Theory of Computation

State Elimination Method:

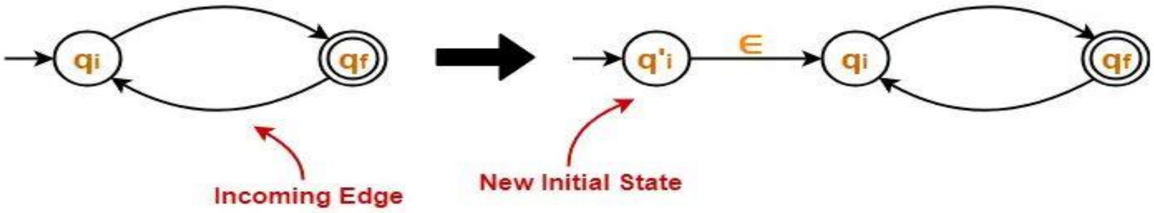
Application: Used to convert DFA/NFA/Ɛ-NFA into Regular Expression

The following are the Steps.

Step 1:

The initial state of the FA must not have any incoming edge. If there exists any incoming edge to the initial state, then create a new initial state having no incoming edge to it and an outgoing edge to the old start state with Ɛ-transition. The initial state before is now normal state with added incoming Ɛ-transition.

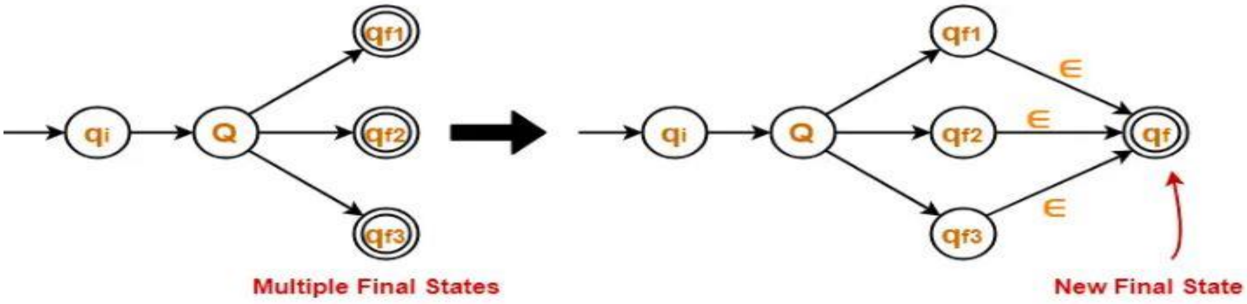
Example:



Step 2:

There must exist only one final state in the FA. If there exist multiple final states in the FA, then convert all the final states into non-final states and create a new single final state and add outgoing Ɛ-transition to new and only final state.

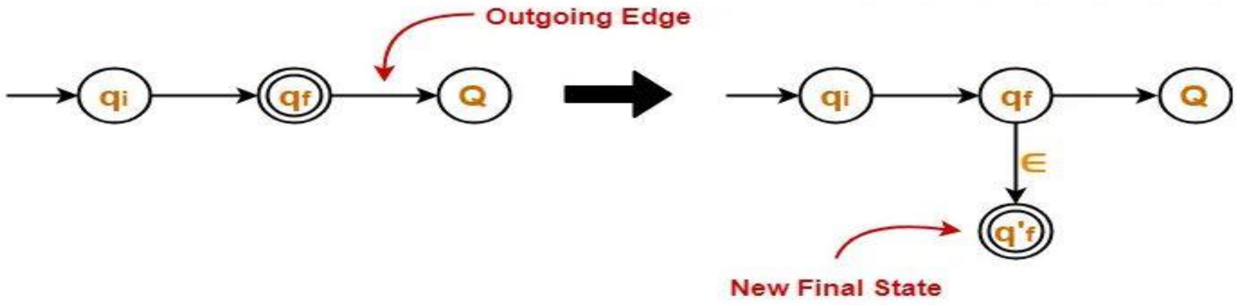
Example:



Step 3:

The final state of the FA must not have any outgoing edge. If there exists any outgoing edge from the final state, then create a new final state with no outgoing edges and an incoming edge from old final state of Ɛ-transition. Old final state is transformed into normal state with the added transition of Ɛ.

Example:



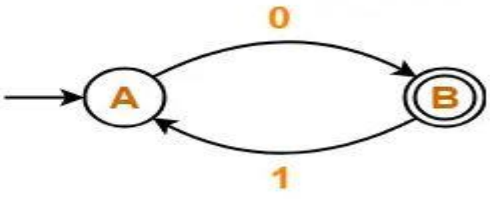
Step 4:

- Eliminate all the intermediate states one by one.
- These states may be eliminated in any order.

In the end,

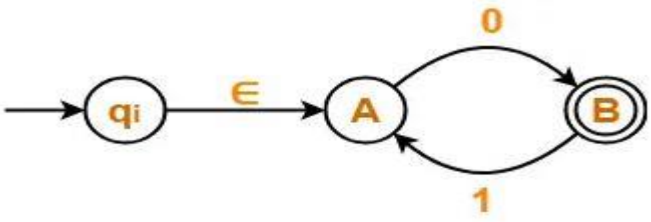
- Only an initial state going to the final state will be left.
- The cost of this transition is the required regular expression.

Question: Find regular expression for the following DFA



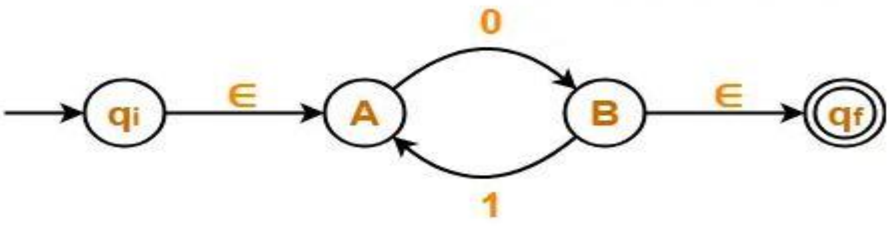
Solution:

Step-1:



Step-2: No multiple final state, so not applicable.

Step-3:



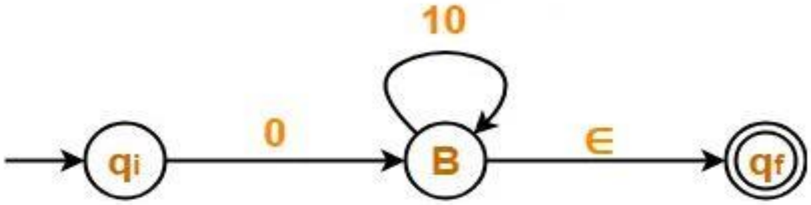
Step-4:

Now, we start eliminating the intermediate states.

First, let us eliminate state A.

- There is a path going from state q_i to state B via state A.
- So, after eliminating state A, we put a direct path from state q_i to state B having cost ∈.0 = 0
- There is a loop on state B using state A.
- So, after eliminating state A, we put a direct loop on state B having cost 1.0 = 10.

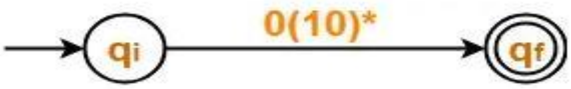
Eliminating state, A, we get-



Now, let us eliminate state B.

- There is a path going from state q_i to state q_f via state B.
- So, after eliminating state B, we put a direct path from state q_i to state q_f having cost 0.(10)*.∈ = 0(10)*

Eliminating state B, we get-

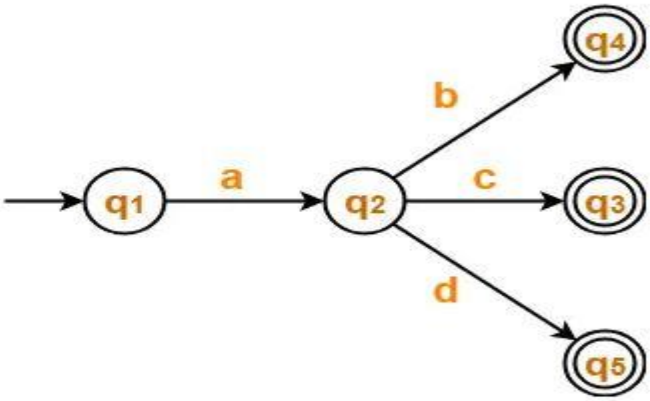


So, the required Regular Expression is = 0(10)*

In the above question,

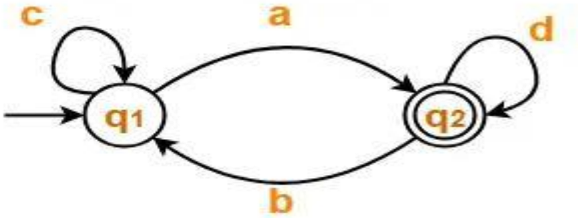
- If we first eliminate state B and then state A, then regular expression would be = (01)*0.
- This is also the same and correct.

Q1. Find regular expression for the following FA



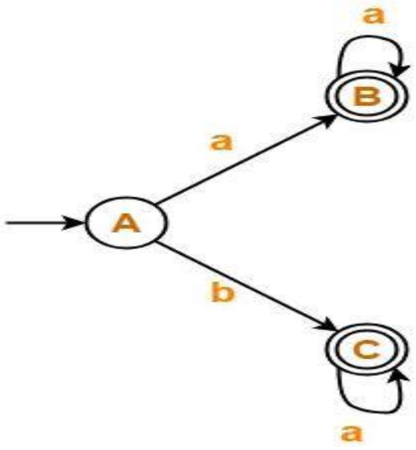
Ans: **Regular Expression = a(b+c+d)**

Q2: Find regular expression for the following FA



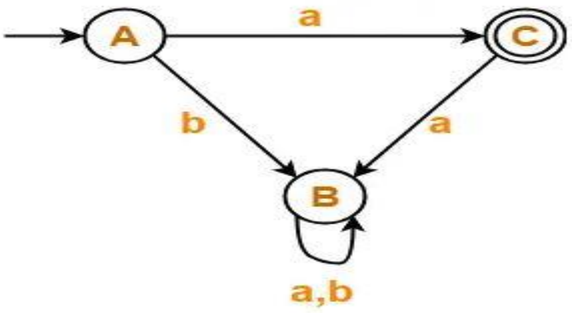
Ans: **Regular Expression = c*a(d+bc*a)***

Q3: Find regular expression for the following FA



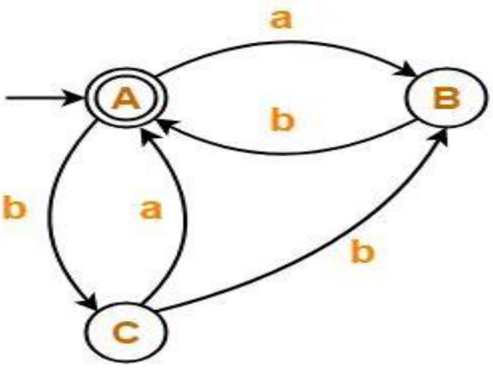
Ans: **Regular Expression = aa* + ba***

Q4: Find regular expression for the following FA



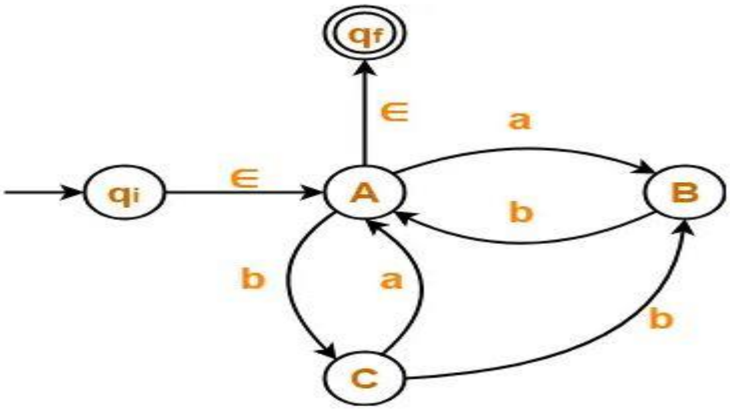
Ans: **Regular Expression = a**

Q5: Find regular expression for the following FA

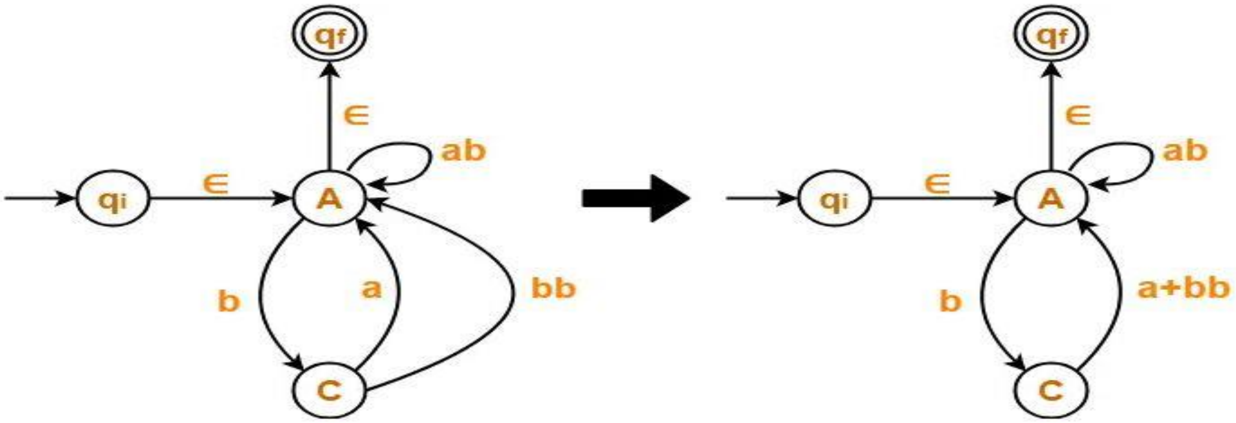


Ans:

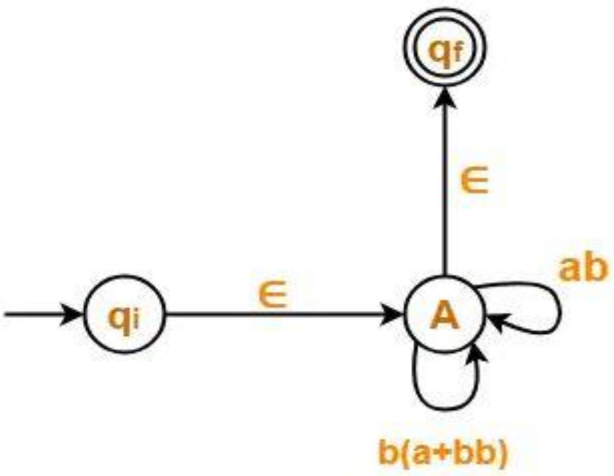
- Since initial state A has an incoming edge, so we create a new initial state q_i.
- Since final state A has an outgoing edge, so we create a new final state q_f.



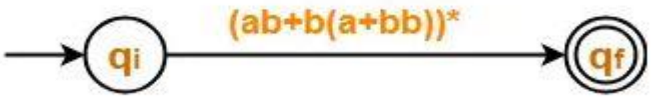
Eliminating state B, we get-



Eliminating state C, we get-



Eliminating state A, we get-



So, Regular Expression = (ab + b(a+bb))*