CS 4510 Automata and Complexity 1/10/2024

Homework 1: Regular Languages

Due:1/17/2024

You should submit a typeset or *neatly* written pdf on Gradescope. The grading TA should not have to struggle to read what you've written; if your handwriting is hard to decipher, you will be asked to typeset your future assignments. Four bonus points if you use LaTeX, and our template. You may collaborate with other students, but any written work should be your own.

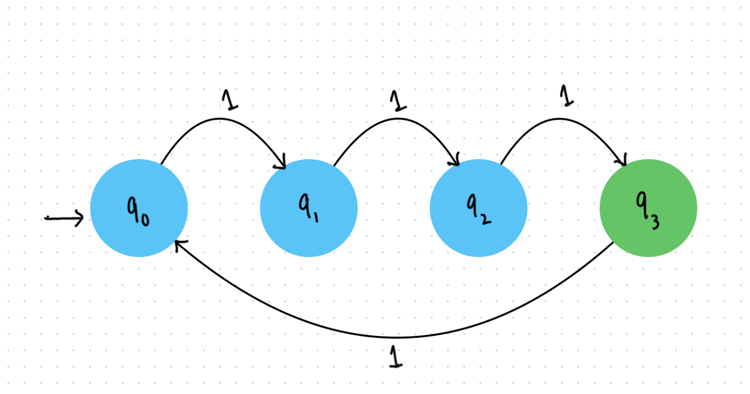
1. One perspective of this class is that we are developing a theory of representation, or definability. For finite sets, this is easy, but it becomes more challenging for infinite languages. This class is really about the ability of different finite objects to understand and describe infinite sets.

   (a) (5 points) Prove every finite language is regular.
   A finite language consists of a limited number of strings. For each string in this finite set, we can construct a DFA that recognizes just that string. By taking the union of these individual DFAs, we can create a DFA that recognizes the entire finite language. Since DFAs are used to recognize regular languages, this proves that every finite language is regular.

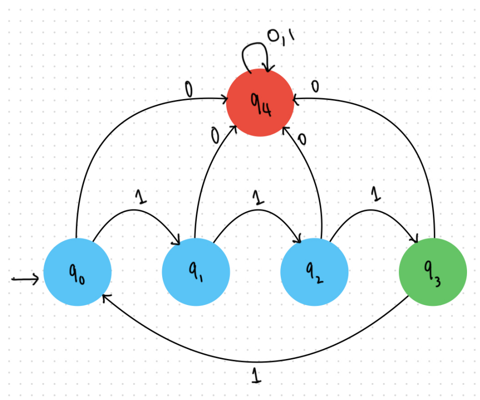
   (b) (5 points) Suppose we relax our definition of DFA too much. We define a DIA to be exactly like a DFA except $|Q| = |\mathbb{N}|$, and δ, F are adjusted appropriately. Prove that every language is decidable by a DIA.
   For any language, no matter how complex, we can theoretically construct a DIA that has a state for every possible string in the language and transitions according to the language's rules. This means that for any given string, the DIA will reach a final state that indicates whether the string is in the language, thus proving that every language is decidable by a DIA.

2. Give an NFA, DFA, and regular expression for the following languages. Your solution doesn't have to be minimal, but it may assist the TA in grading. These may seem similar, but I promise they have different solutions. Here $int(w, 2)$ converts the binary string w to a number.

   (a) (6 points) NFA, DFA, Regex for $L_1 = \{1^n \mid n \equiv 3 \pmod 4\}$
   Green is an accepting state, red is a rejection state

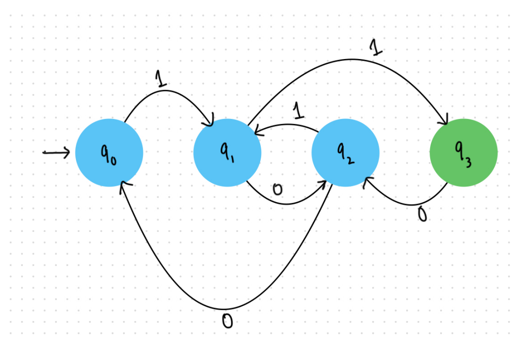
   NFA:

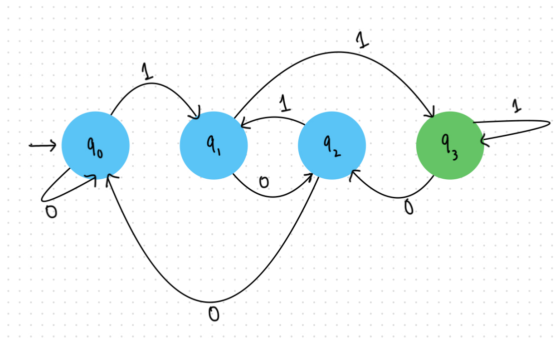
   DFA:

   Regex: $\varepsilon \in 111(1111)^*$

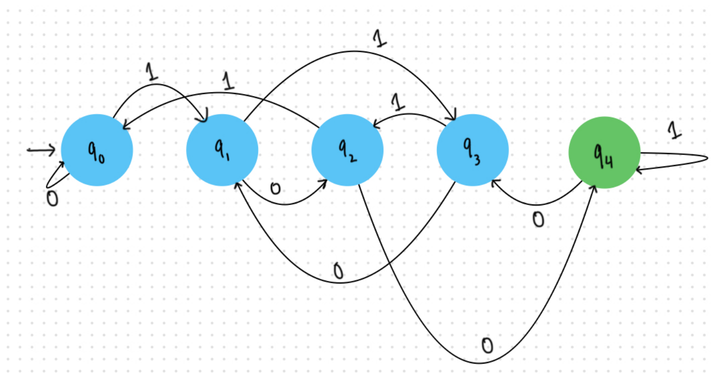
   (b) (6 points) NFA and DFA for $L_2 = \{w \in \Sigma^* \mid int(w, 2) \equiv 3 \pmod 4\}$
   Green is an accepting state

   NFA:

DFA: 

(c) (6 points) DFA for $L_3 = \{w \in \Sigma^* \mid int(w, 2) \equiv 3 \pmod 5\}$
Green is an accepting state

DFA: 

3. (6 points) Let L be a language. Define $L' = \{xay \in \Sigma^* \mid xy \in L, a \in \Sigma\}$. We take the strings in L, and add a single symbol anywhere in the string, even possibly at the beginning or end. Prove that if L is regular, then so is L'.

   Since L is regular, there exists a DFA D that recognizes L. We can construct a DFA for L', calling it D', by modifying D as follows:

   1. For each state q in D, and for each symbol $a \in \Sigma$, add a transition from q to a new state q_a, which represents the insertion of a after the string recognized by q.

   2. Each new state q_a will have transitions similar to the initial state of D, as if we are starting to process a new string after adding a.

   3. If a state q in D is an accepting state, then for each q_a in D', add transitions back to the initial state of D for every symbol in Σ, since adding more symbols after xy would still keep the string in L'.

   By this construction, D' recognizes L', thus proving that if L is regular, then L' is also regular.

4. (6 points) Consider the following poorly drawn gravity powered device. It has two input channels which may receive marbles and two output channels. There are three levers in the shown positions initialized as such. As a marble rolls down, it follows the path the level shows. After a marble hits a lever, the lever flips such that the next marble will go down the opposing channel. Consider a sequence of marbles like the letters of a word. Give a DFA to decide the language of sequences in which the last marble accepts. For example your DFA should accept a, aa, aaa but not $aaaa$. It should accept b, ba but not baa. (Hint: There are more than ten states)
Green is an accepting state, red is a rejecting state