Homework 2: DFAs and NFAs

CSCI 161

Due 9pm, Thursday, October 1, 2020

Krehbiel

Overview. This assignment has you construct several finite automata for regular languages. The first three must be DFAs, meaning that your diagrams should have an arrow for every state-character pair. For LaTeX extra credit, turn in this tex file with screenshots of your jff files for the five JFLAP questions as well as your collaboration statement and your writeup for Question 6.

Question 0. Who, if anyone, did you consult for help on this homework? Elaborate to the extent helpful.

Question 1. Construct a state diagram for the DFA $M = (\{q_1, q_2, q_3, q_4, q_5\}, \{u, d\}, \delta, q_3, \{q_3\})$ where δ is described by the following transition table:

	u	d
q_1	q_1	q_2
q_2	q_1	q_3
q_3	q_2	q_4
q_4	q_3	q_5
q_5	q_4	q_5

What could this machine represent?

Question 2. Given an alphabet of $\Sigma = \{0, 1\}$, design a DFA that recognizes the language $L = \{\epsilon, 0\}$.

Question 3. On Friday of Week 1 we proved that the regular languages are closed under union. We can similarly prove that regular languages are also closed under intersection by using the same DFA construction from the proof of closure under union but requiring *both* attributes in a pair to be an accept state in their corresponding machine in order for the pair to be accepted by the larger machine. With that approach, design a DFA that recognizes the language of binary strings that have an even number of ones and end with one.

Question 4. Design a DFA or NFA that recognizes binary strings with at least two 0s or at least three 1s.

Question 5. Design a DFA of NFA that recognizes allowed filenames for your jff homework submissions: a single digit, a dash, another single digit, another dash, one or more letters, a period, and finally the string "`jff`". Pretend SCU usernames are only strings of `j`s and `f`s and there are only 2 homeworks with 2 questions each so you can use the alphabet $\Sigma = \{\mathtt{1}, \mathtt{2}, \mathtt{j}, \mathtt{f}, ., -\}$.

Question 6. Recall that an NFA is a 5-tuple $N = (Q, \Sigma, \delta, q_0, F)$ for finite set of states Q, finite alphabet Σ, transition function $\delta : Q \times \Sigma_\epsilon \to \mathcal{P}(Q)$, start state $q_0 \in Q$, and accept states $F \subseteq Q$. Give a formal specification of an NFA for the language described in Question 4.