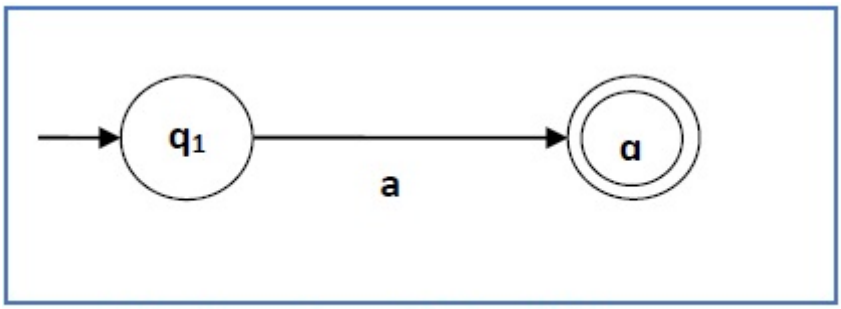
Construction of an FA from an RE

https://www.tutorialspoint.com/automata_theory/constructing_fa_from_re.htm

We can use Thompson's Construction to find out a Finite Automaton from a Regular Expression. We will reduce the regular expression into smallest regular expressions and converting these to NFA and finally to DFA.

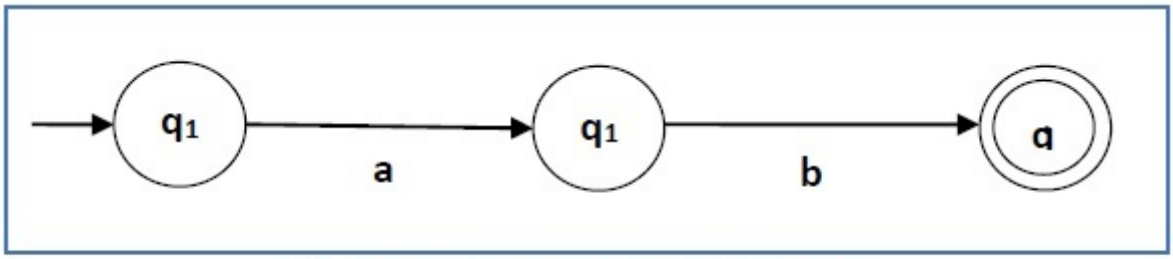
Some basic RA expressions are the following −

Case 1 − For a regular expression ‘a’, we can construct the following FA −



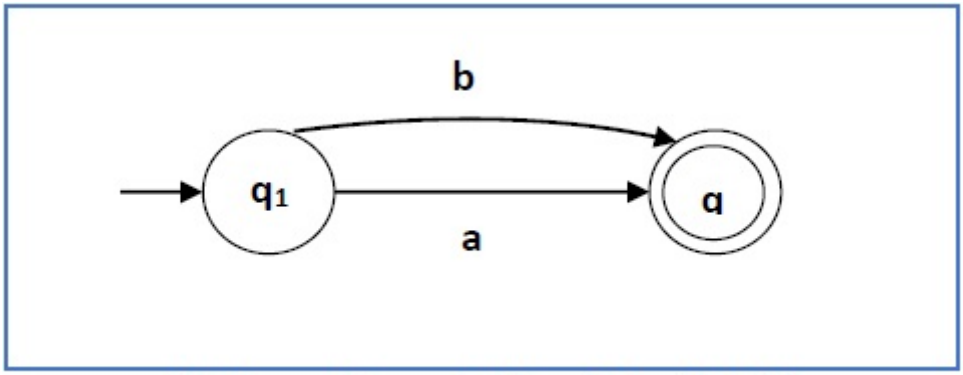
Finite automata for RE = a

Case 2 − For a regular expression ‘ab’, we can construct the following FA −



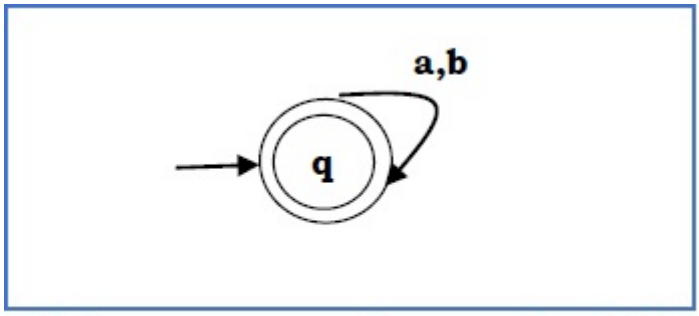
Finite automata for RE = ab

Case 3 − For a regular expression $a + b$, we can construct the following FA −



Finite automata for RE= (a+b)

Case 4 − For a regular expression $a + b$*, we can construct the following FA −



Finite automata for RE= (a+b)*

Method

Step 1 Construct an NFA with Null moves from the given regular expression.

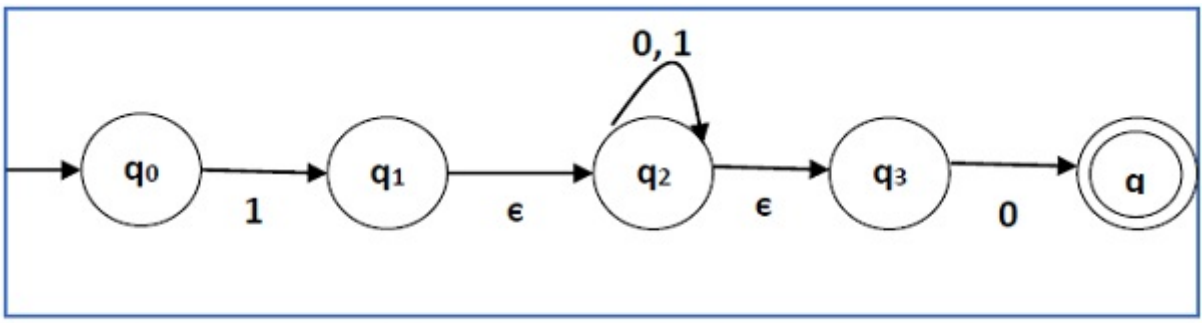
Step 2 Remove Null transition from the NFA and convert it into its equivalent DFA.

Problem

Convert the following RA into its equivalent DFA − 1 0 + 1* 0

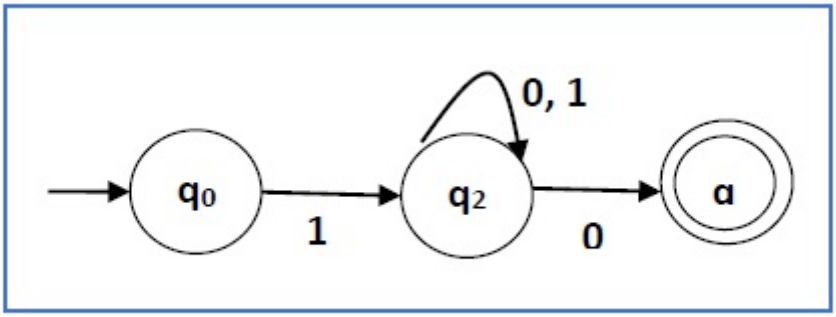
Solution

We will concatenate three expressions "1", "0 + 1*" and "0"



NDFA with NULL transition for RA: 1 (0 + 1)* 0

Now we will remove the **ε** transitions. After we remove the **ε** transitions from the NDFA, we get the following −



NDFA without NULL transition for RA: 1 (0 + 1)* 0

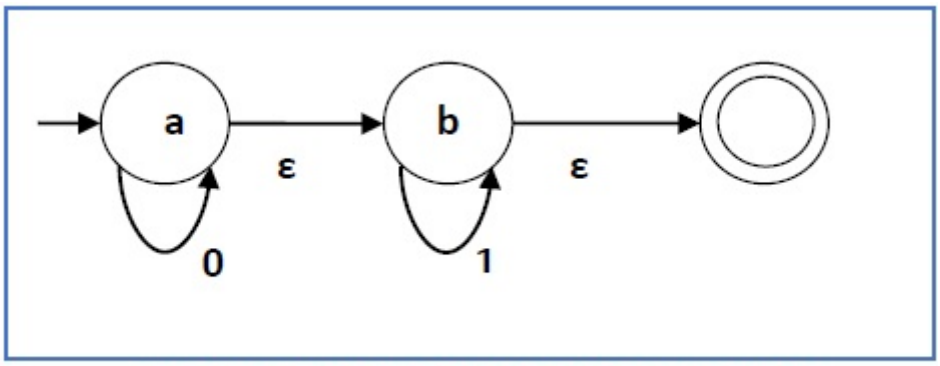
It is an NDFA corresponding to the RE − 1 0 + 1* 0. If you want to convert it into a DFA, simply apply the method of converting NDFA to DFA discussed in Chapter 1.

Finite Automata with Null Moves $NFA - \varepsilon$

A Finite Automaton with null moves $FA - \varepsilon$ does transit not only after giving input from the alphabet set but also without any input symbol. This transition without input is called a **null move**.

An NFA-ε is represented formally by a 5-tuple (Q, ∑, δ, q_0, F), consisting of

- **Q** − a finite set of states
- ∑ − a finite set of input symbols
- **δ** − a transition function $\delta : Q \times \sum \cup \varepsilon \rightarrow 2^Q$
- $\mathbf{q_0}$ − an initial state $q_0 \in Q$
- **F** − a set of final state/states of Q $F \subseteq Q$.



Finite automata with Null Moves

The above $FA - \varepsilon$ accepts a string set − {0, 1, 01}

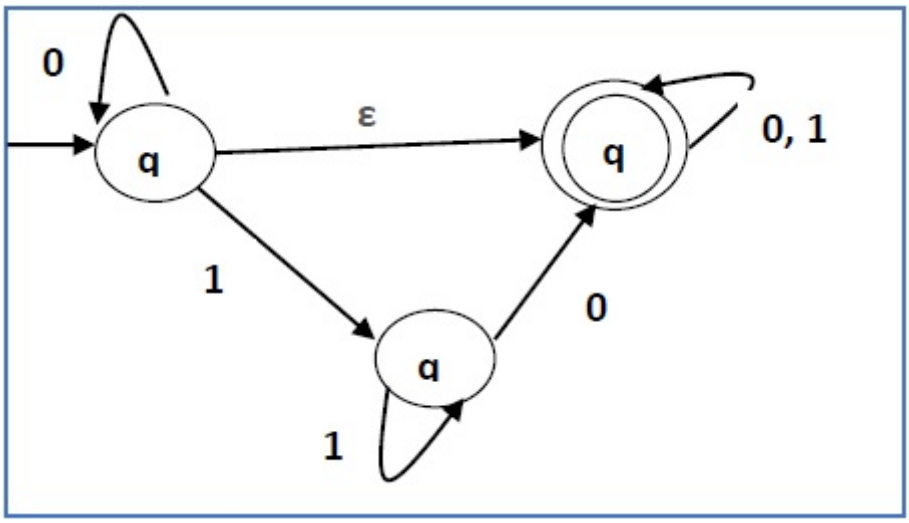
Removal of Null Moves from Finite Automata

If in an NDFA, there is ϵ-move between vertex X to vertex Y, we can remove it using the following steps −

- Find all the outgoing edges from Y.
- Copy all these edges starting from X without changing the edge labels.
- If X is an initial state, make Y also an initial state.
- If Y is a final state, make X also a final state.

Problem

Convert the following NFA-ε to NFA without Null move.



Solution

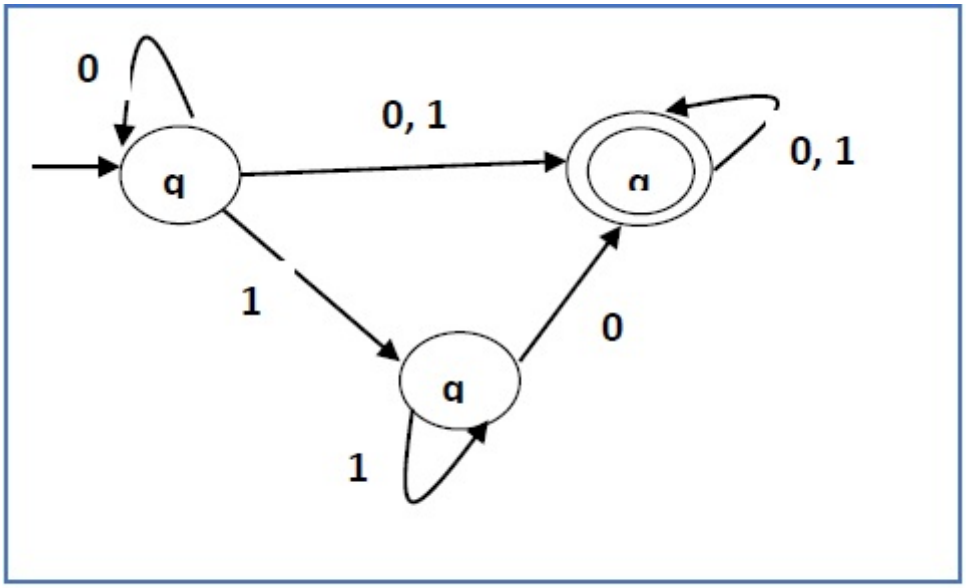
Step 1 −

Here the ε transition is between $\mathbf{q_1}$ and $\mathbf{q_2}$, so let $\mathbf{q_1}$ is **X** and $\mathbf{q_f}$ is **Y**.

Here the outgoing edges from q_f is to q_f for inputs 0 and 1.

Step 2 −

Now we will Copy all these edges from q_1 without changing the edges from q_f and get the following FA −

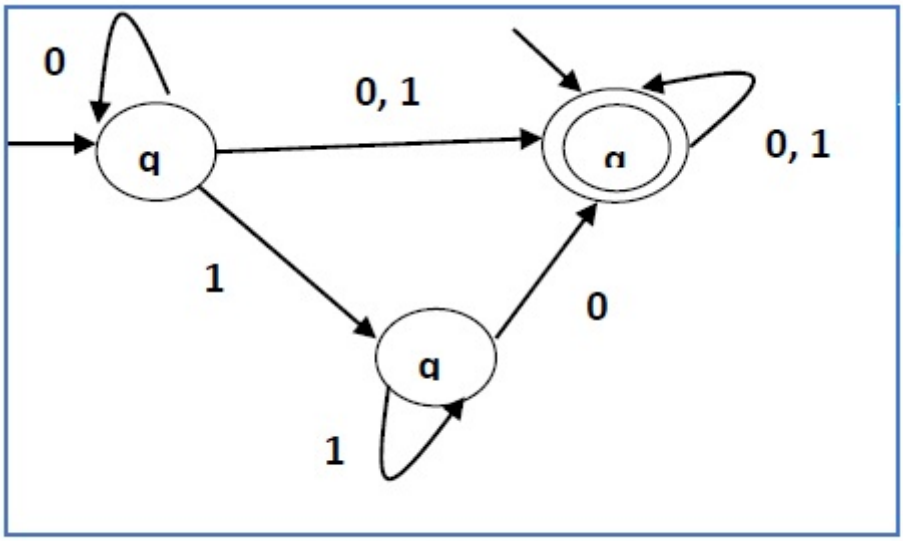


NDFA after step 2

Step 3 −

Here q_1 is an initial state, so we make q_f also an initial state.

So the FA becomes −

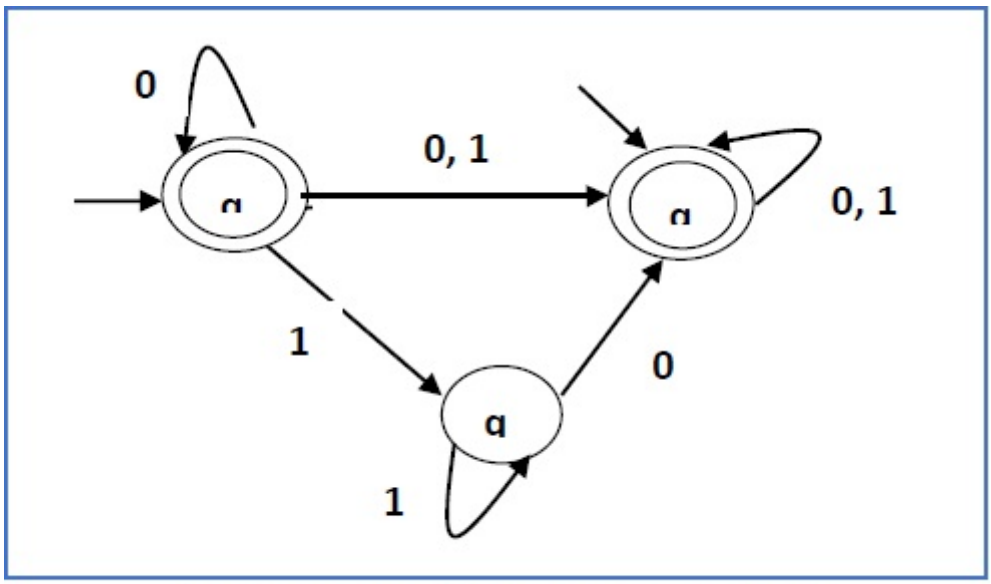


NDFA after Step 3

Step 4 −

Here q_f is a final state, so we make q_1 also a final state.

So the FA becomes −



Final NDFA without NULL moves