COL352: Assignment 1

Sachin 2019CS10722

January 2022

1 Question 1

2 Question 2

An all-NFA M is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ that accepts $x \in \Sigma^*$ if every possible state that M could be in after reading input x is a state from F . Note, in contrast, that an ordinary NFA accepts a string if some state among these possible states is an accept state. Prove that all-NFAs recognize the class of regular languages.

To prove that all-NFA's recognise the class of regular languages we need to show two things, firstly that the language accepted by all-NFA's is regular, and secondly given any regular language there exists an all-NFA which accepts it. Following are the proofs of these parts,

To Prove: Language accepted by all-NFA is regular.
Proof: Now by the definition, all-NFA M is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ that accepts $x \in \Sigma^*$ if every possible state that M could be in after reading input x is a state from F . This would mean the all-NFA's are NFA because NFA accepts the string even if some of the states reached after reading an input x is in accept state F. NOw we know that the language accepted by NFA is regular. Therefore the language accepted by all-NFA is also regular. Hence proved.

To Prove: For every regular language there exists an all-NFA that accepts it.
Proof: We know that for every regular language there exists a DFA which accepts it. Now the definition of a DFA M is that it is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ that accepts $x \in \Sigma^*$ if the state that M could be in after reading input x is a state from F. Now we also know that the set of states DFA M would be in after reading the input x is a singleton set (Deterministic nature) and the state belongs to F if x is accepted by DFA. So every DFA is an all-NFA. Therefore for every regular language, there exists an all-NFA that accepts it. Hence proved.

Now above two facts would imply that the all-NFA's recognize the class of regular languages.

3 Question 3

4 Question 4

Step1(complementation):Generate another DFA D_3 which is complement of D_2 i.e by reversing final states as non final states and vice-versa.Then D_3= (Q_2,Σ,r_0,δ_2,
Q_2-F_2).
Step2(intersection):Generate another DFA D_4 by intersecting D_1 and D_3. Now this can be done by making a new DFA with $|Q_1| * |Q_2|$ number of states. Now the states in this dfa are touple of the states from the two original DFA's, first element from D1 and second from d_3. The start state is the touple (s_1, s_2) where s_1 and s_2 are start states of D_1 and D_3 A transtion exist between a state (p, q) to (x, y) on alphabet a if in original DFAs p to x and q to y transitions existed on aphabet a. All the touples which have either of the state as an accept state in the orginal DFA is an accept state in the new DFA.
Step3(BFS):Perform BFS on start state of D_4 if it find any final state then return false else
Step4:Repeat steps 1,2,3 for computing L(D_1)-L(D_2) if bfs in this also finds nothing then return true.
Proof of correctness:
Claim1: Let A,B be two DFAs then if they recognise same language then the given algorithm would return true
Proof by contradiction: Now we know that 2 language is same when both L(A)-L(B) and L(B)-L(A) are null. And when both the language is same then the DFA'a A and B recognise the same language. Lets say L(A)-L(B) is not null then there there exist a string in L(A) which will be in L(A) and not L(B). Now the DFA D_4 which we have created would accept it and hence by our algortihm false is returned. Now similary when L(B)-L(A) is not null false is retuned when doing BFS. So our alogorithm only return true when both L(A)-L(B) and L(B)-L(A) are null. Thus our algorithm is indeed correct.
Time complexity: Worst case Time complexity of this algorithm will be
O($|Q_1| * |Q_2|$+no of transistions in D_4)

5 Question 5

For any string $w = w_1 w_2 ... w_n$ the reverse of w written w^R is the string $w_n ... w_2 w_1$. For any language A, let $A^R = w^R | w \in A$. Show that if A is regular, then so is A^R. In other words, regular languages are closed under the reverse operation.

As every regular language has a DFA which accepts it, let D be $(Q, \Sigma, \delta, q_0, F)$ that accepts the language A. Now we will construct an NFA N from this DFA. The steps of the construction are given below. We will then show that the language accepted by NFA is indeed A^R.

Construction: To construct this NFA N we will have to reverse all the edges of the DFA D. Also make the start state of the D as the accepting state of the N. Add ϵ transitions from the accepting states of D to a new state s in NFA. Make s the start state of the NFA.
So N is $(Q_1, \Sigma, \delta^{'}, q_0^{'}, F^{'})$ such that

$$
\begin{aligned}
Q_1 &= Q \cup \{s\} \\
F^{'} &= q_0 \\
\delta^{'}(q_1, a) &= \begin{cases} q_2 & \text{if } q_1 \in Q \setminus F \text{ and } \delta(q_2, a) = q_1 \\ s & \text{if } q_1 \in F \text{ and } a = \epsilon \end{cases} \qquad (1) \\
q_0^{'} &= s
\end{aligned}
$$

To Prove: The language accepted by N is A^R.
Proof: Now take any string $w_1 w_2 .... w_n$ from the language A. The path(path is state then alphabet taken then next state reached) taken by this string to accept state in D would be $q_0, w_1, q_1, w_2, q_2 .....q_n, w_n, f$ where $q_i \in Q$ and $f \in F$. Now take the reverse of the string $w_n w_{n-1} ....w_1$. There exist a path from start to accept state in N as which is as follows $s, \epsilon, f, w_n, q_n, w_{n-1} .....q_1, w_1, q_0$. We know that q_0 is an accept state of N.Thus we have got an NFA that has an accepting path for any string w in A^R.
Now we have proved that for every string in A we have a accepting path in N. Now we can also similarly prove the reverse direction too i.e. for every string w in A^R we have accepting path in D for reverse of w. The proof of this goes as follows:
Take any string $w_1 w_2 ....w_n$ from the language A^R. The pathtaken by this string to accept state in N would be $s, \epsilon, f, w_1, q_n, w_2 .....q_1, w_n, q_)$ where $q_i \in Q$ and $f \in F$. Now take the reverse of the string $w_n w_{n-1} ....w_1$. There exist a path from start to accept state in D as which is as follows $q_0, w_n, q_1, w_{n-1}, q_2 .....q_n, w_1, f$. We know that f is an accept state of D.Thus we have got an DFA that has an accepting path for any string w in A. Hence we have proved that regular languages are closed under the reverse operation.

6 Question 6