Homework 5

Due: Wednesday 10/7 at 2:40pm

Complete the following. Submit your solutions to the last problem using `handin` as assignment `hwk5`. Submit the first two problems on paper or via email.

1. (4 points) Give a DFA or NFA that accepts the language (over alphabet \{0, 1\}) of words that do not have between three and five 0s (i.e. 2 and 6 are fine, but not 3, 4, or 5). You may give the machine either graphically or using textual notation. Include a brief justification that the automata accepts the appropriate language.

2. (6 points) Convert the following NFA into a DFA:

   ![Diagram of NFA](image)

3. (6 points) In class, we’ve talked about how DFAs can be encoded as a 5-tuple. The following is an example of one way such a tuple could be entered into Haskell:

   ```haskell
   even0s = (["even", "odd"], -- states
             "01", -- alphabet
             [("even", '0', "odd"), -- transition function
              ("even", '1', "even"),
              ("odd", '0', "even"),
              ("odd", '1', "odd")],
             "even", -- start state
             ["even"] -- accepting states
   ```

   This definition is for a DFA that accepts words of 0s and 1s that have an even number of 0s. States are represented as strings and the transition function is a list of tuples. For example, the first line of the transition function says that from state `even`, you go to state `odd` on input character 0. Write a Haskell function `validateDFA` that returns whether its input is a valid DFA in the format above. In order to be valid, none of the lists can contain duplicate elements, all combinations of states and input characters must appear in the range of transition function exactly once, and no invalid states or alphabet symbols can appear anywhere. It is not necessary that all the states be reachable or that the set of accepting states be non-empty. Elements of the various lists can also appear in any order.

   Hint: You’ll need helper functions for this task. For example, a function to tell if a particular value occurs in a list would be useful, as would a function to tell if list elements are all unique.