Page No.
Class
Roll No.
Date: / /202

Practical No. 07

Title :- Implement Deterministic Finite Automata.

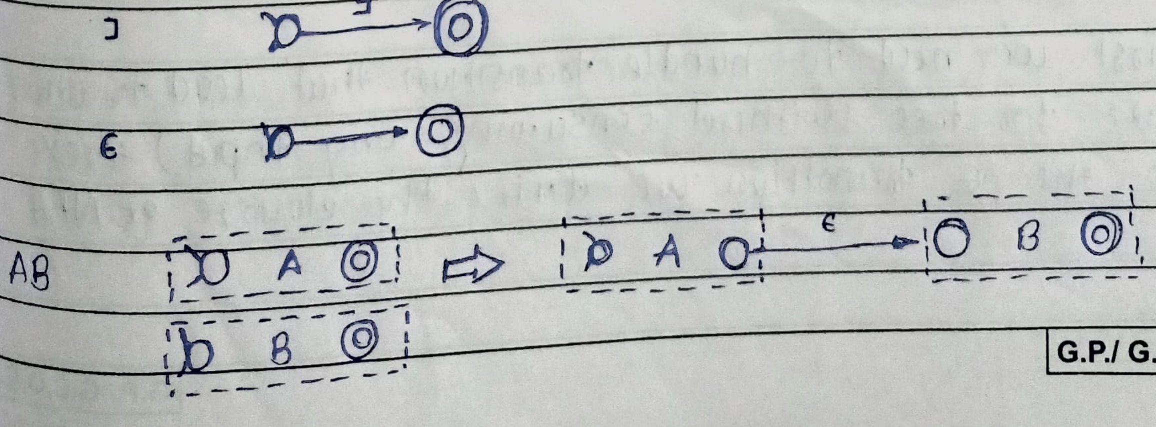
Theory :-

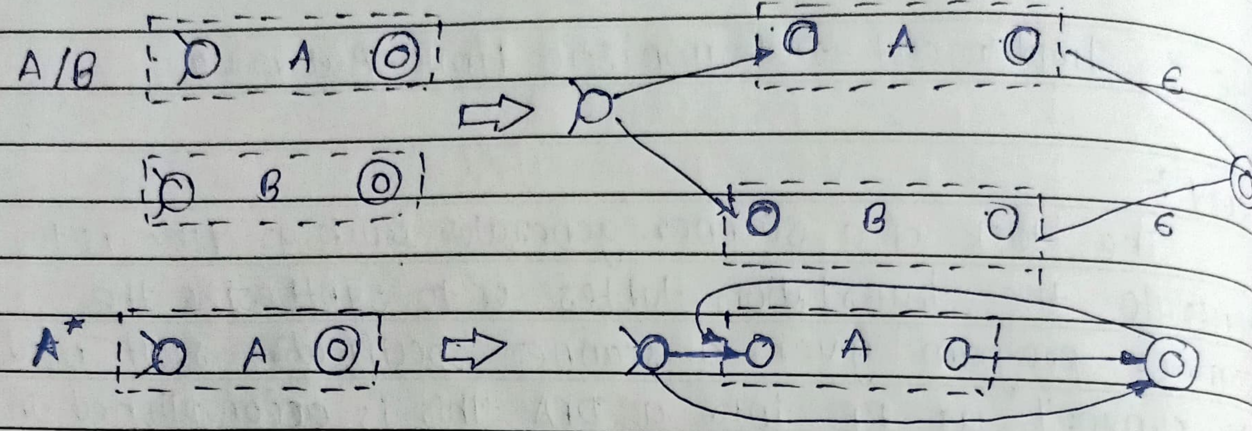
The task of a scanner generator such as flex is to generate the translation tables or to synthesize the scanner program given a scanner specification so it need to convert ~~ar~~ RE into a DFA this is accomplished in two steps first it convert a RE into a non-deterministic finite automation (NFA) and then it convert the NFA into DFA.

A NFA is similar to a DFA but it also permits multiple transition over the same characters and transition over ε.

Clearly DFA's are a subset of NFA's. But it turns out that DFA's ~~and~~ NFAs have the same expressive power the problem is not when converting NFA to DFA we may get an exponential blowup in the number of states

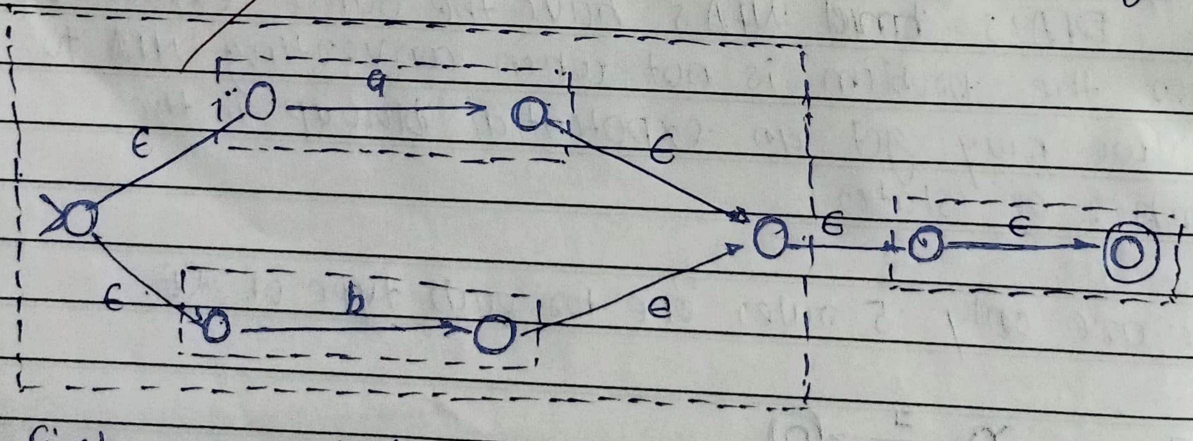
There are only 5 rules one for each type of RE.





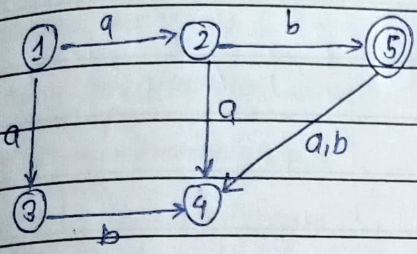
The algorithm construct NFAs with one final state fo
example the third rule indicates that to construct the
NFA for the RE AB, we construct the NFA s for A an
which are represented as two boxes with one st
and one final state for each box.

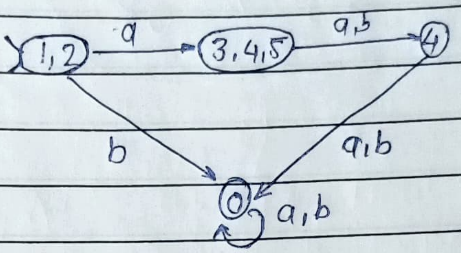
for ex. The RE (a/b)c is mapped to the following NF



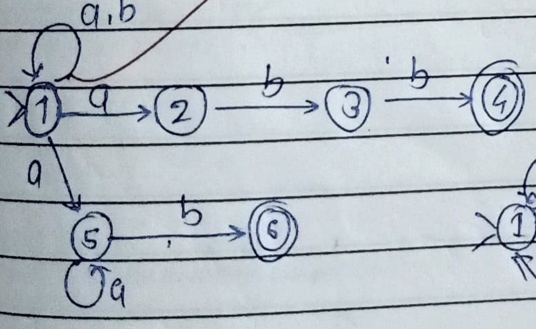
first we need to handle transition that lead to othe
state for free (without consuming any input) these
are the ε transition we define the closure of NFA

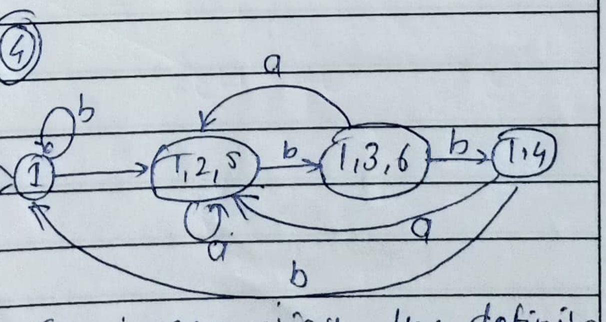
node as the set of all the nodes reachable by this nodi using one or ϵ tranoition for example the clousre of node I in the left figure below.





State for every DFA state labeled by some set {S1, --- Sn}. and for every characters C in the language alphabet you find all the state reachable by S1, S2 ..... or Sn using e arrows and you find all the states reachable by S1, S2 ---- or Sn using C arrows and you union. together the closer of the modes.





Result :- thus the program for implementing the definite finite Automate was executed and the output was verified successfully.