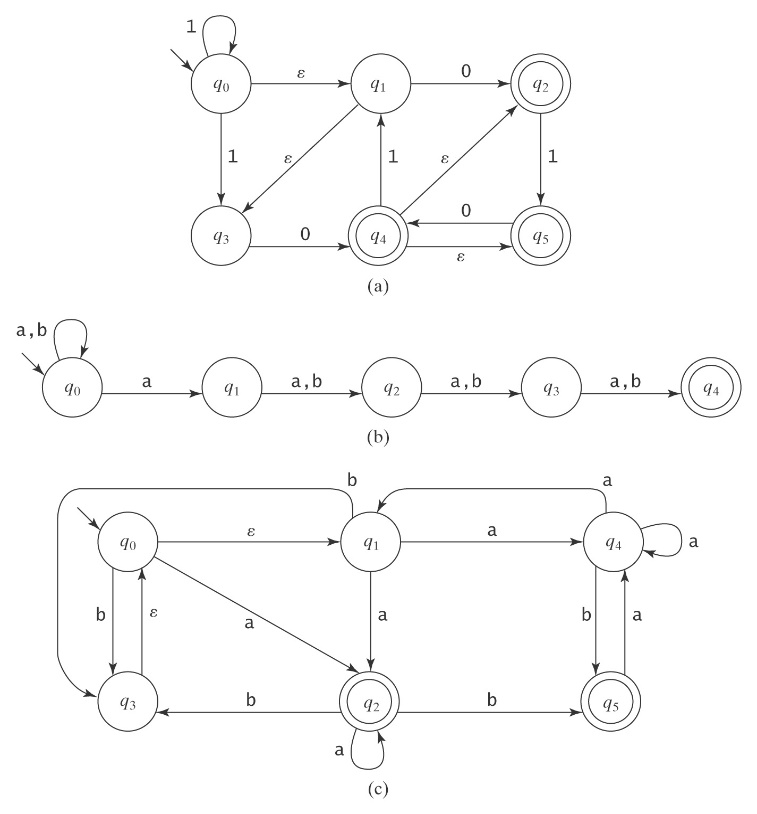
**Homework problems**

1. Convert the following NDFSM to a DFSM using algorithm ndfsmToDfsm.

****

1. eps(q0) = {q0, q1, q3}.

eps(q1) = {q1, q3}.

eps(q2) = {q2}.

eps(q3) = {q3}.

eps(q4) = {q2, q4, q5}.

eps(q5) = {q5}.

2. s'={ q0, q1, q3}

3.

|  |  |  |
| --- | --- | --- |
| {q0, q1, q3} | 0 | {q2, q4, q5} |
|  | 1 | {q0, q1, q3} |
| {q2, q4, q5} | 0 | {q2, q4, q5} |
|  | 1 | {q1, q3, q5} |
| {q1, q3, q5} | 0 | {q2, q4, q5} |
|  | 1 | { } |

4. K'={{q0, q1, q3}, {q2, q4, q5}, {q1, q3, q5}, φ}

5. A'={{q2, q4, q5}, {q1, q3, q5}}

1 0 0

0,1,3 0 2,4,5 1 1,3,5

//DFSM.java

import java.util.Vector;

import java.util.Iterator;

import java.util.Scanner;

class DFSM

{

private static Scanner Keyboard=new Scanner(System.in);

private static String startState=null;

private static Vector<String> finalStates=new Vector<String>();

private static Vector<String> transitions= new Vector<String>();

public static void main(String[] args)

{

readDFSM();

while (true)

{

System.out.print("Enter input string, enter done to end:");

String w=Keyboard.nextLine();

if (w.equals("done")) break;

String curState=startState;

String config;

for (int j=0; j<w.length(); ++j)

{

config=curState+","+w.charAt(j);

curState=moveTo(config);

if (curState.equals("Reject")) break;

}

if (curState.equals("Reject")) System.out.println("Reject");

else if (finalStates.contains(curState)) System.out.println("Accept");

else System.out.println("Reject");

}

}

//Pre: config is "state,char"

//Return: next state if there is a move from config, "Reject" if there is no move

public static String moveTo(String config)

{

String answer;

for (int k=0; k<transitions.size(); ++k)

{

answer = transitions.get(k);

if (answer.startsWith(config)) return answer.substring(4);

}

return "Reject";

}

public static void readDFSM()

{

System.out.print("Enter start state:");

startState=Keyboard.nextLine();

System.out.println("Enter final states, 1 on each line. Enter 0 to end:");

String s=Keyboard.nextLine();

while (s.equals("0")==false)

{

finalStates.add(s);

s=Keyboard.nextLine();

}

System.out.print("Enter transitions, 1 on each line with no ws: state,letter,state. ");

System.out.println("Enter 0,0,0 to end:");

s=Keyboard.nextLine();

while (s.equals("0,0,0")==false)

{

transitions.add(s);

s=Keyboard.nextLine();

}

}

}

//Test data: a DFSM that accepts L={w∈{a, b}\*:w contains an even number of a’s and an odd number of b’s}.

a a

-1

2

a

b b b b

a

4

3+

a

Start state: 1

Final states: 3

Transitions:

1,a,2

1,b,3

2,a,1

2,b,4

3,b,1

3,a,4

4,b,2

4,a,3

Accept: ababb

Reject: abab

//Test data: a DFSM that accepts L={all strings on Σ={0,1} except those containing the substring 001}.

0 0,1

1 0 1

4

+3

+2

±1

1

0

Start state: 1

Final states: 1, 2, 3

Transitions:

1,1,1

1,0,2

2,0,3

2,1,1

3,0,3

3,1,4

4,0,4

4,1,4

Accept: 0101

Reject: 010010