Compiler Construction

Lexical Analysis

Chapter -3

Compiler Construction (Lexical Analysis)

- ✓ Non-Deterministic Finite Automata with ε (NFA- ε)
- ✓NFA- ε To NFA Conversion
- ✓NFA to DFA Conversion
- ✓Minimization of DFA
- ✓**Convert DFA to Regular Expression**

Convert Deterministic Finite Automata To Regular Expression

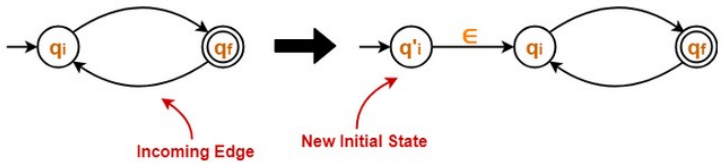
- **Applying Arden's Theorem**
- **State Elimination Method**

State Elimination Method

Rule : 1

If there exists any incoming edge to the initial state, then create a new initial state having no incoming edge to it.

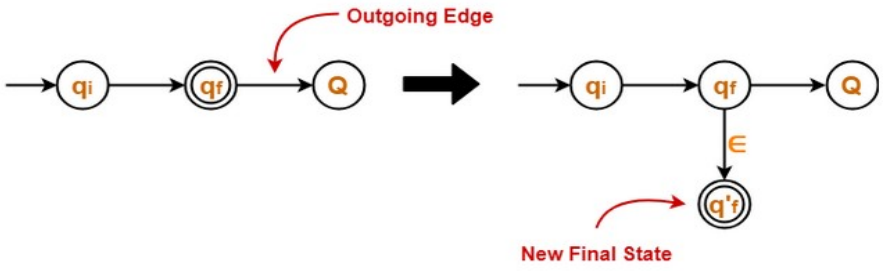
Example-



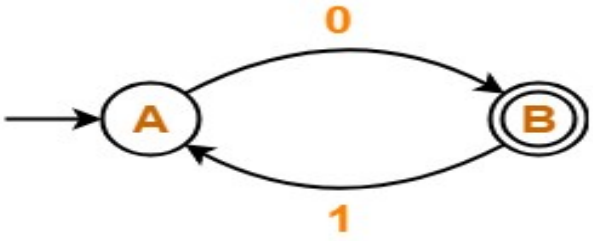
State Elimination Method

Rule : 2

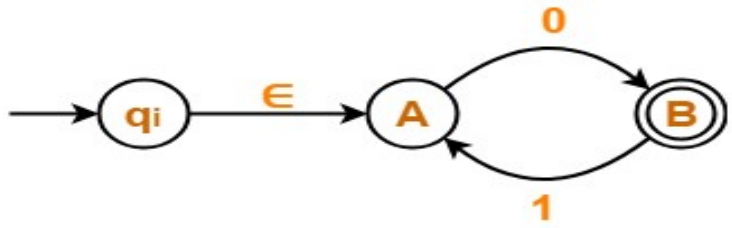
If there exists any outgoing edge from the final state, then create a new final state having no outgoing edge from it.

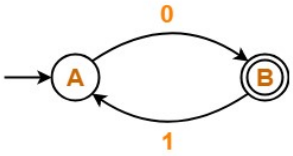


State Elimination Method

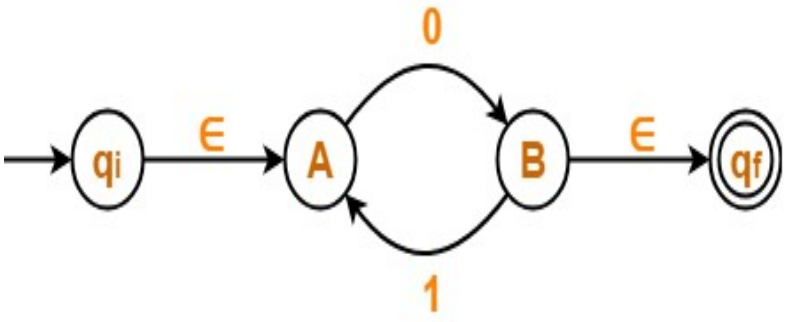


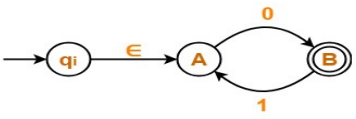
State Elimination Method





State Elimination Method

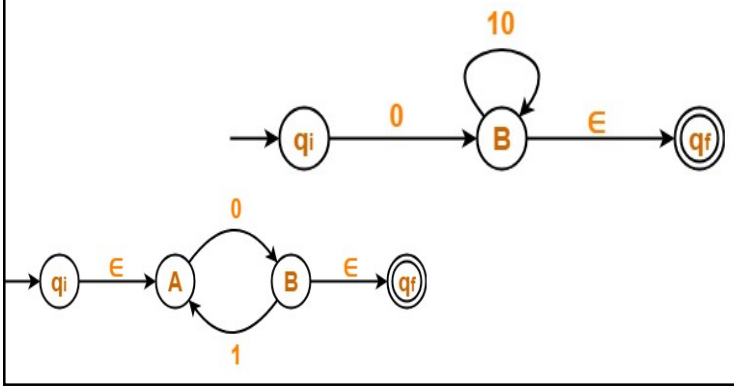




State Elimination Method

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 $\epsilon.0 = 0$
- 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 $1.0 = 10$.

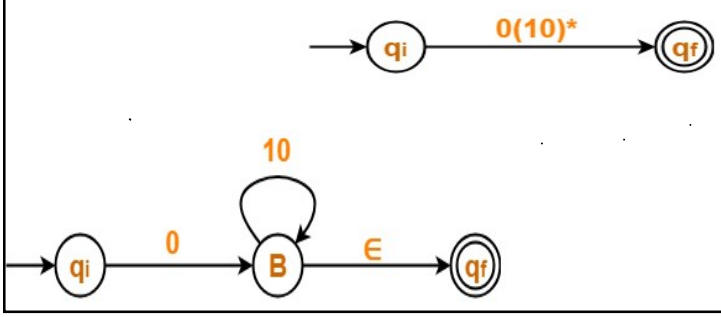


State Elimination Method

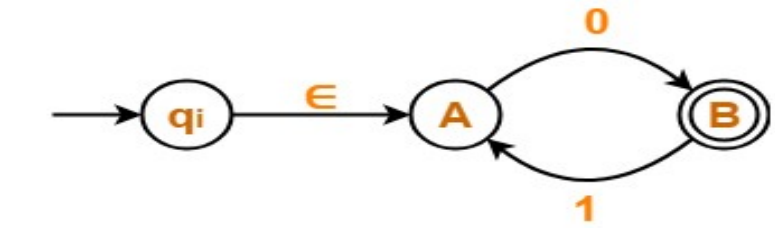
Now, let us eliminate state B.

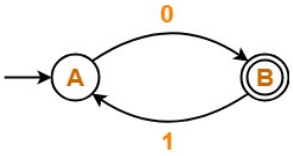
- There is a path going from state q_i to state q_f via state B.
- So, after eliminating state B, we put a direct path from state q_i to state q_f having cost $0.(10)^*.\epsilon = 0(10)^*$

Eliminating state B, we get-

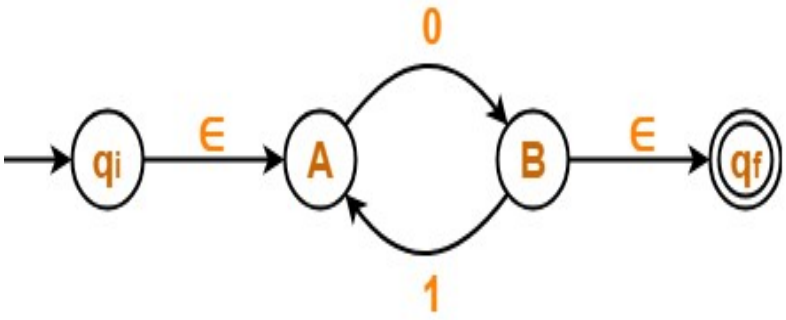


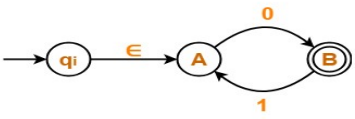
State Elimination Method (Revisited)



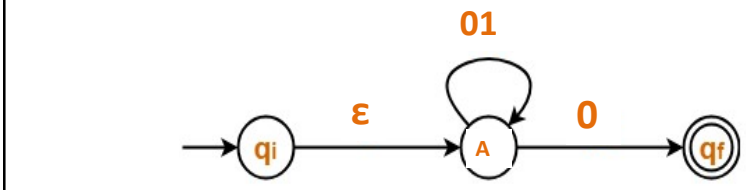


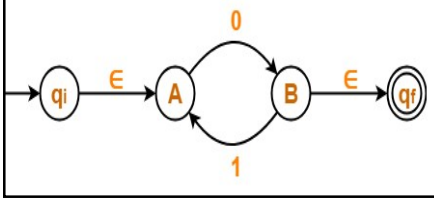
State Elimination Method (Revisited)



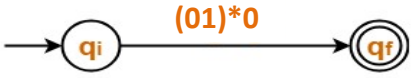


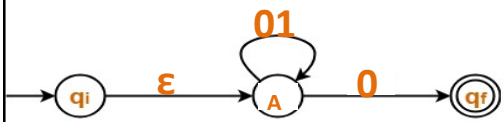
State Elimination Method (Revisited)



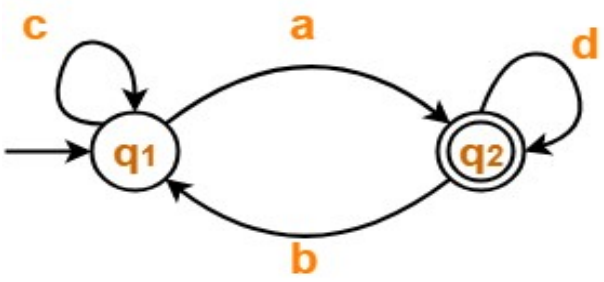


State Elimination Method (Revisited)

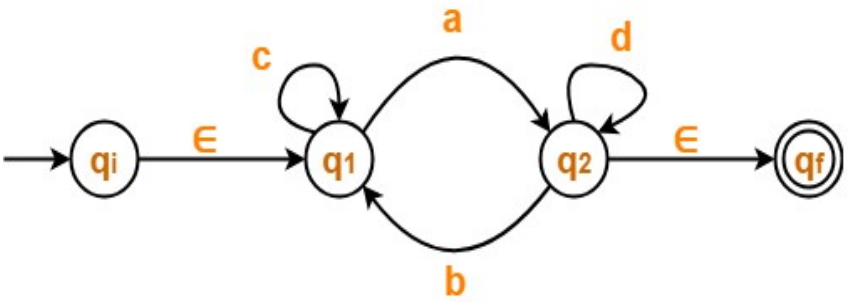


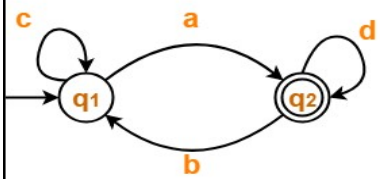


State Elimination Method



State Elimination Method



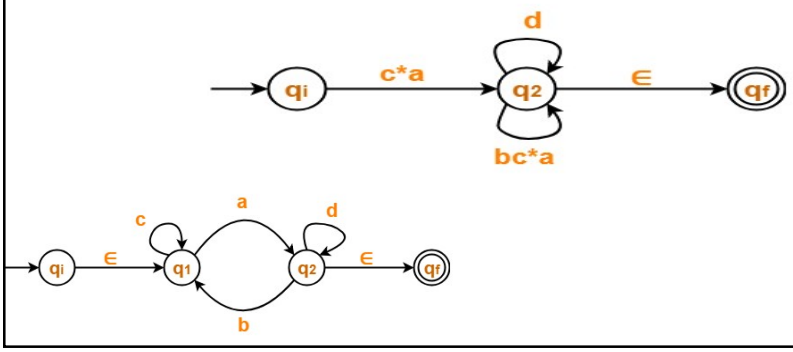


State Elimination Method

First, let us eliminate state q_1.

- There is a path going from state q_i to state q_2 via state q_1 .
- So, after eliminating state q_1, we put a direct path from state q_i to state q_2 having cost $\in.c^*.a = c^*a$
- There is a loop on state q_2 using state q_1 .
- So, after eliminating state q_1 , we put a direct loop on state q_2 having cost $b.c^*.a = bc^*a$

Eliminating state q_1, we get-

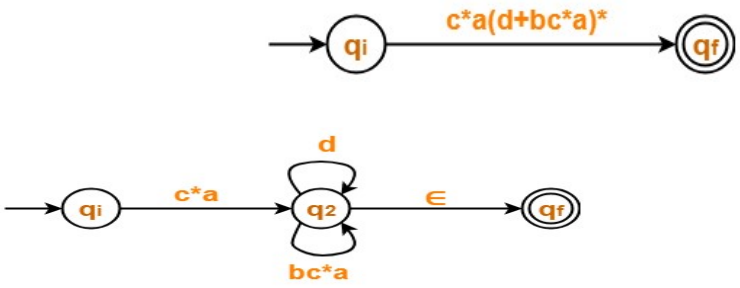


State Elimination Method

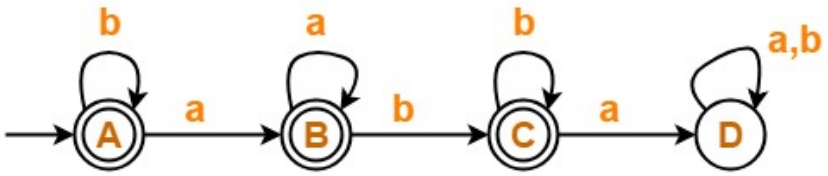
Now, let us eliminate state q_2.

- There is a path going from state q_i to state q_f via state q_2 .
- So, after eliminating state q_2, we put a direct path from state q_i to state q_f having cost $c^*a(d+bc^*a)^*\in = c^*a(d+bc^*a)^*$

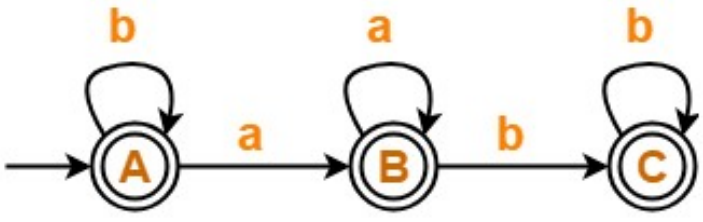
Eliminating state q_2, we get-

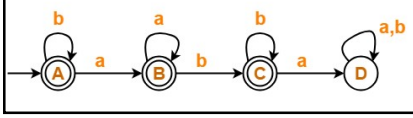


State Elimination Method



State Elimination Method

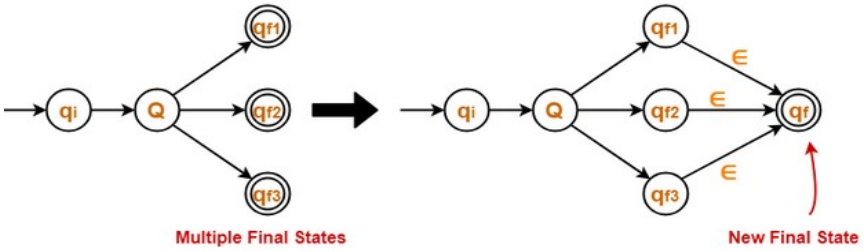




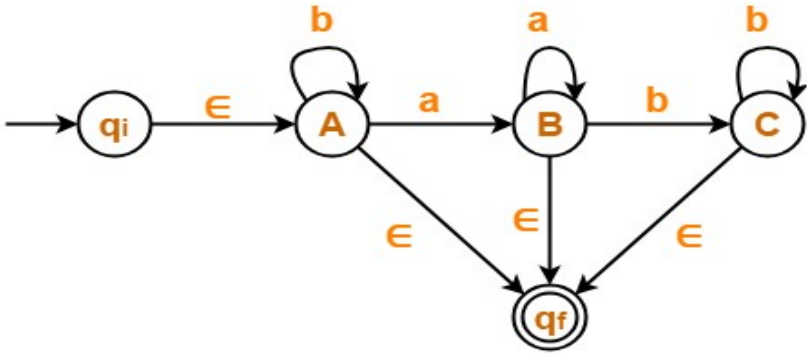
State Elimination Method

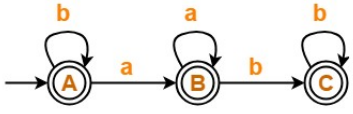
Rule :3

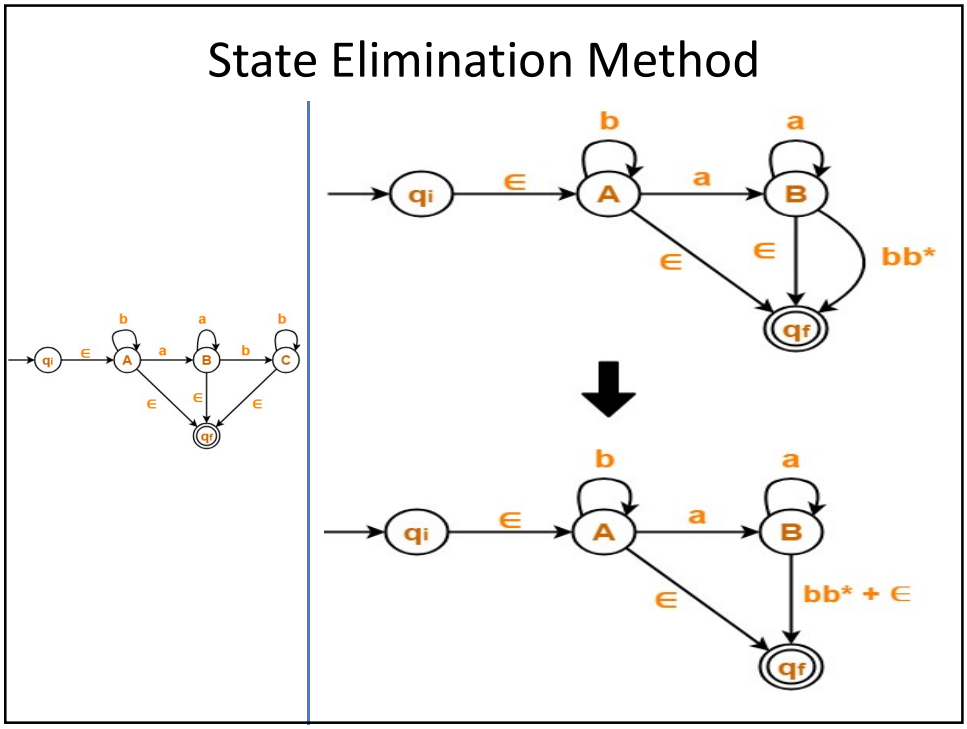
If there exists multiple final states in the DFA, then convert all the final states into non-final states and create a new single final state..

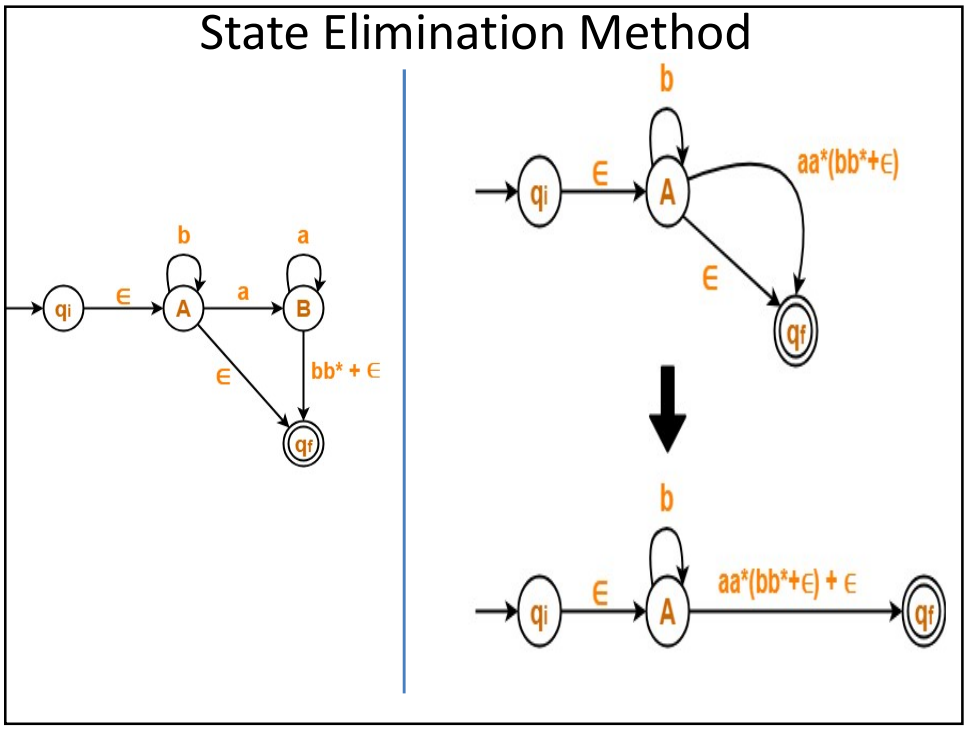


State Elimination Method

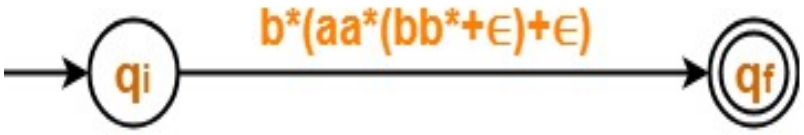


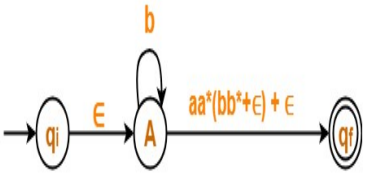



State Elimination Method
qi
A
B
C
qf
b
a
b
ϵ
a
ϵ
ϵ
ϵ
b
a
ϵ
a
ϵ
bb*
b
a
ϵ
a
ϵ
bb* + ϵ

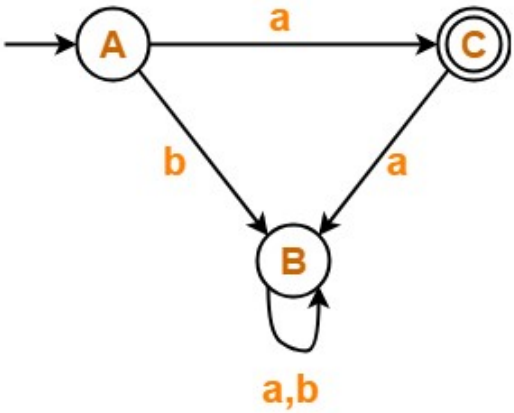

State Elimination Method
qi
A
B
qf
b
a
ϵ
a
ϵ
bb* + ϵ
b
ϵ
aa*(bb*+ϵ)
ϵ
b
ϵ
aa*(bb*+ϵ) + ϵ

State Elimination Method

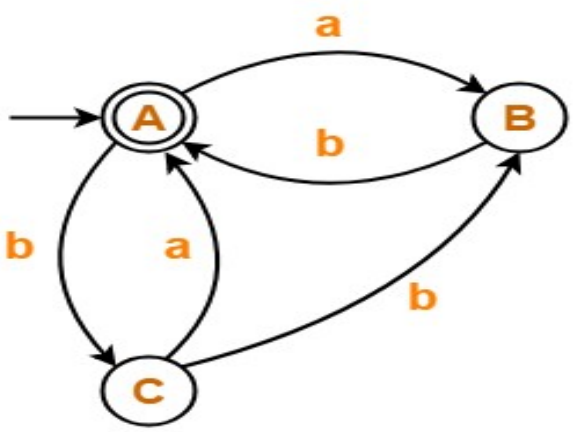




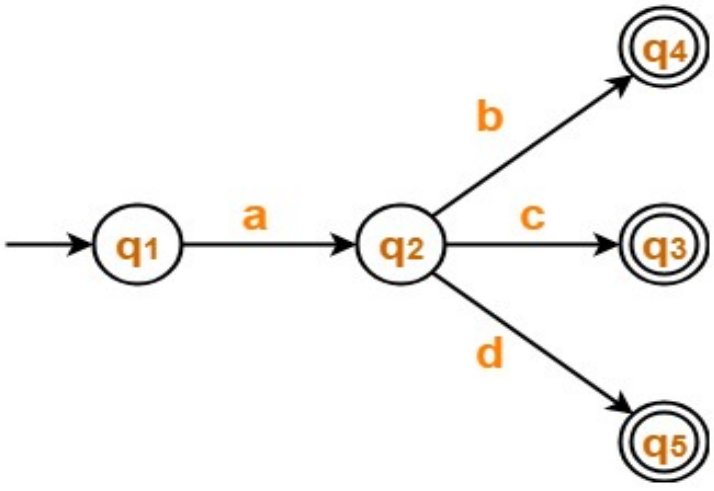
Try Yourself



Try Yourself



Try Yourself



Try Yourself