Assignment-07

Previously we looked into how to convert from RE to NFA.
Things to remember:

1. RE expresses the language or it generates the string.
2. DFA and NFA checks whether a string belongs to a language.
3. All DFAs are NFAs

Now we will deal with converting NFA to DFA and NFA to Regular Expressions.

NFA to DFA:

To convert from NFA to DFA we need to do the following steps-

1. Eliminate Ɛ transition (because DFA do not have any Ɛ transition)
2. Convert into simpler NFA (which means no non deterministic characteristic will exist here along with no Ɛ transition)

Process:

1. **Intuition-**
   For a given NFA,

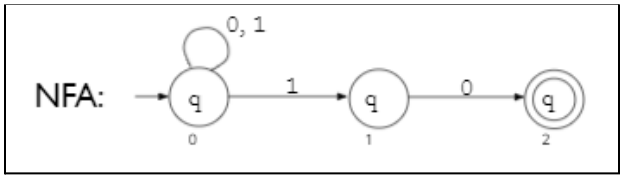


Figure-1

To convert this to DFA we first find the power set of the states for this NFA. which will be 2^3 (number of states = 3). The power set will have all possible combinations of states. In NFA, there are cases where a single state has two same actions going through different transitions. Suppose when q0 consumes 1, it goes to q1 and simultaneously stays at q0 thus we write the new state as {q0,q1}. States containing more than 1 combination are called compound states. After noting down all the combinations of states just as shown in figure 1 then we populate them which means we will find all actions from each state as shown in figure 2.

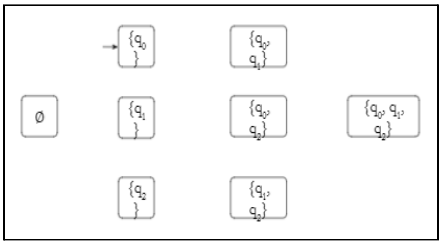


Figure-2

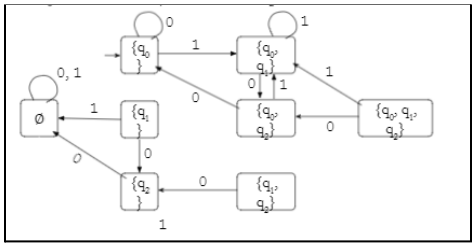
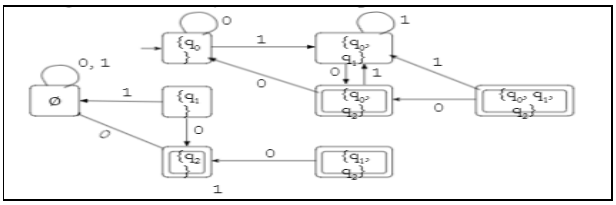
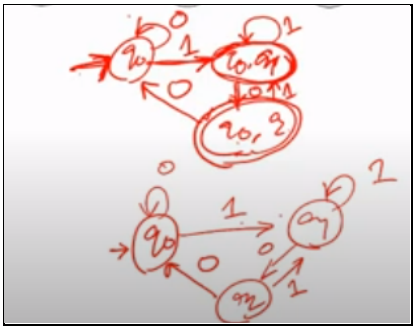


Figure-3



The states which will have q2 in it will be the accepting states as shown in figure 4. After that, we have the starting state as q0 just as in NFA and we will eliminate all the states from which there are no ways to reach back to q0. For example, from figure 3, if we look at q1, there is no possible connection/path/way to start from q1 and reach the starting state(q0) thus that state(q1) will be eliminated. Similarly, we will check for all the states and find that all other states will be eliminated except for q0, {q0,q1} and {q0,q2} just as shown in figure 5.



The above process was purely done in the intuition method and by following the conditions set by the NFA however, there might be cases where the number of states will be higher than 3 thus the power set will also contain a greater number of possible states which will be near to impossible to follow the above process. Thus, we will follow another way just by using intuition, we will obtain the DFA. Using the above NFA, we will find the next course of actions from each state instead of listing out all possible states. Suppose, q0 is our starting state because q0 is NFA was the starting state then using the NFA we see what happens when Figure 6 shows us the final DFA diagram.

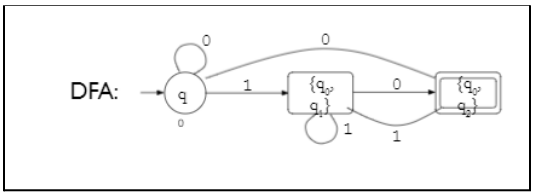


Figure-6

Eliminating Ɛ:

Here first we aim to remove the Ɛ from the NFA then create an NFA without Ɛ which will then be converted into DFA using the same process done above.

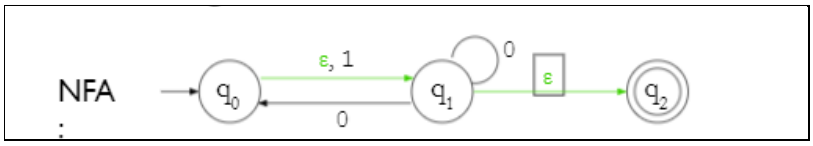


Figure-7

Here we list down for each state we find all the possible transitions to the final destination that could take place for each action. For example, from q0 if it consumes 0,

(q0,0)-> (q1,0)->(q0,Ɛ) this is one way. Another way, (q0, 0)-> (q1,Ɛ) and (q0,0)-> (q1,0)->(q2,0) ->(q2,Ɛ)

	0	1
q0	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$
q1	$\{q_0, q_1, q_2\}$	∅
q2	∅	∅

The accepting states will be q0,q1 and q2. Why will all three states be the accepting states? This is because, due to the empty transition, if we take q0 we can reach to q2 by Ɛ. Similarly for q1 by Ɛ we can reach q2 and q2 itself is the final state.

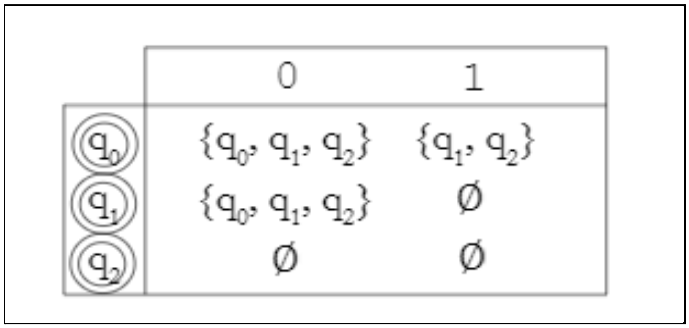


Figure-8

This is the new NFA without epsilon (simplified NFA)
Then using the table and our knowledge of converting NFA to DFA(from the above methods) we obtain the following DFA.

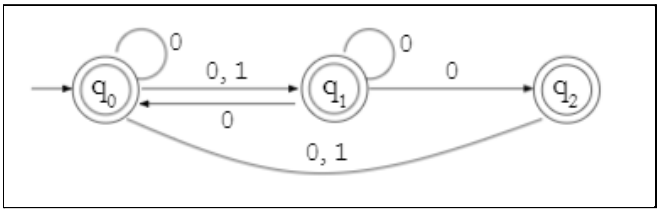


Figure-9

To summarise the whole process, first we eliminate the Ɛ if present then we will have the simplified NFA which we can convert it to DFA.

NFA to Regular Expression:

So far we have changed RE to NFA to DFA. Now we will study how to convert from NFA to Regular Expression.
For a given NFA we will add two new states. A new initial state and only one final state.
After doing that the states within the new two states, we will try to eliminate the states one by one and instead write the transitions in the regular expression format. Suppose,

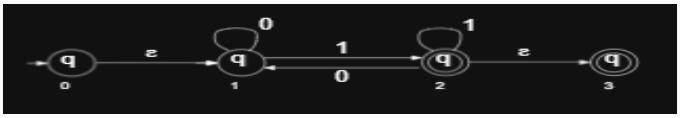


Figure-10

Here, we added two new states at the beginning and then at the end. We will create a direct connection between q0 and q3 by eliminating q1 and q2. For this process, all the loops can be represented by "*" which means any number of times that action can occur. If there are other ways we can change from one state to another but different transitions then we can use an "or/+" sign to join the both expression. We will continue to eliminate until we are left with q0 and q3. The regular expression formed will be the ultimate answer.

Later on we continued with some further examples on how to convert from NFA to RE.