CS250 Lab 2 – Nondeterministic Finite Automata

Objectives: In this lab you will learn how to

- use nondeterminism for regular language recognition
- analyze the equivalence of NFAs and DFAs

Download the lab2 files and start JFLAP. Open the file ex2.1.
If you go to `Test-Highlight Nondeterminism` you will see that in state $q0$ we have a choice of either going to state $q1$ or state $q3$ on input `a`.

Assignment 1:
Use the step by step run to test the input `aa` on the NFA ex2.1. After one step we can be in either $q1$ or $q3$. After one more step we can be in either $q2$ or $q4$. Since $q2$ is an accepting state, the input is accepted. Notice that it does not matter that the other choice is not accepting. An NFA accepts if there **exisits** a choice which leads to an accepting state.

Try step by step with input `aaa`. Does it accept?
Try step by step with input `aaaa`. Does it accept?
Clearly and succinctly describe the language that is accepted by the NFA in ex2.1

Assignment 2:
The file ex2.2 contains an NFA with a λ transistion. That means that there is a choice to move to a state without consuming any input.

Give 3 inputs in the alphabet a,b which are accepted.
Give 3 inputs in the alphabet a,b which are rejected.

Assignment 3:
The file ex2.3 contains an NFA that has a single transition that can consume either two `a`'s or three `a`'s. We want a FA which recognizes the language L of any string of `a`'s of length divisible by either 2 or 3. Does this FA achieve this?

Produce at least 4 strings which either are in L but are not accepted, or are not in L but are accepted.
Modify the FA so that it actually recognizes the language L. Simulate those same strings again to verify your modification.

Any language that can be accepted by an NFA can also be accepted by a DFA. The algorithm for converting an NFA to an equivalent DFA is given on p61 of your text but JFLAP can assist you with the process.

Go back to ex2.2. In the editor window, select `Convert to DFA`. This will bring up a new split screen with the NFA on the left. If it is too tiny, simply drag the vertical line between the screens to make it larger. On the right we will build the DFA. You are given an initial state. The subscripts of 0,1 on the state indicate that initially we can be in either state 0 or state 1 of our NFA without consuming any input. Do you see why this is so?

To add the next state click on the `Expand on Terminal` tool and click and hold on q0 and drag to a location for the next state. You will be asked to enter a terminal and type `a`. You will then be asked to enter what states can be reached from either 0 or 1 of the NFA on an input of `a`. —– The answer is q2 and q3. Do you see why? Type 2,3 and JFLAP will add that state.

Continue the process of adding new states. If you press the `Done?` button, JFLAP will inform you of what you have left to do, if anything. Test out the DFA to convince yourself that it accepts the same language.

Assignment 4:
An NFA recognizes language L if for each string w in L there exists of a path labeled w to an accepting state. We want to define the complement of L, which is all the strings from the alphabet that are not contained in L. Exercises 7 and 8 on p65 of your text attempt to do this, although only one of them is correct. You may have some trouble reading the mathematical notation but exercise 7 says that none of the paths labeled w lead to accepting states. Exercise 8 says that there exists a path labeled w which leads to a non-accepting state. Read the notation carefully so that you understand what it is saying.

For each of exercise 7 and 8 on p65, either explain why it correctly describes the complement of L or give a counterexample (some NFA M whose complement does not meet the definition). Be sure to carefully and clearly explain your answers in each case.

Submit your files in goucherLearn for grading.