CS410 Project-1 Description

alp demirezen

10/12/2021

1 General Description

Start Date: 10/12/2021

In this assignment you are expected to simulate a Deterministic Finite Automata (DFA). Note that your simulation should work for any given alphabet, number of states, number of variables and transitions.

Your program should be working exactly as a DFA would do. Given a string the simulated DFA should be able to tell if the string is accepted or rejected. Your program should output the information about the given string being accepted or rejected as well as which states it visited.

The programming languages allowed for this assignment are Java and C++.

For this assignment we first expect you to prepare a design for DFA simulation within a week. Later you will be implementing your simulation according to your design. The given time will be two weeks for the implementation part.

You will have a Demo for the assignment. In the Demo, you will need to showcase that your program works as it supposed to. Also you should prove that you implemented your work according to your design.

2 Example input and output files

Note that you are free to construct your own input file, it is fine as long as your program works as it supposed to. Our goal here is to give you a brief understanding of how to construct an input file.

An input file should contain the number of states, number of variables and the goal state(s). Once those two inputs are given you should express the state diagram that you want to simulate. Finally, the string to detect could be given as an input. An example file could be constructed as follows.

2.1 Input file

2 (number of states)
2 (number of variables)
1 (number of goal states)
q1 q2 (states)
q2 (goal state(s))
a b (variables)
q1 a q1 (q1 state'inden a ile q1 stateini gidiyor)
q1 b q2 (q1 state'inden b ile q2 stateini gidiyor)
q2 a q2 (q2 state'inden a ile q2 stateini gidiyor)
q2 b q1 (q2 state'inden b ile q1 stateini gidiyor)
aba (string to be detected)
abababa (string to be detected)

An output file should contain information about whether the string is accepted or rejected, as well as the path string followed. An example output file could be constructed as follows:

2.2 Output file

q1 q2 q2 (route taken)
Accepted
q1 q2 q2 q1 q1 q2 q2 q1 q1 (route taken)
Rejected

3 Submission and Grading criteria

You will be submitting both your implementation and design through LMS.
Design Grade: 30
Implementation Grade: 70