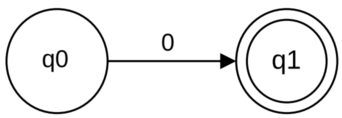
Assignment-06

Regular expression into NFA:
Languages in which we can create regular expressions, NFA and DFA, we call them regular languages. Regular language examples: start, end with , contain and so on..

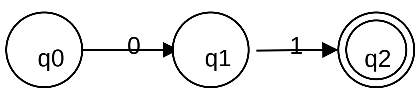
First we will learn to convert from regular expression to NFA to DFA then back to regular expression.

Examples:

- R1 = 0
  Thus, we can say that the language will be L1 ={0} so if we draw a NFA it will look like this,

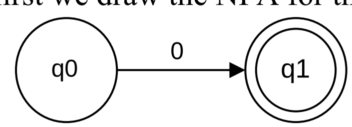


- R2 = 01
  So L2 ={01}
  We can draw a NFA from this and it will look like this,

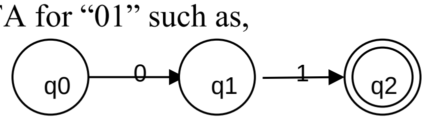


Remember that if the symbols/actions not drawn from any state will automatically represent that it went to die state and in NFA we do not show the die state.

- R3 = 0+01
  L3 = {0,01} so here first we draw the NFA for the string “0” which is,

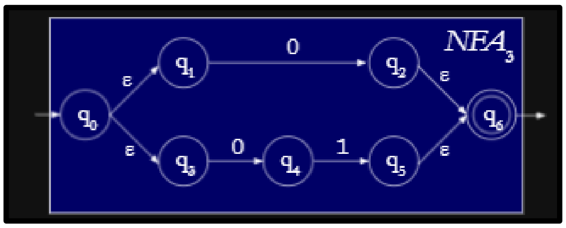


Then we draw the NFA for “01” such as,



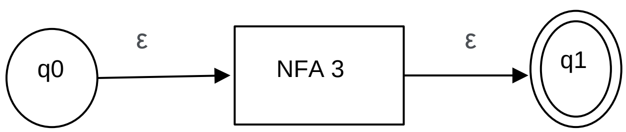
We have 0 + 01 so we might reach the final state either by 0 or 01. Now we merge both of the NFAs together by branching them using ε as shown in the diagram below.

This is one way of doing but we can follow another method which will have one starting state and one accepting(final) state(the previous diagram had two final states).

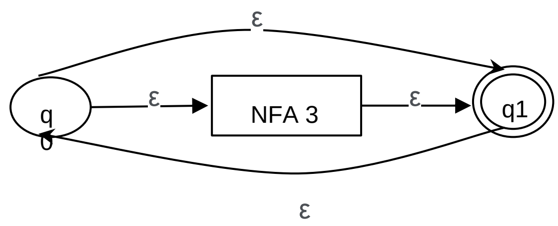


To summarise, first we made an NFA for 0 and then for 01. Later, we took both the NFAs of 0 and 01 and merged them together by branching them using ε for both start and end states.
Remember ε is the transition function which will let us allow to change state to another without any symbol

- R4 = (0+01)*
  Any possible combinations can form the language from 0 or 01 (except “11” as a substring will not work). L4 = {0,01,001,0001,010,…}. Due to “*” , ε will also be a part of the language.
  If we try to draw this we can notice that it is pretty similar with what we have drawn for R3 so what we will do is that we will make the whole R3 NFA diagram into a box called NFA3 which will represent the whole diagram. So if we make a new NFA for R4, we will get,

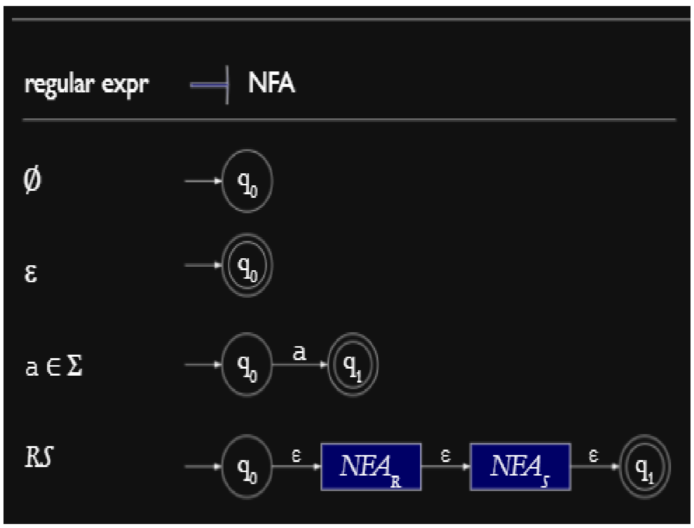


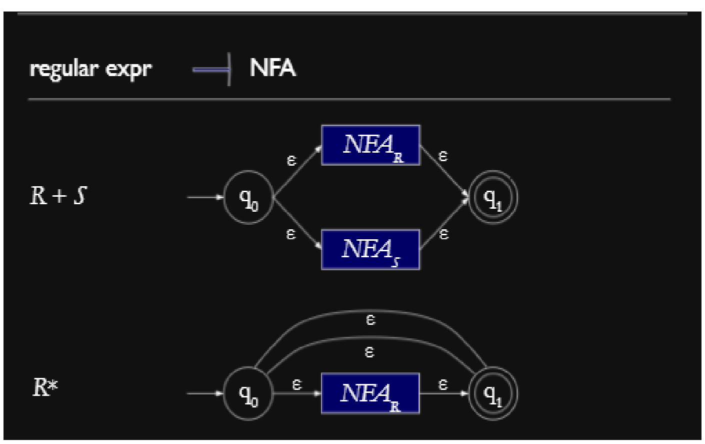
In Addition to this diagram, as we discussed, ε will be part of the language so action from q0 to q1 will occur and we go back from q1 to q0(to the initial state) if we do not stop yet.



Suppose we get 001 , first for 0 we will reach the final state but we have 001 left. So through ε action we will go back to the initial state(q0). Then finally we will go to the final state for 01.

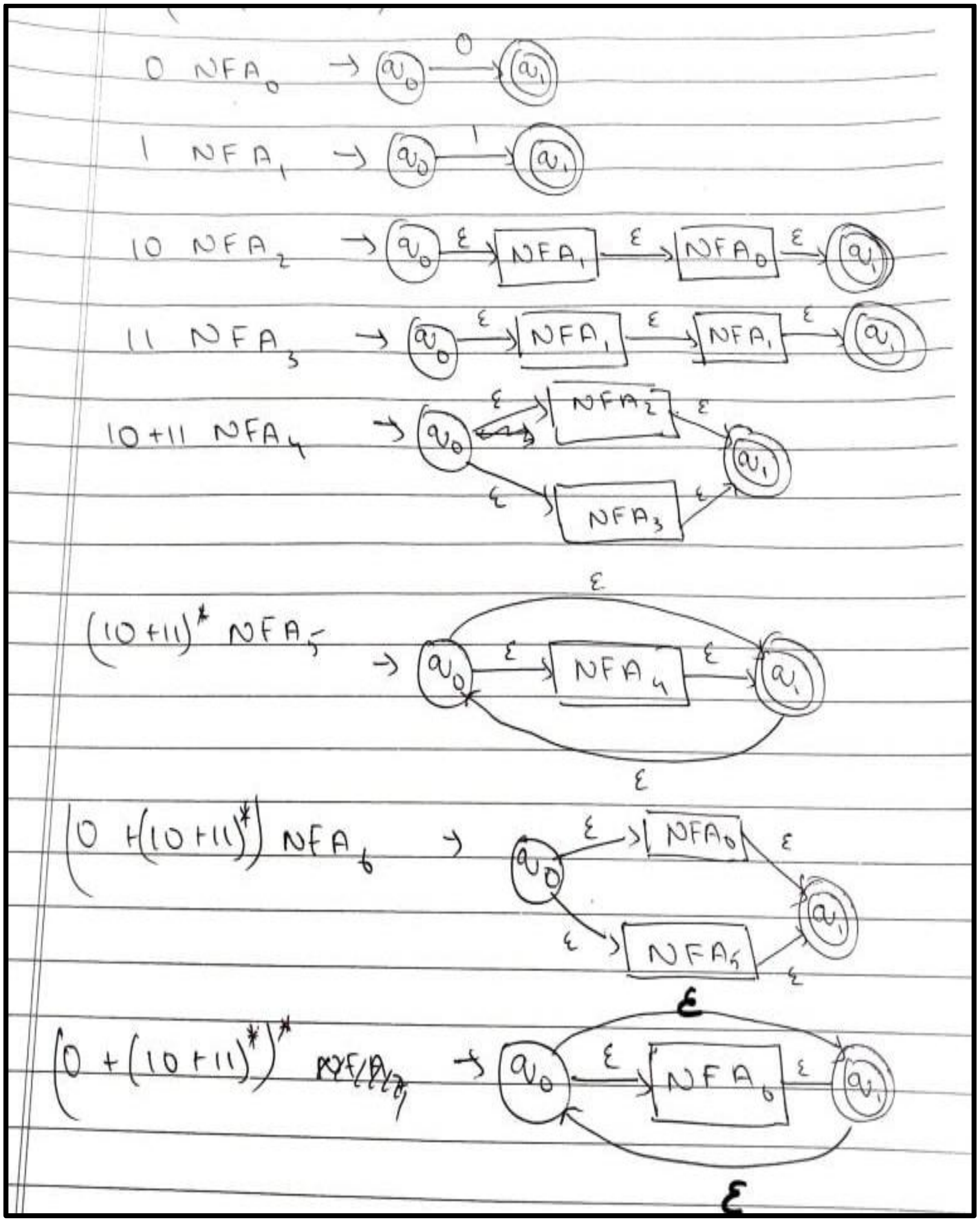
General Method:



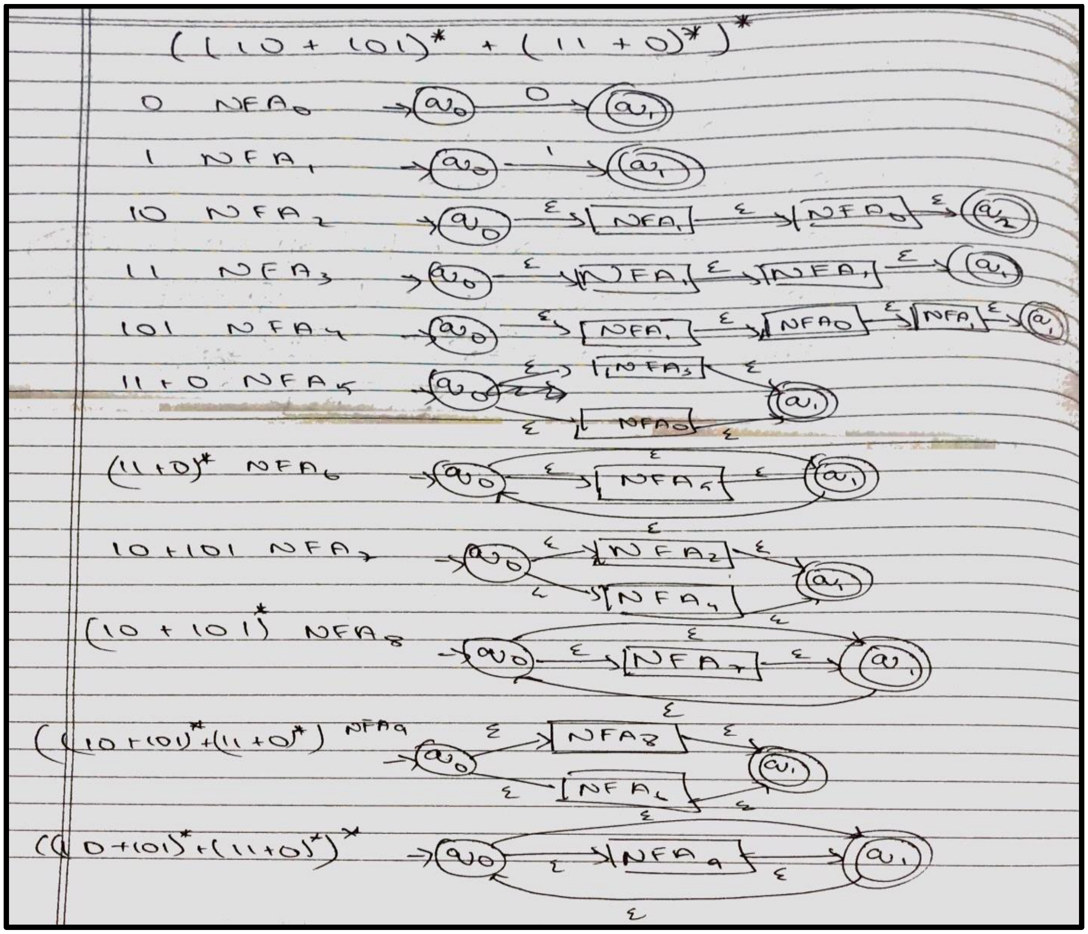


Practice:

- (0+(10+11)*)*



- ((10+101)*+(11+0)*)*


((10+101)* + (11+0)*)*
0 NFA0
1 NFA1
10 NFA2
11 NFA3
101 NFA4
11+0 NFA5
(11+0)* NFA6
10+101 NFA7
(10+101)* NFA8
((10+101)*+(11+0)*) NFA9
((10+101)*+(11+0)*)*