Date: 07/08/2024

Experiment 1.3

AIM

To convert NFA with ε transition to NFA without ε transition.

ALGORITHM

1. Start
2. Create utility functions for NFA data structure to read and write transitions.
3. Create a DFS function that finds epsilon closure of a state and stores in a boolean array as follows:
    1. If state is visited, terminate function call.
    2. Mark state as visited.
    3. For each transition from the given state via input alphabet ε:
        1. Recursively call the DFS function for the target state.
4. Read NFA input as follows:
    1. The first line contains the number of states (n) , number of final states (f) , number of input alphabets(m), and number of transitions(t).
    2. The next line contains f space separated integers denoting the final states.
    3. The next line contains the m input alphabets as a single string.
    4. The next t lines contain transitions as “qi qj c” representing a transition from qi to qj on input alphabet c. Here, the alphabet ‘e’ denotes epsilon.
5. For each state in the ε- NFA, find the ε- closure using DFS.
6. Create new NFA with same number of states as input NFA.
7. For each state s in the original NFA:
    1. For each state s’ in the ε- closure of s:
        1. For each transition from s to a state t via input symbol c such that c!= ε :
            1. For each state t’ in the ε- closure of t:
                1. Add transition from s to t’ in the new NFA.
8. For each state s in the original NFA:
    1. For each state s’ in the ε- closure of s:

    1. If s’ is a final state in original NFA, mark s as a final state in the output NFA.

9. Print the new NFA.
10. Stop

RESULT

Successfully converted NFA with ε transition to NFA without ε transition.