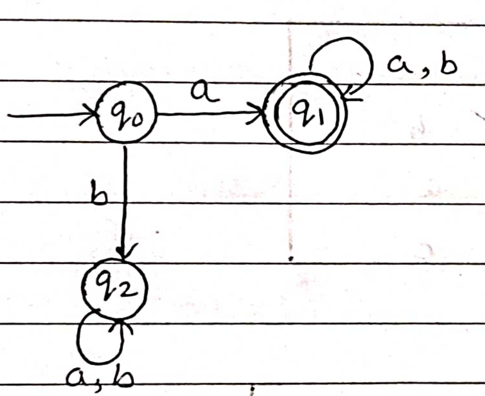
Deterministic Finite Automata (DFA) :-

The finite automata are called deterministic finite automata if the machine is reads an input string one symbol at a time.

- * A state should have paths for all its input symbol.
- * Only one path for specific input from current state to the next state
- * DFA contains multiple final states.
- * DFA does not accept the null move. i.e. in DFA we cannot change state without any input character.

Eg:-



* DFA with $\Sigma = (a, b)$ accepts all strings starting with 0.

→ here, q_0 is starting state with both the inputs a and b.

→ q_1 is final state "double circled" and contains a self loop of both the inputs of a and b.

→ q_2 is deadstate, trapstate or rejectstate.

↳ For a, it goes to $q_0 \rightarrow q_1$ and string is accepted if a is the starting symbol.

↳ If b is the starting symbol then $q_0 \rightarrow q_2$ which is reject state means string is not accepted.

Formal Tuple Representation of DFA :-

$\delta, Q, \Sigma, q_0, F$

$\boxed{\delta\text{(Transition Function) :- } \delta : Q \times \Sigma \rightarrow Q}$

Eg :- $Q = \{a, b, c\}$
$\Sigma = \{0, 1\}$
$q_0 = \{a\}$
$F = \{c\}$

Transition Table :-

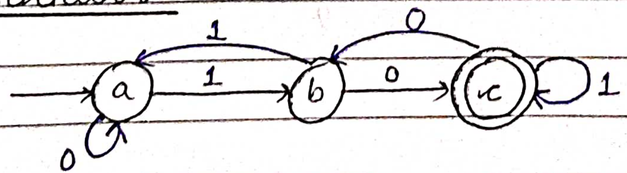
State	Input 0	Input 1
a	a	b
b	c	a
c	b	c

Transition Function :-

$\boxed{\delta : Q \times \Sigma \rightarrow Q}$

$\delta(a,0) = a$
$\delta(a,1) = b$
$\delta(b,0) = c$
$\delta(b,1) = a$
$\delta(c,0) = b$
$\delta(c,1) = c$

Representation :-

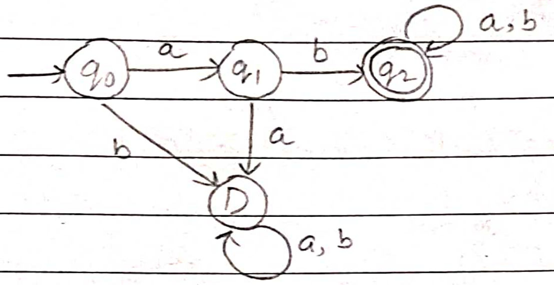


Example of DFA

1 Strings starting with ab over a, b.

⇒ Create a regular ~~e~~expression. RE i.e $ab(a+b)^*$ → either a, b or combination of ab or ε or null move

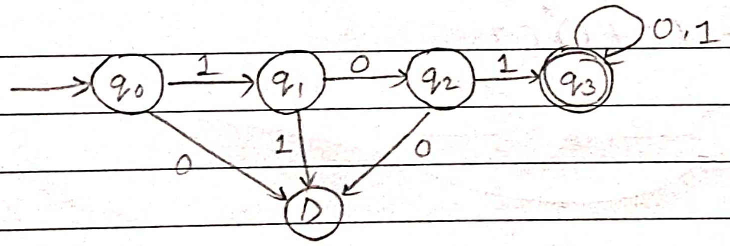
Accepted Strings → ab, aba, abab, abb, abba



2. Strings starting with ~~101~~ 101 over $\Sigma = \{0,1\}$

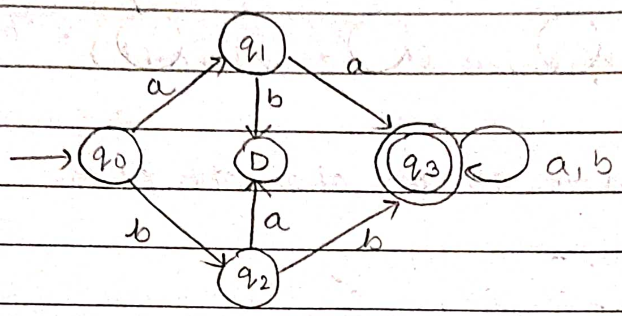
$RE = 101(0+1)^*$

Accepted Strings :- 101, 1010, 101011, 10101010010



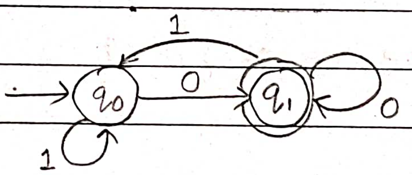
3. Set of all strings starting with aa or bb over $\Sigma = \{a, b\}$

R.E = $(aa+bb)(a+b)^*$



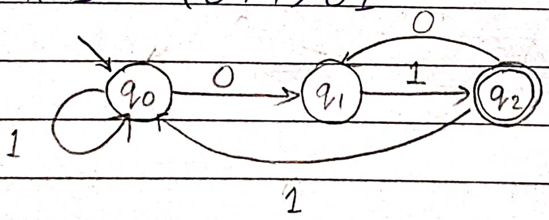
4. Set of all strings ending with 0 over $\Sigma(0,1)$

R.E = $(0+1)^*0$



5. Set of all strings ending with 01 over $\Sigma = (0,1)$

R.E = $(0+1)^*01$

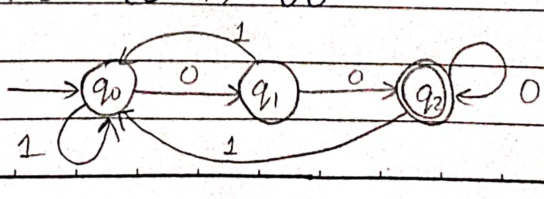


6. Set of all string ending with 00 over $\Sigma = (0,1)$

R.E = $(0+1)^*\ 00$

Accepted = 00, 0100, 11100, 000 0100100, 110100

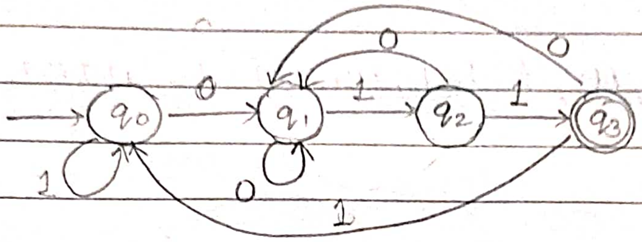
Rejected - 001, 0, 1, 00010



7. Ending with 011 over input $\Sigma = \{0,1\}$

$$R.E = (0+1)^{*} 011$$

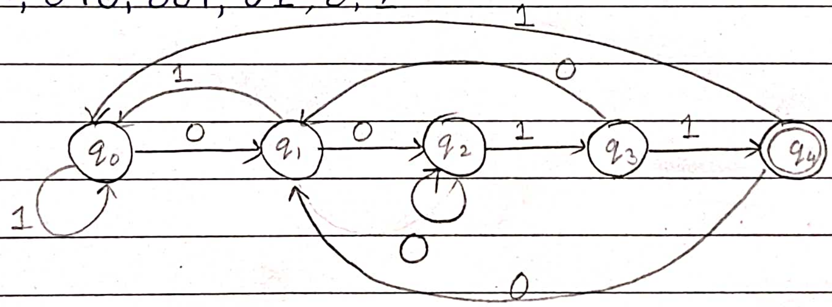
Accept = 0011, 001011, 011, 1011, 11101011, 111011, 001011, 10101011
Reject = 010, 0, 1, 0110, 01, 11



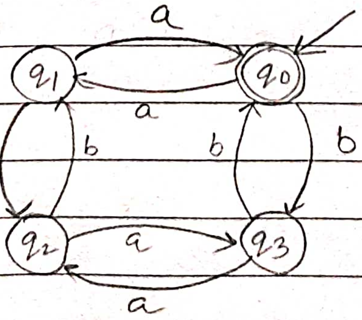
8. Accept string ending with 0011 in $\Sigma = \{0,1\}$

$$R.E = (0+1)^{*} 0011$$

Accept = 0011, 00011, 10011, 010011, 01000011, ~~0~~1100011, 10110011, 0011001
Reject = 011, 010, 001, 01, 0, 1



9. Design FA which accept even no. of a's and even no. of b's over $\Sigma = (a,b)$



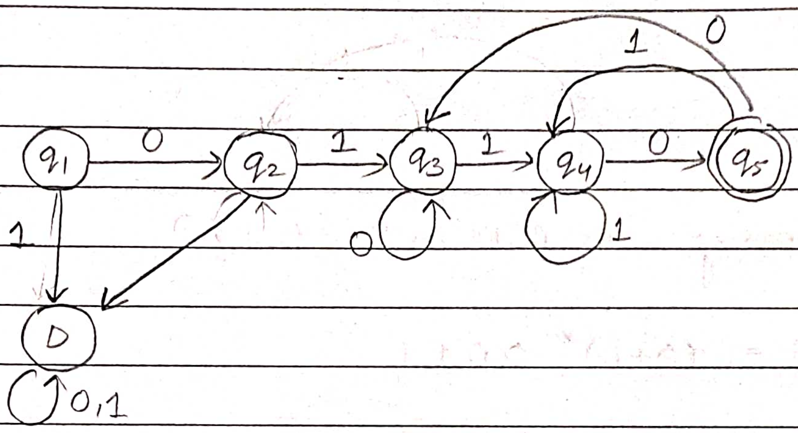
Accept ⇒ abab, aabb, abba
Reject ⇒ aab, ab, a, b, abbab

10. DFA for the following language (0,1) Input :-
L = Start 01, end with 10

$$RE = 01\,(0+1)^*\,10$$

Accept :- 0110, 01010, 01110, 010110, 011010, 0111010110010

Reject :- 0101, 01, 10, 01101, 1010, 1110, 11010110


q1
q2
q3
q4
q5
D
0
1
1
0
1
0
0
1
1
0,1