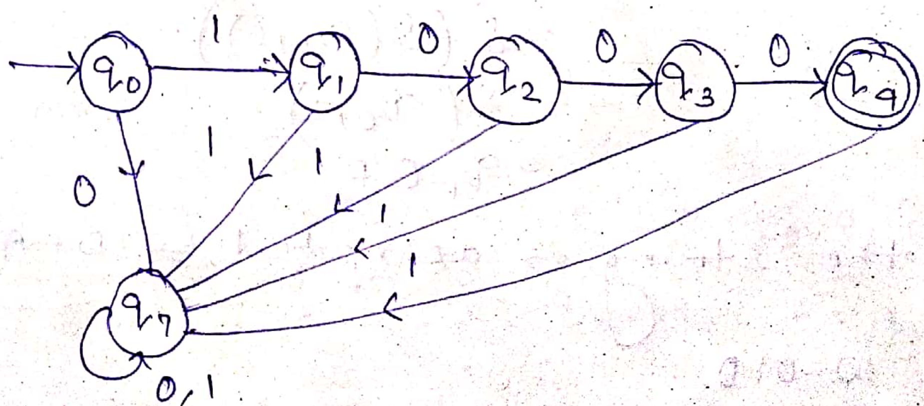
$M = (Q, \Sigma, \delta, q_0, F)$

$Q = \{q_0, q_1\}$

$\Sigma = (0, 1)$

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

2. Accept the string 1000 only.

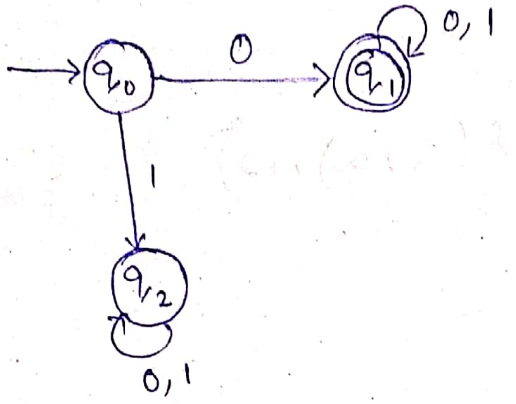


$Q = \{q_0, q_1, q_2, q_3, q_4\}$

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

3. Started with 0.

$L = \{00, 01, 0110, 0111, \ldots\}$



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

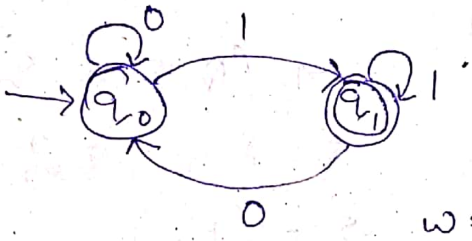
$Q = \{q_0, q_1, q_2\}$

$\Sigma = (0,1)$

$q_0 = q_0$, $F = q_1$

4. Ended with 1.

$L = \{01, 11, 011, \ldots\}$



$Q = \{q_0, q_1\}$

$q_0 = q_0$, $F = q_1$

$w = 0101$

$$\begin{aligned}
\delta(q_0, 0101) &= \delta(\delta(q_0, 0), 101) \\
&= \delta(q_0, 101) \\
&= \delta(\delta(q_0, 1), 01) \\
&= \delta(q_1, 01) \\
&= \delta(\delta(q_1, 0), 1) \\
&= \delta(q_0, 1) \\
&= q_1 \in F
\end{aligned}$$

So the string is accepted by DFA.

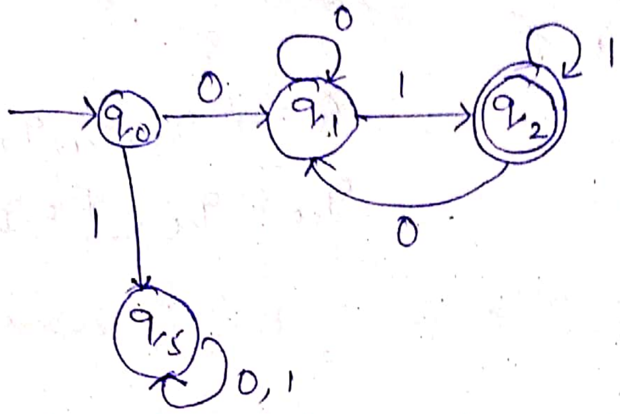
$w = 010$

$$\begin{aligned}
\delta(q_0, 010) &= \delta(\delta(q_0, 0), 10) \\
&= \delta(q_0, 10) \\
&= \delta(\delta(q_0, 1), 0) \\
&= \delta(q_1, 0) = q_0 \notin F
\end{aligned}$$

So language is not accepted by DFA.

Dt - 23/01/21.

5. Started with 0 & ended with 1.



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

$q_0 = q_0, \; f = q_2$.

$w = 0101$.

$\delta(\delta q_0, 0101) = \delta(\delta(q_0, 0), 101)$

$= \delta(q_1, 101)$

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

$= \delta(q_2, 01)$

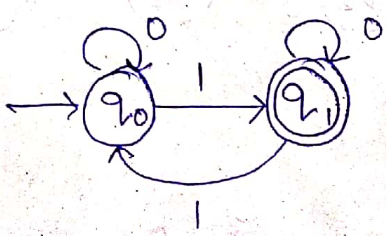
$= \delta(\delta(q_2, 0), 1)$

$= \delta(q_1, 1)$

$= q_2 \in f$.

Hence, the string is accepted by DFA.

6. Odd number of 1.

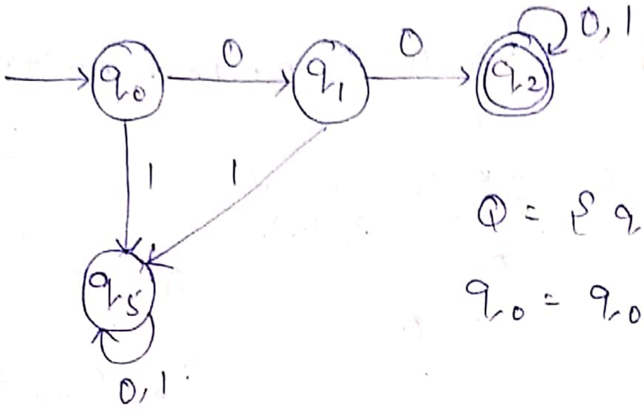


$Q = \{q_0, q_1\}$

$q_0 = q_0, \; f = q_1$

7. Started with 00.

$L = \{00, 000, 001, 0010, 0011, \ldots\}$

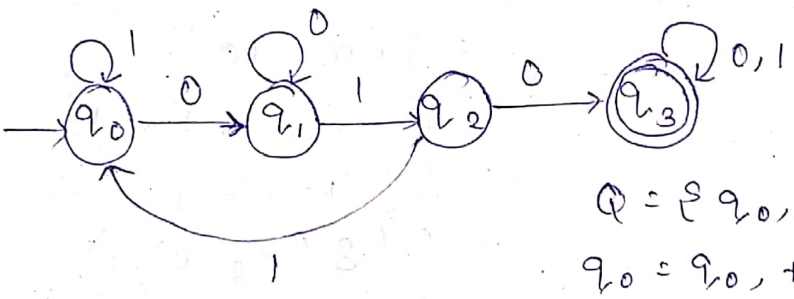


$Q = \{q_0, q_1, q_2, q_5\}$

$q_0 = q_0, \quad f = q_2.$

8. Accept the string having substring 010.

$L = \{010, 0101, 11010, 01011, \ldots\}.$

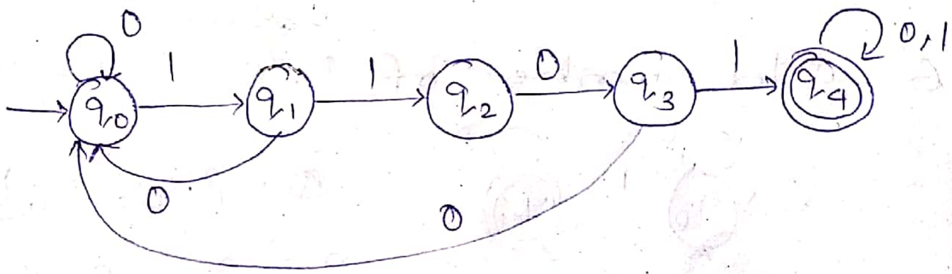


$Q = \{q_0, q_1, q_2, q_3\}$

$q_0 = q_0, \quad f = q_3.$

9. Accept the string having substring 1101.

$L = \{1101, 01101, 11010, \ldots\}$

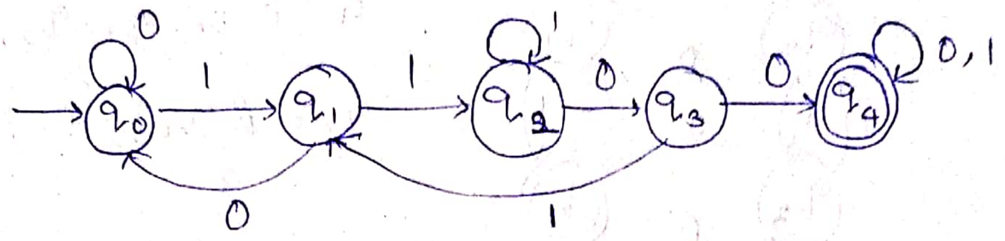


$Q = \{q_0, q_1, q_2, q_3, q_4\}$

$q_0 = q_0, \quad f = q_4$

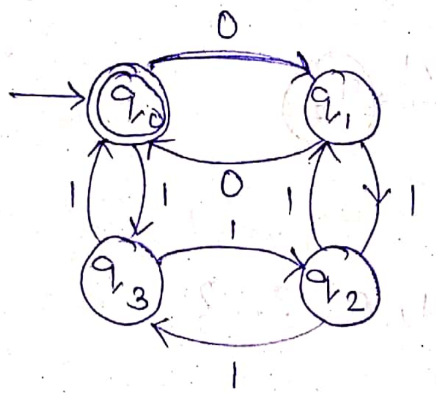
10. Accept the string having substring 1100.

$L = \{1100, 01100, 11100, \ldots\}$



Dt- 27/07/21

11. Even number of 0 & Even number of 1.

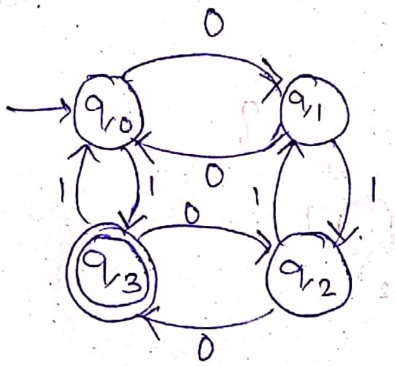


$M = \{Q, \Sigma, \delta, q_0, F\}$

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

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

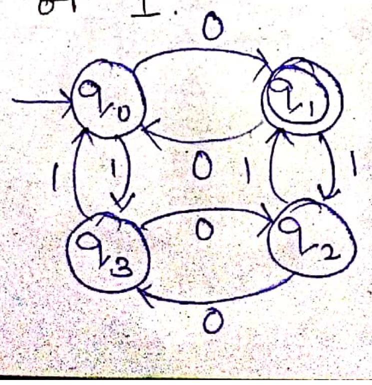
12. Even number of 0 & odd number of 1.



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

$q_0 = \{q_0\}, f = \{q_3\}$

13. Odd number of 0 & Even number of 1.

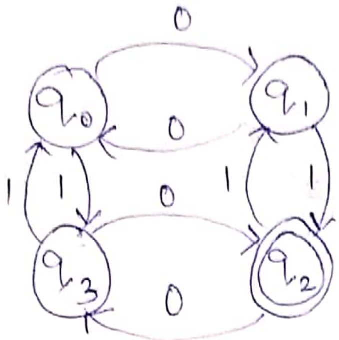


$Q = \{q_0, q_1, q_2, q_3\}$

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

Scanned with CamScanner

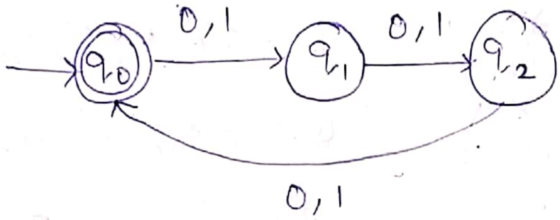
14. Odd number of 1 & odd number of 0.



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

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

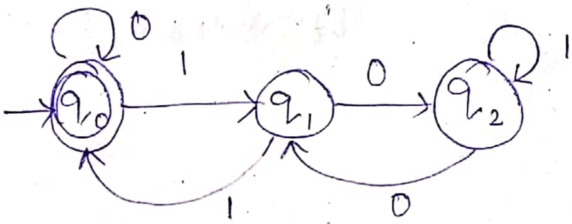
15. String length of modulo 3=0 (Length is multiple of 3)



$L = \{011, 010, 000111, \varepsilon, \ldots\}$.

16. Decimal representation of string modulo 3 = 0.

$L = \{011, 110, 000, 1001, \ldots\}$



$\delta(q_0, 0) = (2 * 0 + 0) \% 3$
$= 0 \% 3$
$= 0$

$\delta(q_0, 1) = (2 * 0 + 1) \% 3$
$= 1 \% 3$
$= 1$

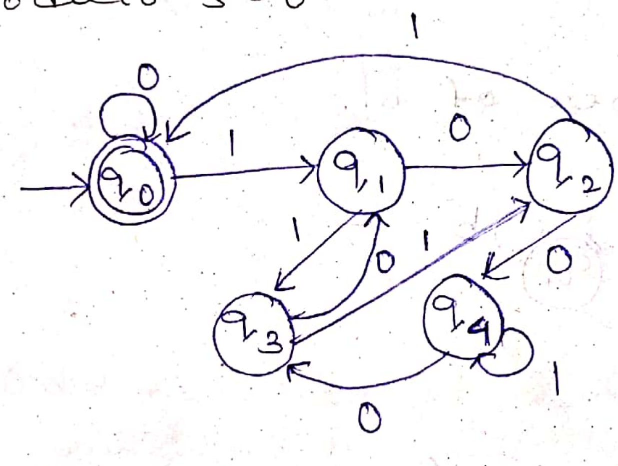
$\delta(q_1, 0) = (2*1+0)\%3$
$= 2\%3$
$= 2.$

$\delta(q_1, 1) = (2*1+1)\%3$
$= 3\%3$
$= 0.$

$\delta(q_2, 0) = (2*2+0)\%3$
$= 4\%3$
$= 1$

$\delta(q_2, 1) = (2*2+1)\%3$
$= 5\%3$
$= 2.$

17. Decimal representation of string modulo $5 = 0$.



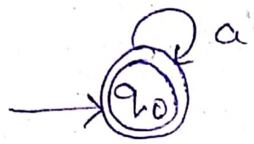
$\delta(q_0, 0) = (2*0+0)\%5 = 0\%5 = 0$
$\delta(q_0, 1) = (2*0+1)\%5 = 1\%5 = 1$
$\delta(q_1, 0) = (2*1+0)\%5 = 2\%5 = 2$
$\delta(q_1, 1) = (2*1+1)\%5 = 3\%5 = 3$
$\delta(q_2, 0) = (2*2+0)\%5 = 4\%5 = 4$
$\delta(q_2, 1) = (2*2+1)\%5 = 5\%5 = 0$
$\delta(q_3, 0) = (2*3+0)\%5 = 6\%5 = 1$
$\delta(q_3, 1) = (2*3+1)\%5 = 7\%5 = 2$
$\delta(q_4, 0) = (2*4+0)\%5 = 8\%5 = 3$
$\delta(q_4, 1) = (2*4+1)\%5 = 9\%5 = 4$

Dt - 28/01/21.

DFA Design over $\Sigma = (a, b)$

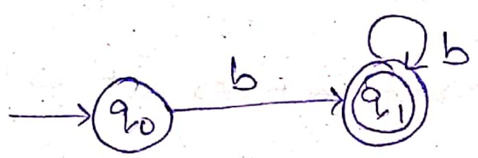
1. Accept the string a^*

$a^* = \{a, aa, aaa, aaaa, \ldots, \varepsilon\}$

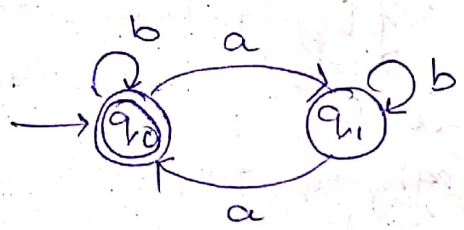


2. Accept the string b^+.

$b^+ = \{b, bb, bbb, \ldots\}$

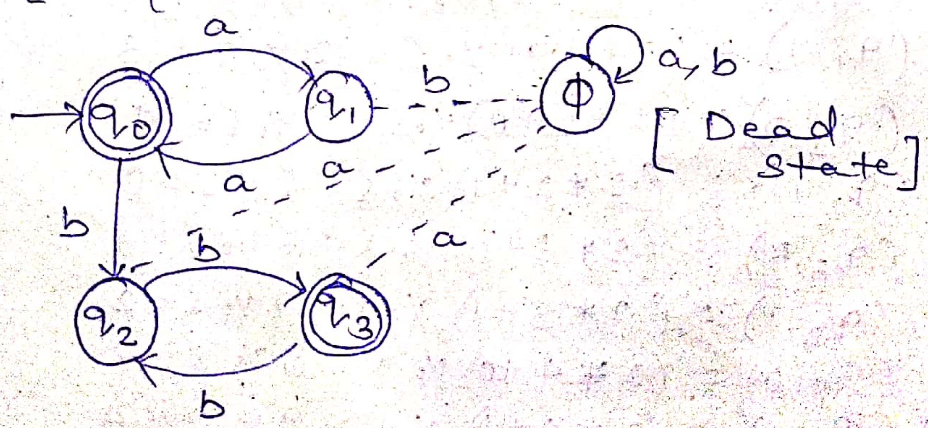


3. Even number of a.



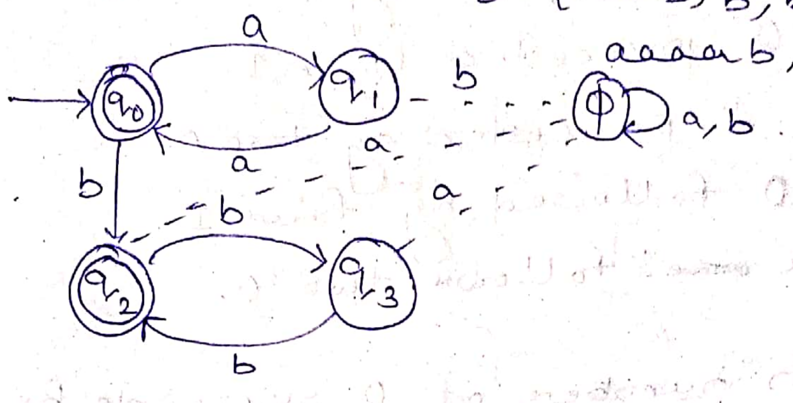
4. Even number of a followed by even number of b.

$L = \{aabb, aabbbb, aa, bb, \ldots \varepsilon\}$



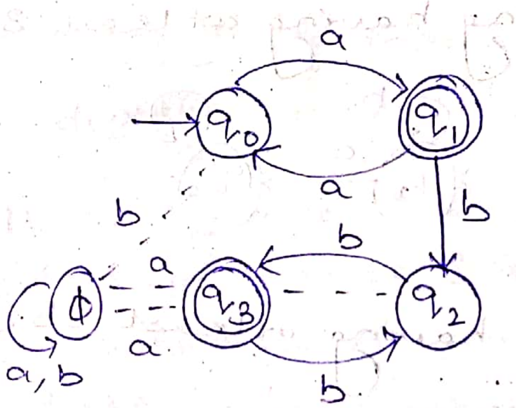
5. Even number of a followed by odd number of b.

L = {aab, b, bbb, aaaab, ... }



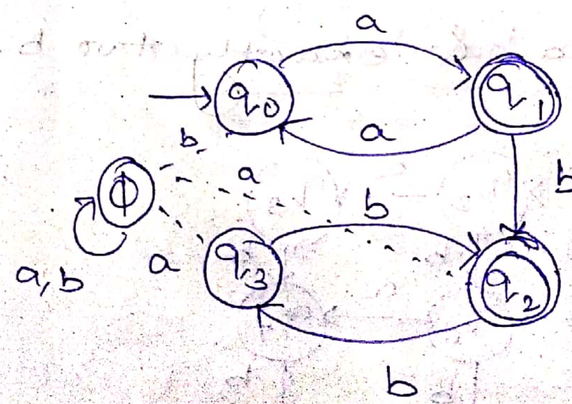
6. Odd number of a followed by even number of b.

L = {a, abb, aaabb, ... }



7. Odd number of a followed by odd number of b.

L = {ab, aaabbb, ... }

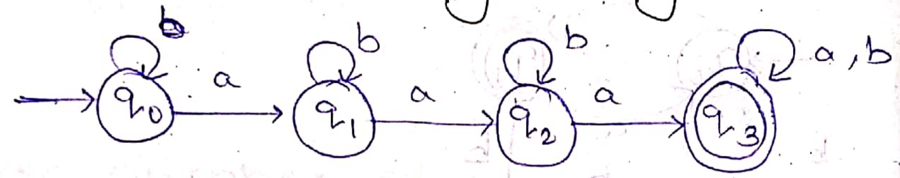


1. Foure 1 succeeded two 0.
2. Two 0 succeeded by foure 1.
3. Two 0 preceded foure 1.
4. foure 1 preceded by two 0.
5. Two 0 followed by foure 1.
6. foure ~~one~~ 1 follows two 0.

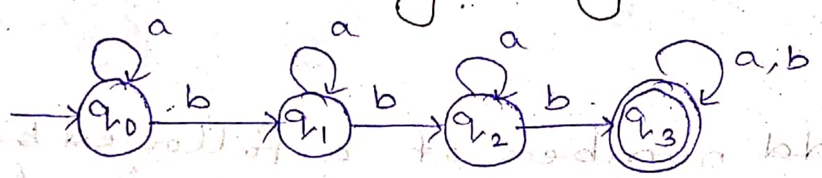
Q. Even number of 0 succeeds by even no. of 1.

Ans- foure 1 preceded by two 0.

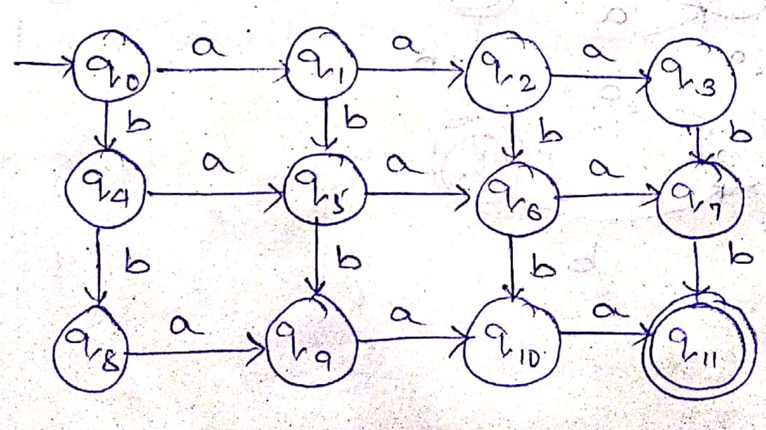
8. Accept the string having atleast 3a.



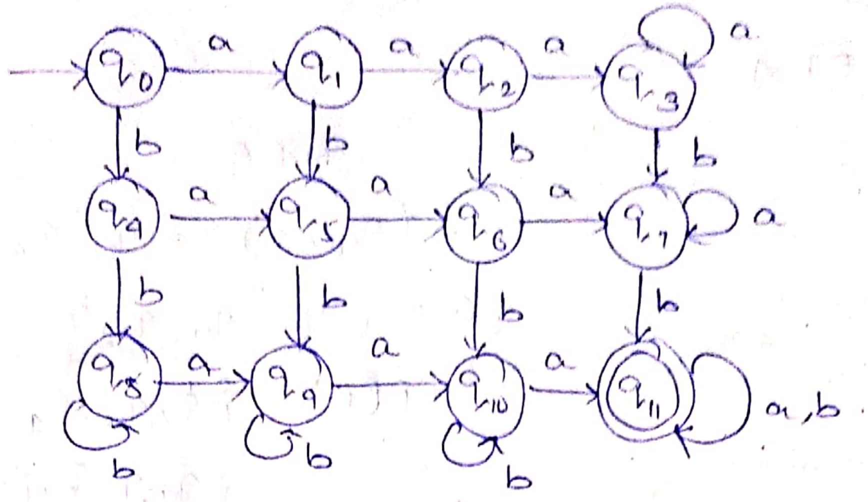
9. Accept the string having atleast 3b.



11. Exactly three a & exactly two b.



10. Atleast 3a & 2b.



Dt- 30/01/21

Non Deterministic finite Automata:
(NFA/NDFA)

It is a model of machine that has finite set of state & transition function, where the machine in a current state takes i/p symbol & not uniquely determine the next state.

formally,

It is defined as

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

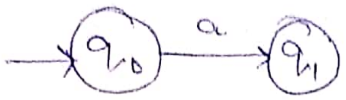
where, Q : finite set of states

Σ : finite Alphabet

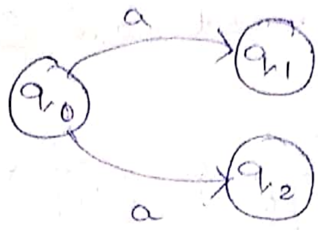
δ : Transition function from $Q \times \Sigma \to 2^Q$

q_0 : Initial/~~Start~~ state

F : Set of final/accepting state.



DFA



NFA

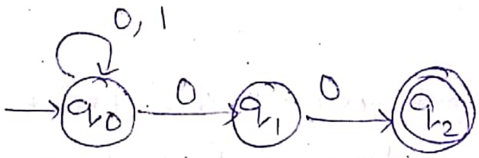
$\delta(q_0, a) = \{q_1, q_2\}$

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

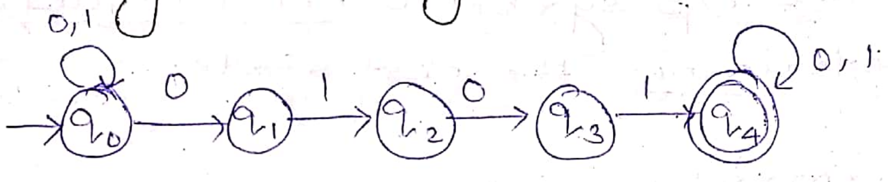
$2^Q = P(Q) = \{\phi, q_0, q_1, q_2, \{q_0, q_1\}, \{q_0, q_2\}, \{q_1, q_2\}, \{q_0, q_1, q_2\}\}$

Q. NFA Design over $\Sigma = (0,1)$

1. Ended with 00.

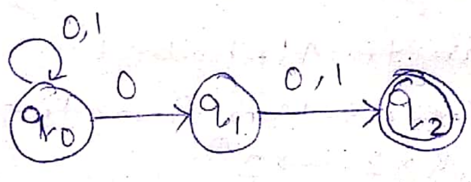


2. Having substring 0101.



Q. Design one NFA 2nd last symbol is 0.

L = {00, 100, 101, 1100, 1101, . . .}

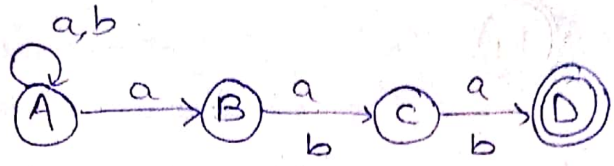


$\delta(q_0, 101)$
$= \delta(\delta(q_0, 1), 01)$
$= \delta(q_0, 01)$
$= \delta(\delta(q_0, 0), 1)$
$= \delta(\{q_0, q_1\}, 1)$
$= \delta(q_0, 1) \cup \delta(q_1, 1)$

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

Hence, Language is accepted by NFA.

3. Third symbol from right is a over $\Sigma = (a, b)$.



Acceptance of string by NFA

A string w accepted by NFA 'M' if $\delta(q_0, w) = p$ for some p in final state f.

i.e., we can reach to the final state after processing to all symbol of the string.

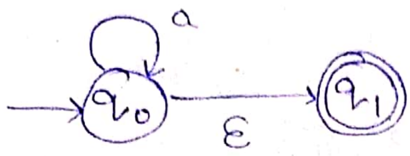
$* \quad \delta(\{q_0, q_1\}, w) = \delta(q_0, 0) \cup \delta(q_1, 0)$

Q. Check the string abb is accepted by NFA or not.

$$
\begin{aligned}
\delta(A, abb) &= \delta(\delta(A, a), bb) \\
&= \delta(\cancel{\delta}\{A, B\}, bb) \\
&= \delta(\delta(A, b) \cup \delta(B, b), b) \\
&= \delta(\cancel{\delta}\{A, C\}, b\} \\
&= \cancel{\delta}(\delta(A, b) \cup \delta(C, b)\} \\
&= \cancel{\delta}\{A, D\} \in F
\end{aligned}
$$

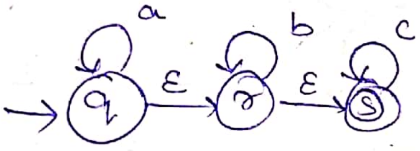
Hence, the language is accepted by NFA.

ε - closure

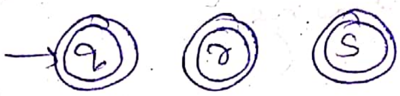


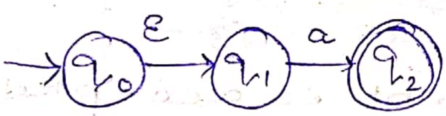
The ε - closure of the state q, denoted ε-closure(q), is the set that contains q, together with all the states that can be reached starting at q by following on ε - transitions.

with ε



without ε





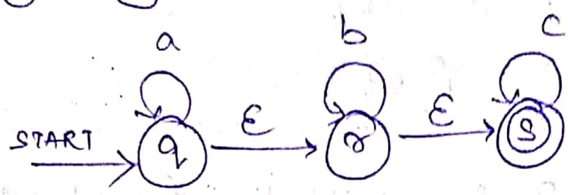
* If in with ε you are ~~reaching at~~ containing the final state by ε - ~~transition~~ closure then you are a final state ~~at~~ in without ε condition.

$$\delta'(q, a) = \delta((q, r, s), a)$$
$$= q$$

$$\delta'(q, b) = \delta((q, r, s), b)$$
$$= r$$

Ex: $\Sigma = (a, b)$

Accept the language having any number of a followed by any number of b followed by any number of c.



ϵ-closure $(q) = \{q, r, s\}$

ϵ-closure $(r) = \{r, s\}$

ϵ-closure $(s) = \{s\}$

Construct NFA without ϵ from NFA with ϵ

Let NFA with ϵ is $M = (Q, \Sigma, \delta, q_0, f)$

We have to construct NFA without ϵ,

Let it be $M' = (Q, \Sigma, \delta', q_0, f')$.

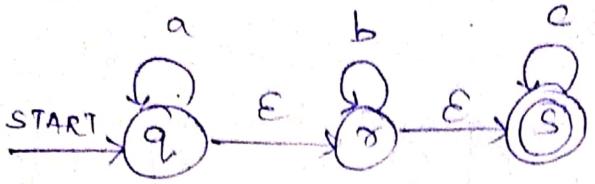
STEP 1: find ϵ-closure of all state.

If ϵ-closure of any state contains final state of NFA with ϵ then that will be the final state of NFA without ϵ.

STEP 2: find transition δ' by the rules

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

Ex:



$\epsilon\text{-closure}(q) = \{q, r, s\}$

$\epsilon\text{-closure}(r) = \{r, s\}$

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

Therefore, q, r, s will be final state in NFA without ε.

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

$Q = \{q, r, s\}$

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

$q_0 = q \quad , \quad F' = \{q, r, s\}$

$\delta' =$

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

$= \epsilon\text{-closure}(\delta(\{q, r, s\}), a)$

$= \cancel{\epsilon\text{-closure}(\delta(q, a))}$

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

$= \{q, r, s\}$.

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

$= \epsilon\text{-closure}(\delta(\{q, r, s\}, b))$

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

$= \{r, s\}$

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

$= \epsilon\text{-closure}(\delta(\{q, r, s\}, c))$

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

$= \{s\}$

$$\delta'(r, a) = \epsilon\text{-closure}(\delta(\epsilon\text{-closure}(r), a))$$
$$= \epsilon\text{-closure}(\delta(\{r, s\}, a))$$
$$= \epsilon\text{-closure}\{\}$$
$$= \{\}$$

$$\delta'(r, b) = \epsilon\text{-closure}(\delta(\epsilon\text{-closure}(r), b))$$
$$= \epsilon\text{-closure}(\delta(\{r, s\}, b))$$
$$= \epsilon\text{-closure}(r)$$
$$= \{r, s\}$$

$$\delta'(r, c) = \epsilon\text{-closure}(\delta(\epsilon\text{-closure}(r), c))$$
$$= \epsilon\text{-closure}(\delta(\{r, s\}), c))$$
$$= \epsilon\text{-closure}(s)$$
$$= \{s\}$$

$$\delta'(s, a) = \epsilon\text{-closure}(\delta(\epsilon\text{-closure}(s), a))$$
$$= \epsilon\text{-closure}(\delta\{s\}, a\}$$
$$= \epsilon\text{-closure}\{\}$$
$$= \{\}$$

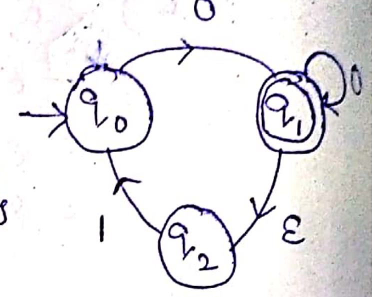
$$\delta'(s, b) = \epsilon\text{-closure}(\delta(\epsilon\text{-closure}(s), b))$$
$$= \epsilon\text{-closure}(\delta\{s\}, b)$$
$$= \epsilon\text{-closure}\{\}$$
$$= \{\}$$

$$\delta'(s, c) = \epsilon\text{-closure}(\delta(\epsilon\text{-closure}(s), c))$$
$$= \epsilon\text{-closure}(\delta(s, c))$$
$$= \epsilon\text{-closure}(s)$$
$$= \{s\}$$

Dt - 02/02/21

Ex 2 :

State/Input	0	1	ε
$\rightarrow q_0$	q_1		
$q_1(F)$	q_1		q_2
q_2		q_0	



ε - closure $(q_0) = \{q_0\}$

ε - closure $(q_1) = \{q_1, q_2\}$

ε - closure $(q_2) = \{q_2\}$

$$\delta'(q_0, 0) = \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_0), 0))$$
$$= \varepsilon\text{-closure}(\delta(q_0, 0))$$
$$= \varepsilon\text{-closure}(q_1)$$
$$= \{q_1, q_2\}$$

$$\delta'(q_0, 1) = \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_0), 1))$$
$$= \varepsilon\text{-closure}(\delta(q_0, 1))$$
$$= \varepsilon\text{-closure}\{\}$$
$$= \{\}$$

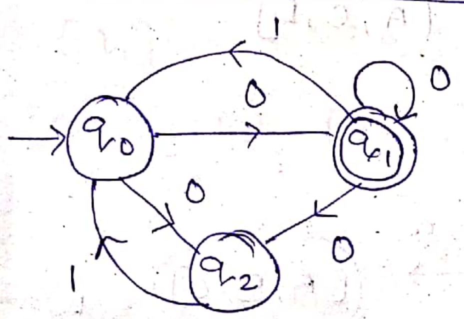
$$\delta'(q_1, 0) = \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_1), 0))$$
$$= \varepsilon\text{-closure}(\delta\{q_1, q_2\}, 0)$$
$$= \varepsilon\text{-closure}(\delta(q_1, 0) \cup \delta(q_2, 0))$$
$$= \varepsilon\text{-closure}(q_1)$$
$$= \{q_1, q_2\}$$

$$\delta'(q_1,1) = \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_1),1))$$
$$= \varepsilon\text{-closure}(\delta\{q_1,q_2\},1)$$
$$= \varepsilon\text{-closure}(\delta(q_1,1) \cup \delta(q_2,1))$$
$$= \varepsilon\text{-closure}(q_0)$$
$$= \{q_0\}$$

$$\delta'(q_2,0) = \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_2),0))$$
$$= \varepsilon\text{-closure}(\delta(q_2,0))$$
$$= \varepsilon\text{-closure}\{\}$$
$$= \{\}$$

$$\delta'(q_2,1) = \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_2),1))$$
$$= \varepsilon\text{-closure}(\delta(q_2,1))$$
$$= \varepsilon\text{-closure}(q_0)$$
$$= \{q_0\}$$

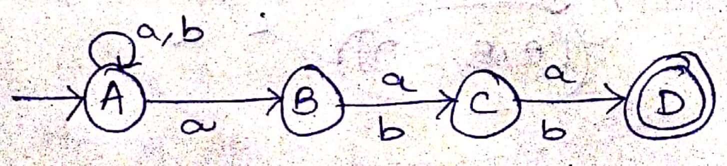
NFA <u>without ε</u>



Starting with 0
Ending with 0 in between 1 should not be consecutive.

NFA & DFA are equivalent.

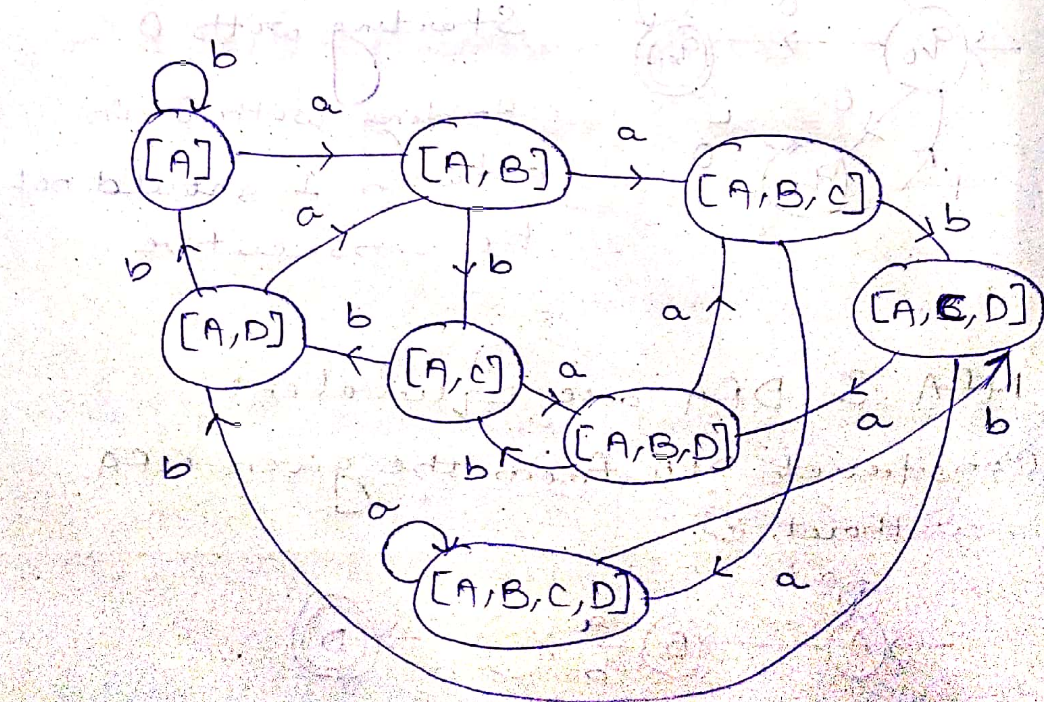
Construct DFA from the given NFA without ε.



State/Σ	a	b
--> A	{A, B}	A
B	C	C
C	D	D
D(F)	ϕ	ϕ

$\delta'([A,B], a)$
$= \delta(A,a) \cup \delta(B,a)$
$= \{A, B, C\}$

State/Σ	a	b
--> [A]	[A, B]	[A]
[A, B]	[A, B, C]	[A, C]
[A, C]	[A, B, D]	[A, D]
[A, B, C]	[A, B, C, D]	[A, C, D]
[A, B, D]	[A, B, C]	[A, C]
[A, D]	[A, B]	[A]
[A, C, D]	[A, B, D]	[A, D]
[A, B, C, D]	[A, B, C, D]	[A, C, D]



Dt - 03/02/21

Construct DFA from the given NFA without ε

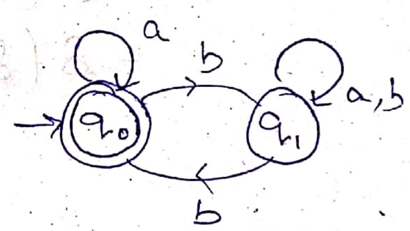
Construct a deterministic automation equivalent to given NFA.

State/Σ	a	b
$\to q_0$ (F)	q_0	q_1
q_1	q_1	q_0, q_1

find a deterministic acceptor equivalent to

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

State/Σ	a	b
$\to [q_0]$ (F)	$[q_0]$	$[q_1]$
$[q_1]$	$[q_1]$	$[q_0, q_1]$
$[q_0, q_1]$ (F)	$[q_0, q_1]$	$[q_0, q_1]$



THEOREM

Any transition function δ, input string x & symbol y

$$\delta(q, xy) = \delta(\delta(q, x), y)$$

Any transition function δ & two input string x & y, $\delta(q, xy) = \delta(\delta(q, x), y)$

Proof

We can proof by method of induction for $|y|$.

Let $|y| = 1$ i.e., $y = a$

By known result $\delta(q, xa) = \delta(\delta(q, x), a)$

Let it is true for $|y| = n$

$\delta(q, xy) = \delta(\delta(q, x), y)$

We have to prove that $|y| = n+1$;

L.H.S: $\delta(q, xy)$

$= \delta(q, xy_1 a)$

$= \delta(q, x_1 a)$

$= \delta(\delta(q, x_1), a)$

$= \delta(\delta(q, xy_1) a)$

$= \delta(\delta(\delta(q, x) y_1) a) = \delta(\delta((q, x), y_1 a)$

$= \delta(\delta(q, x), y)$

R.H.S: $\delta(\delta(q, x), y)$

$= \delta(\delta(q, x), y_1 a)$

$= \delta(\delta(\delta(q, x) y_1) a)$

$= \delta(\delta((q, x) y_1 a)$

$= \delta(\delta(q, x), y)$

Ex 2 : Construct a deterministic automation equivalent to given NFA.

State/Σ	a	b
$\to q_0$	$[q_0, q_1]$	q_0
q_1	q_2	q_1
q_2	q_3	q_3
q_3 (F)		q_2

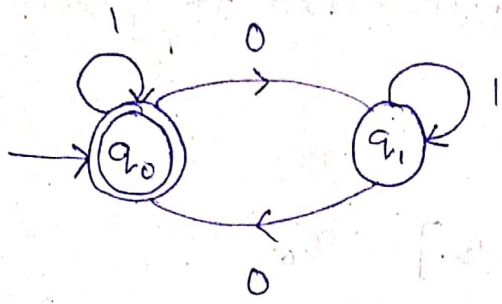
Sol^n: Let $Q = \{q_0, q_1, q_2, q_3\}$

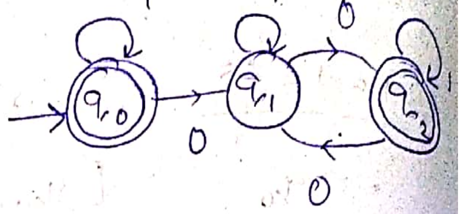
Then the deterministic automation M_1 equivalent to M is given by

$$M_1 = (2^Q, \{a, b\}, \delta, [q_0], F)$$

State/Σ	a	b
$\to [q_0]$	$[q_0, q_1]$	$[q_0]$
$[q_0, q_1]$	$[q_0, q_1, q_2]$	$[q_0, q_1]$
$[q_0, q_1, q_2]$	$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_3]$
$[q_0, q_1, q_3]^*$	$[q_0, q_1, q_2]$	$[q_0, q_1, q_2]$
$[q_0, q_1, q_2, q_3]^*$	$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_2, q_3]$

Minimization of FA

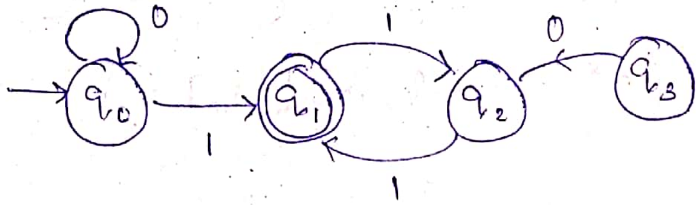




Even number of zeroes.

* When $\delta(q_1, a)$ & $\delta(q_2, a)$ will go to final / nonfinal state then they are said to be Equivalent.

Unreachable state



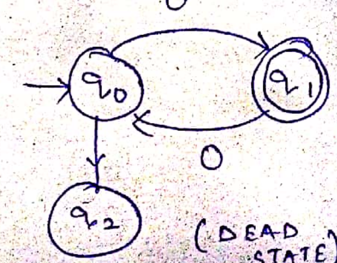
Language: Any no. of zeros followed by odd no. of 1.

The states that are not reachable from the initial state for any input string.

Here ~~are~~ $\{q_3\}$ ~~are~~ is the unreachable state.

Dead State

It is a non-final state where there is no outgoing transaction except itself.

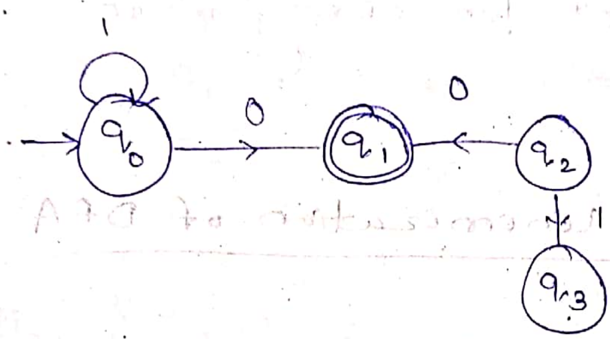


(DEAD STATE)

Dt - 04/02/21

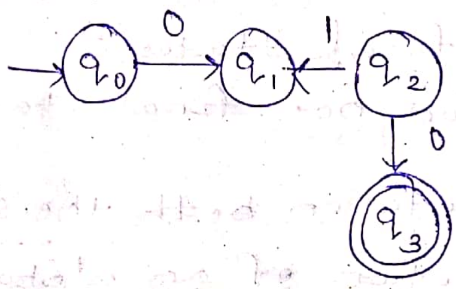
NOTE :

Equivalence : Two state q_0 & q_1 said to be equivalent if both $\delta(q_0, x)$ & $\delta(q_1, x)$ both are final state or both ~~one~~ of them are non final state.



$\delta(q_0, 0) = q_1$
$\delta(q_2, 0) = q_1$
$\delta(q_0, 1) = q_0$
$\delta(q_2, 1) = q_3$

$\{q_0, q_2\}$ are said to be equivalent.
It is also called as Zeroth equivalent.



$\delta(q_0, 0) = q_1$ (Non-final)
$\delta(q_2, 0) = q_3$ (final)

$\{q_0, q_2\}$ are not equivalent.

Let $\{q_0, q_1, q_2\}$ $\{q_3, q_4\}$ are k^{th} equivalent because they we in same group.

$k+1^{th}$ equivalent

$\{q_0, q_1\}$ are said to be $k+1^{th}$ equivalent

Scanned with CamScanner

- k^{th} equivalent
- $\delta(q_0, a)$ & $\delta(q_1, a)$ result state lies in same group.

Two state q_0 & q_1 said to be $(k+1)$ equivalent if they are k equivalent & both $\delta(q_0, x)$ & $\delta(q_1, x)$ are also k-equivalent for every a in alphabet.

Algorithms Minimization of DFA

STEP 1: Remove all unreachable states (The states which can never be reached from initial state)

STEP 2: Divide all vertices into two sets i.e., One set of final states & Second one for non-final states.

STEP 3: for each state in both the sets find the transition of an alphabet (only one alphabet). If the transition state is the element of another set or class then separate these states from that set & make another class.

STEP 4: Repeat the Step 3 until all the classes are made.

STEP 5: Repeat step 3 & 4 for all alphabets.

STEP 6: After completion of all step draw the DFA.

1.

State / Σ	0	1
$\rightarrow q_0$	q_1	q_5
q_1	q_6	q_2
q_2 (f)	q_0	q_2
q_3	q_2	q_6
q_4	q_7	q_5
q_5	q_2	q_6
q_6	q_6	q_4
q_7	q_6	q_2

$\Pi_0 = \{ \{q_2\}, \{q_0, q_1, q_3, q_4, q_5, q_6, q_7\} \}$

$\Pi_1 = \{ \{q_2\}, \{q_0, q_4, q_6\}, \{q_1, q_3, q_5, q_7\} \}$

$= \{ \{q_2\}, \{q_0, q_4, q_6\}, \{q_1, q_7\}, \{q_3, q_5\} \}$

$\Pi_2 = \{ \{q_2\}, \{q_0, q_4\}, \{q_6\}, \{q_1, q_7\}, \{q_3, q_5\} \}$

$\Pi_3 = \{ \{q_2\}, \{q_0, q_4\}, \{q_6\}, \{q_1, q_7\}, \{q_3, q_5\} \}$

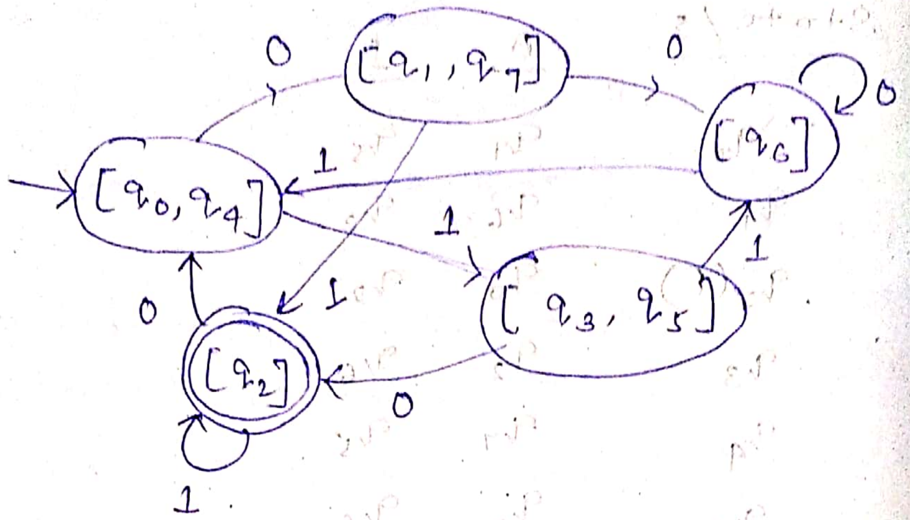
$M = (Q', \{0, 1\}, \delta', q_0', F')$

$Q' = \{ [q_2], [q_0, q_4], [q_6], [q_1, q_7], [q_3, q_5] \}$

$q_0' = \{[q_0, q_4]\}$

$F' = \{ [q_2] \}$

$\delta' =$



State/Σ	0	1
$\rightarrow [q_0, q_4]$	$[q_1, q_7]$	$[q_3, q_5]$
$[q_1, q_7]$	$[q_6]$	$[q_2]$
$[q_3, q_5]$	$[q_2]$	$[q_6]$
$[q_6]$	$[q_6]$	$[q_0, q_4]$
$[q_2]$(F)	$[q_0, q_4]$	$[q_2]$

2.

State/Σ	a	b
$\rightarrow q_0$	q_0	q_3
q_1	q_2	q_5
q_2	q_3	q_4
q_3	q_0	q_5
q_4	q_0	q_6
q_5	q_1	q_4
q_6(F)	q_1	q_3

$\Pi_0 = \{\{q_6\}, \{q_0, q_1, q_2, q_3, q_4, q_5\}\}$

$\Pi_1 = \{\{q_6\}, \{q_0, q_1, q_2, q_3, q_5\}, \{q_4\}\}$

$\Pi_2 = \{\{q_6\}, \{q_0, q_1, q_3\}, \{q_2, q_5\}, \{q_4\}\}$

$\Pi_3 = \{\{q_6\}, \{q_0\}, \{q_1, q_3\}, \{q_2, q_5\}, \{q_4\}\}$

$= \{\{q_6\}, \{q_0\}, \{q_1\}, \{q_3\}, \{q_2, q_5\}, \{q_4\}\}$

$\Pi_4 = \{\{q_6\}, \{q_0\}, \{q_1\}, \{q_3\}, \{q_2\}, \{q_5\}, \{q_4\}\}$

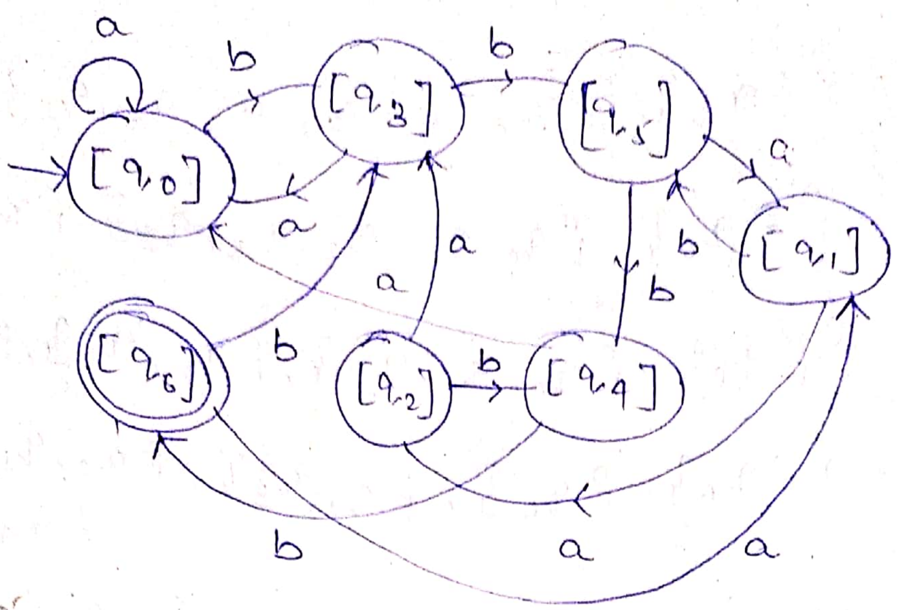
$\Pi_5 = \{\{q_6\}, \{q_0\}, \{q_1\}, \{q_3\}, \{q_2\}, \{q_5\}, \{q_4\}\}$

$M = (Q', \{a, b\}, \delta', q_0', F')$

$Q' = \{[q_6], [q_0], [q_1], [q_3], [q_2], [q_5], [q_4]\}$

$q_0' = \{[q_0]\}$ $\quad F' = \{[q_6]\}$

State/Σ	a	b
$\rightarrow [q_0]$	$[q_0]$	$[q_3]$
$[q_3]$	$[q_0]$	$[q_5]$
$[q_5]$	$[q_1]$	$[q_4]$
$[q_1]$	$[q_2]$	$[q_5]$
$[q_4]$	$[q_0]$	$[q_6]$
$[q_2]$	$[q_3]$	$[q_4]$
$[q_6]$(F)	$[q_1]$	$[q_3]$



b

3.

State/Σ	a	b
$\to q_0$	q_1	q_0
q_1	q_0	q_2
q_2	q_3	q_1
q_3 (F)	q_3	q_0
q_4	q_3	q_5
q_5	q_6	q_4
q_6	q_5	q_6
q_7	q_6	q_3

$$\Pi_0 = \{\{q_3\}, \{q_0, q_1, q_2, q_4, q_5, q_6, q_7\}\}$$

$$\Pi_1 = \{\{q_3\}, \{q_0, q_1, q_5, q_6\}, \{q_2, q_4, q_7\}\}$$

$$= \{\{q_3\}, \{q_0, q_1, q_5, q_6\}, \{q_2, q_4\}, \{q_7\}\}$$

$$\Pi_2 = \{\{q_3\}, \{q_0, q_6\}, \{q_1, q_5\}, \{q_2, q_4\}, \{q_7\}\}$$

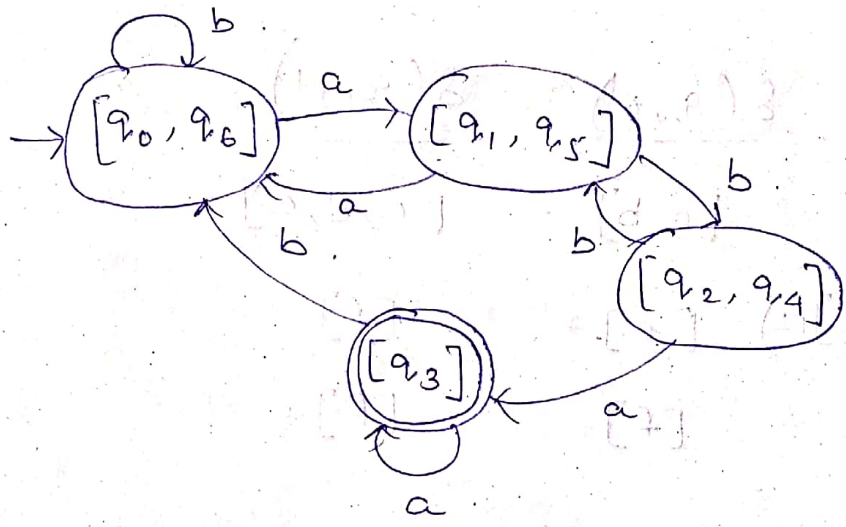
$$\Pi_3 = \{\{q_3\}, \{q_0, q_6\}, \{q_1, q_5\}, \{q_2, q_4\}, \{q_7\}\}$$

State/Σ	a	b
$\rightarrow[q_0, q_6]$	$[q_1, q_5]$	$[q_0, q_0]$
$[q_1, q_5]$	$[q_0, q_0]$	$[q_2, q_4]$
$[q_2, q_4]$	$[q_3]$	$[q_1, q_5]$
$[q_3]$(F)	$[q_3]$	$[q_0]$

$M = (Q', \{a, b\}, \delta', q_0', F')$

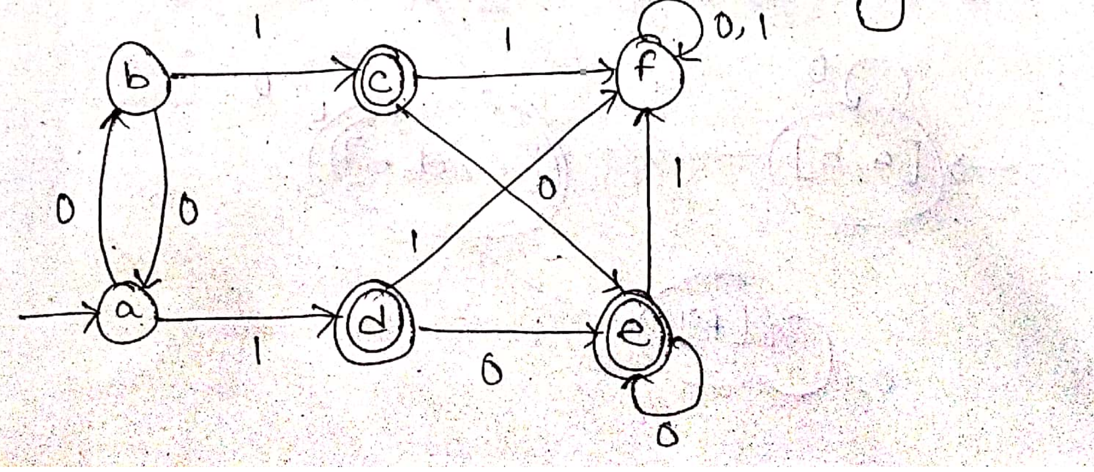
$Q' = \{ [q_3], [q_0, q_6], [q_1, q_5], [q_2, q_4],$

$q_0' = \{[q_0, q_6]\}$ $F' = \{[q_3]\}$



Dt- 09|02|21

Let us considere the following DFA:



q	$\delta(q,0)$	$\delta(q,1)$
a	b	c
b	a	d
c	e	f
d	e	f
e	e	f
f	f	f

$\Pi_0 = \{\{c, d, e\}, \{a, b, f\}\}$

$\Pi_1 = \{\{c, d, e\}, \{a, b\}, \{f\}\}$

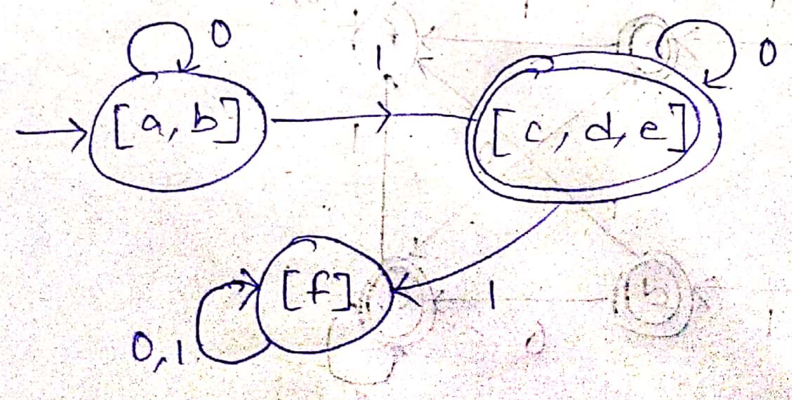
$\Pi_2 = \{\{c, d, e\}, \{a, b\}, \{f\}\}$

q	$\delta(q,0)$	$\delta(q,1)$
$\rightarrow$ [a, b]	[a, b]	[c, d, e]
[c, d, e] (f)	[e]	[f]
[f]	[f]	[f]

$M = (Q', \delta', \{0, 1\}, q_0', F')$

$Q' = \{[a, b], [c, d, e], [f]\}$

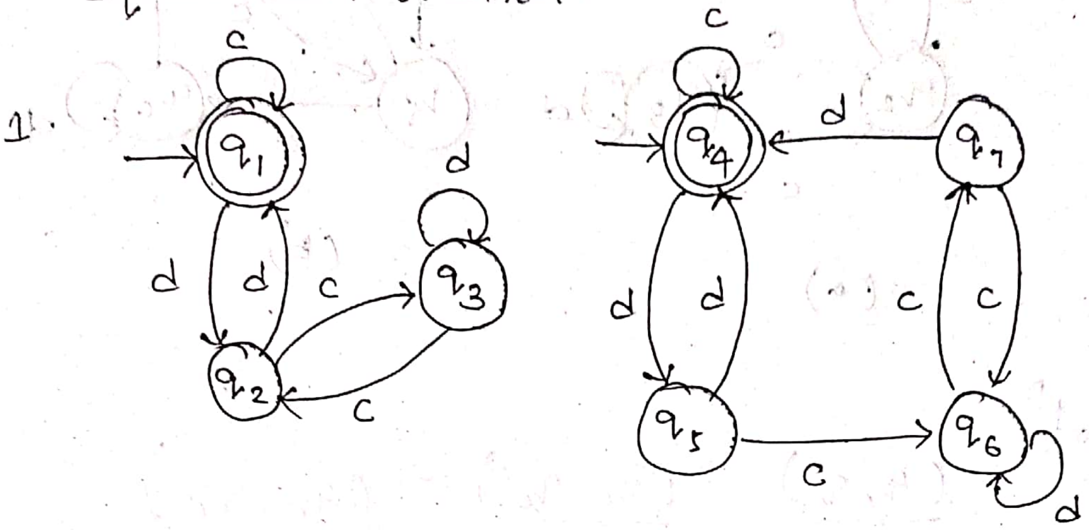
$q_0' = \{[a, b]\}$, $F' = \{[c, d, e]\}$



Equivalence of Two Finite Automata

Check whether these two given FA are equivalent or not.

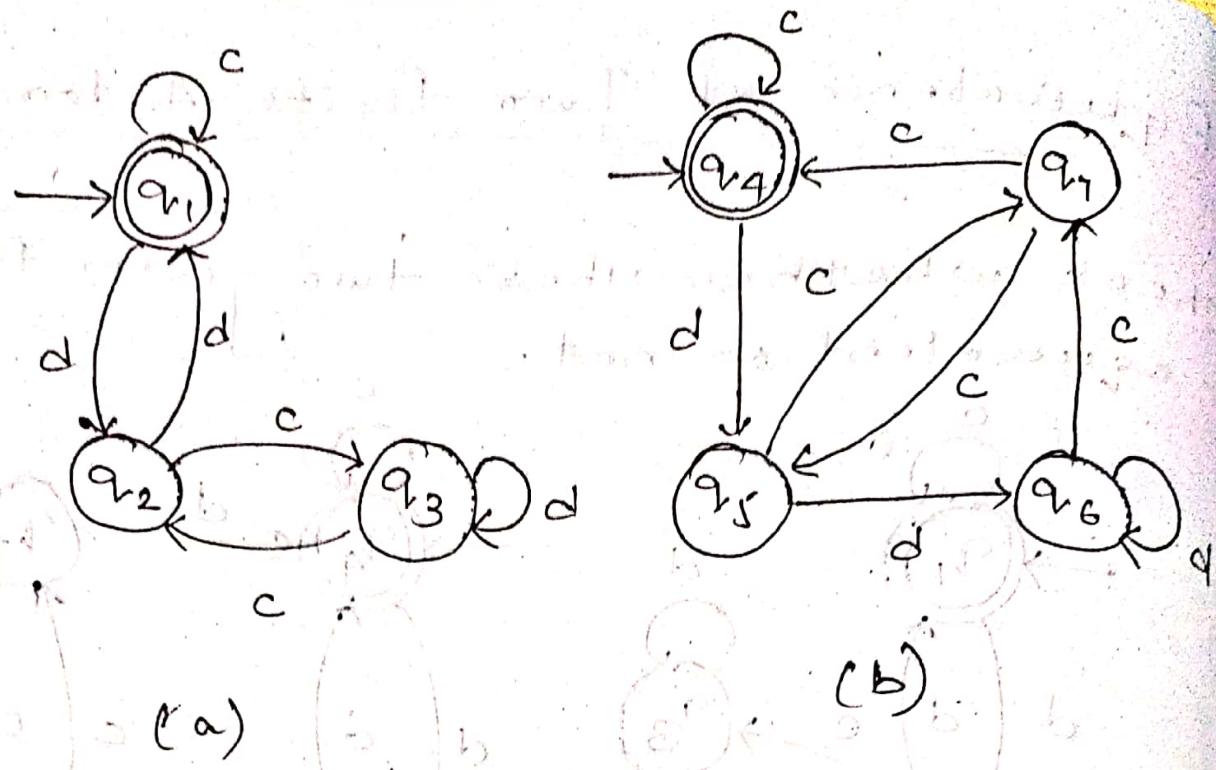
1.



Sol^n

(q, q')	(q_c, q_c')	(q_d, q_d')
(q_1, q_4)	(q_1, q_4)	(q_2, q_5)
(q_2, q_5)	(q_3, q_6)	(q_1, q_4)
(q_3, q_6)	(q_2, q_7)	(q_3, q_6)
(q_2, q_7)	(q_3, q_6)	(q_1, q_4)

As in the pair (q, q') are both final state or both are non-final state so two FA M & M' are equivalent.

2. 

Sol^n

(q, q')	(q_c, q_c')	(q_d, q_d')
(q_1, q_4)	(q_1, q_4)	(q_2, q_5)
(q_2, q_5)	(q_3, q_7)	(q_1, q_6)
(q_1, q_6)		

As in the pair (q_1, q_6), q_1 is final state in M & q_6 is non final state in M', the given two FA is not equivalent.