Theory Computing - CSC/CSE 273 (Section 1)

Programming assignment

(Total 100 Points)

Due on 13th August by 11:59 PM

Write a program to converts an ε-NFA into a DFA. There must be two parts in your implementation.

1. Convert an ε-NFA into a DFA.
2. Take a string as input and it should print the list of states the DFA goes through, and then print either "Accepted" or "Rejected".

1. Convert an ε-NFA into a DFA

The Algorithm for this is as follows:-
Let the DFA be D & its transition table be DTran. Dstates represents the states of the DFA. Let NFA be N.

Initially e-closure(s){start state of NFA} is the only state in Dstate and it is unmarked.
While there is an unmarked state T in Dstate do begin
    Mark T;
    For each input symbol a do begin
        U=e-closure(move(T,a));
        If U is not in Dstates then
            Add U as unmarked state to Dstate.
        Dtran[T,a]=U;
    End
End

A state D is an accepting state if it is a set of NFA states containing at least one accepting state of the N.

Transition table for ε - NFA

STATES	a	b	e (Epsilon)
->0	-	-	{1,7}
1	-	-	{4,2}
2	{3}	-	-
3	-	-	{6}
4	-	{5}	-
5	-	-	{6}
6	-	-	{1,7}
7	{8}	-	-
8	-	{9}	-
9	-	{10}	-
*10	-	-	-

Input format:

Number of states: 11
Start state: 0
Number of final state(s): 1
1. 10
Number of input symbols: 2
1: a
2: b
Number of transitions for epsilon from state 0: 2
1. 1
2. 7
Number of transitions for a from state 0: 0
Number of transitions for b from state 0: 0
Number of transitions for epsilon from state 1: 2
1. 2
2. 4
Number of transitions for a from state 1: 0
Number of transitions for b from state 1: 0
Number of transitions for epsilon from state 2: 0
Number of transitions for a from state 2: 1
1. 3
Number of transitions for b from state 2: 0
Number of transitions for epsilon from state 3: 1
1. 6
Number of transitions for a from state 3: 0
Number of transitions for b from state 3: 0
Number of transitions for epsilon from state 4: 0
Number of transitions for a from state 4: 0
Number of transitions for b from state 4: 1
1. 5
Number of transitions for epsilon from state 5: 1
1. 6
Number of transitions for a from state 5: 0
Number of transitions for b from state 5: 0
Number of transitions for epsilon from state 6: 2
1. 1
2. 7
Number of transitions for a from state 6: 0
Number of transitions for b from state 6: 0
Number of transitions for epsilon from state 7: 0
Number of transitions for a from state 7: 1
1. 8
Number of transitions for b from state 7: 0
Number of transitions for epsilon from state 8: 0
Number of transitions for a from state 8: 0
Number of transitions for b from state 8: 1
1. 9

Number of transitions for epsilon from state 9: 0
Number of transitions for a from state 9: 0
Number of transitions for b from state 9: 1
1. 10
Number of transitions for epsilon from state 10: 0
Number of transitions for a from state 10: 0
Number of transitions for b from state 10: 0

The e-closures are calculated as follows:

eclosure(0)={0,1,2,4,7}=**A=start state** in corresponding DFA
eclosure(move(A,a))={1,2,3,4,6,7,8}=**B**
eclosure(move(A,b))={1,2,4,5,6,7}=**C**
eclosure(move(B,a))=B
eclosure(move(B,b))={1,2,4,5,6,7,9}=**D**
eclosure(move(C,a))=B
eclosure(move(C,b))=C
eclosure(move(D,a))=B
eclosure(move(D,b))={1,2,4,5,6,7,10}=**E=final state**
eclosure(move(E,a))=B
eclosure(move(E,b))=C

Output:

The Corresponding **DFA** is:-

Transition table for DFA

STATE	**a**	**b**
->A	B	C
B	B	D
C	B	C
D	B	E
*E	B	C

2. Input and output

Input String: bababb
A -> C -> B -> D -> B -> D -> E
Accepted.
Input String: aabba
B -> B -> D -> E -> B
Rejected.

Extra credit: 30 Points

For this part you have to minimize states in DFA.

Output for the above DFA

Transition table for minimized DFA

STATE	a	b
->{A,C}	B	{A,C}
B	B	D
D	B	E
*E	B	{A,C}

This part will only be counted if your ε - NFA to DFA conversion is correct.