ASSIGNMENT-4 [NFA]

NFA-

If we look into DFA, we see that each state has all possible symbols and the only transition was possible for each symbol. However, in NFA, the automata can exist in multiple states. Each state doesn't need to have all the possible actions. There can be two same actions going to different states or no actions at all from each state. For such cases(no action specified), we automatically consider it to go to die state. Except for these few changes, we can say that NFA is the same as DFA.

Example-1

here, there are in total three symbols leaving from the initial state q0. 0,1 and again 1. When 0 occurs, it will return to q0, but if 1 occurs, then two things will happen, we will stay at q0 and there will be a transition from q0 to q1. Both of the action will coincide and exist at the same time. Then if 0 is read, therefore, we will stay again at q0 and q1 will transit to q2. Similarly, if 1 is found, we will remain at q0 , the transition from q0 to q1 will occur and transition from q2 to q3 will happen. Thus, there are no actions given after q3 so if 1 or 0 is read, we consider that it will die automatically.

Meaning of NFA

Suppose we guess that we are 3 symbols away from reaching the endpoint or final state. Then we guess that we will see a pattern of 101 as we required. Then after reaching the desired position we check whether there are any more symbols after this state. If yes, then it keeps on accepting, or it will just die.

Example-2

For the input 01101:

- 0 is read so that we will stay at q0.
- Now 1 is read, so two transitions will occur here. First is it will stay in q0, and second, it will move from q0 to q1. Both occur simultaneously.
- We got another 1; however, we don't have any symbol 1 from state q1, so it will just die. But a transition will occur from q0 to q1 and another will stay at q0.
- When 0 is read next, it will stay at q0, q1 will move to q2.
- Now, after reading 1, q2 will move to q3, the final state. Q0 will move to q1 and another will stay at q0. Thus in the end, we got all the states from q0 till q3.

Example-3

Here we will design a NFA for over alphabet {0, 1} that accepts those strings that contain the pattern 001. Using the same process we have used earlier, if we find any string containing 001 in it, the machine will accept it. However, we will use here loops for the q0 and q3. In q0, it will wait until the string has reached its end. Once we have reached the final state, we stay here and let the rest of the transition occur.

Definition of NFA

NFA is of 5 tuples (just like DFA) 5-tuple (Q, Σ, δ, q_0, F), here the only difference is the δ transition function[δ: $Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow$ subsets of Q] in which it is the subset of Q because, in NFA, it can transit to many different states simultaneously. Another difference is that it allows ε, which means it can still be

able to transition from one state to another even if nothing is read. An example of this will be discussed in the next section.

Language of NFA

It has the same definition as DFA. It is the set of all strings(can be subsets, too) that the NFA accepts.

ε-Transitions

In **ε,** it can move from one state to another without reading any symbols. For example, given the following diagram, the NFA accepts a, b, aab, bab, aabab,..
If we look at the string "b":

(q_0, b)-> ({ q_1}, b) this is because **ε** lets us move to q1 for free (without reading any symbol)

->({q2},**ε**)-->accepted

Another string we look at is "bab"

(q0,bab)->({q1},bab)[Due to **ε** we were able to move to q2 freely]

({q_2}, ab) ({q_1}, b) ({q_2},**ε**) --> Reached final state so accepted.