Finite Automata

We have previously studied the graphical representation of how the gumball machines operated. From there we got familiar with the terms state and actions. We will now go further into this machine and learn about DFA. So, in the gumball machine, there are \$0,\$5,\$10, and go states. \$0 is considered the start state of the machine. The start state is identified by when an arrow from outside is pointing towards a state. Each state is shown as a circle.
The automata has a set of inputs/action {+5,+10,R}. These actions can either make the automata change from one state to another or the state remains unchanged.
Another state which is the "go" that is enclosed by double circles is known as the accepting state. In this state, the machine/automata have achieved its required goal.
The transition arrows must be coming from each and every state which shows us where the next state is.

DFA
A deterministic finite automata (DFA) is a 5-tuple (Q, Σ, δ, q_0, F), which means it has 5 things that define a graph that we have drawn. For each input symbol, we can determine the state to which the machine will move and it is said to be finite because it has finite number of states.

1. Q is the set of states
2. Σ, it is an alphabet(it is a set of input symbols where symbols are the actions such as for the gumball machine, alphabets will be +5, +10, and R)
3. δ, it determines by taking which steps/action will it reach to what state and result. $\delta : Q \times \Sigma \rightarrow Q$ is the transition function
4. q_0 , among all the states (from the set of states), it is the starting state.$q_0 \in Q$
5. F it is a set of final states. In other words, F is a subset of Q which contains all the final states.($F \subseteq Q$)

Example-1

For the following example, if we show the 5-tuple then :
$Q = \{q_0, q_1, q_2\}$
$F = \{q_0, q_1\}$
$q_0 = q_0$
$\Sigma = \{0,1\}$
To find δ , we need to find the cross product between Q and Σ as we know $\delta : Q \times \Sigma \rightarrow Q$ so using this formula,

$$(q_0, 0) \rightarrow q_0, (q_0, 1) \rightarrow q_1, (q_1, 0) \rightarrow q_2, (q_1, 1) \rightarrow q_1, (q_2, 0) \rightarrow q_2, (q_2, 1) \rightarrow q_2$$

This is can be represented by this or by a table method.
What is the transition function?
Starting from q to each and every state the symbol/action is taken is defined by the transition function.

Languages

The sequence of symbols/actions that are accepted by the machine as it reaches the accepting stage from the starting state. If we accumulate all the “accepted” sequences and form a set, this set is the language of that specific DFA.
Why do we call the language of a DFA a string? That is because, the criteria of a string that we all know is that it is a sequence of alphabets so similarly in the language of DFA, we are also finding a sequence of alphabets(also called actions/symbols).

Examples -2 :

1. Q = {q0, q1}, F ={q1}, q0 = q0
   Given, Σ = {a,b}
   δ :
   { (q_0, a) -> q_1

   (q_0, b) -> q_0

   (q_1, a) -> q_1

   (q_1, b) -> q_0 }

   To find the language, we need to find a sequence that satisfies the criteria.

   Therefore we need to check for Σ*.

   So, (q0,a)-> (q1, Σ) -- q1 ∈ F? Yes

   (q0,b)->(q0, Σ)--q0 ∈ F? No

   (q0,ab)-> (q1, b) -> (q0,Σ)-- q0 ∈ F? No

   (q0,aaa)->(q1,a)->(q1,a)->(q1->a)-- q1 ∈ F? Yes

   By continuing this for all possible sequences, we find a pattern which is the language of this DFA is the set of all strings that end with ‘a’.

2. Q = {q0, q1,q2,q3,q4}, F ={q1,q3}, q0 = q0
   Given, Σ = {a,b}
   Then we go to find the possible outcomes of each state.

The language of this DFA is the set of strings that have the same starting and ending alphabets/symbols.

3. Here is a part of the language because the initial state is an accepting state.

   So, F ={q0,q1} Q = {q0,q1,q2} q0 =q0

   0,1->yes , 00,11-> yes, 10->no , 01->yes, 000,111->yes, 101,110,010,100->no, 011 ->yes after observing these sequences of strings we can assume about the language.

   Language = any strings that does not contain 10 as its a substring.

Examples-3

Here we need to design a DFA and the condition given is that at most three 1s are needed to reach the final state. Hence, we can say '∈' is part of the language as no 0s also fits into the given criteria so the initial state is an accepting state. Therefore q1,q2,q3 the next three states are also accepting states if the actions are leaving from q0 till q3. If we add another action 1 from q3 it will reach q4 state but q4 is NOT an accepting state as we bypassed the criteria and there are four 1s from q0 to q4. There are no conditions for 0s so we can say that 0 will bring no change to the states.

Example-4

This DFA has a language = { 010, 1} over $\Sigma = \{0, 1\}$.

To design this DFA, according to the language, 010 and 1 will be the accepting state. Here ∈ is not a part of the language so we can take this as our initial state. Therefore, we will look for the valid inputs for the and when after reaching the ∈ if that string is part of the language then that will be our accepting state. In DFA, all possible actions must be shown from each state. Any action that comes after "010" will be sent to the q_{die}. To conclude, any sequence that does not follow "010" or "1" will be sent to q_{die}.

Example -5

Here we have seen an approach of drawing the DFA however we found another approach which were more smaller and better.

Example-6

For this example, we looked for when the prefix must be "101" , the suffix must be "101" and lastly "101" must be a substring. We looked how each of them can be designed in such ways that will bring us the required results.