CS 4510 Automata and Complexity 8/22/2024

Homework 1: Finite Automata

Prof. Jaeger *Due: 8/29/2024*

This assignment is due on **10:00 PM ET, Thursday, August 29, 2024**. You may turn it in one day late for no penalty or two days late for a 10% penalty. On-time submissions receive 3% extra credit. Note that a late submission means late feedback, which means less time to study before an exam.

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 required to typeset your future assignments.

You may collaborate with other students, but any written work should be your own. Write the names of the students you work with on the top of your assignment.

0a. **Background Knowledge** (0 points - **NOT GRADED**)

You should be familiar with concepts such as sets and set operations, functions, graphs, and proofs by contradiction and induction. The following problems in the book can help you review them. (Do not turn these in! We will not grade them.)

(0.2) b,e,f; (0.3) e,f; (0.5); (0.6); (0.8); (0.11)

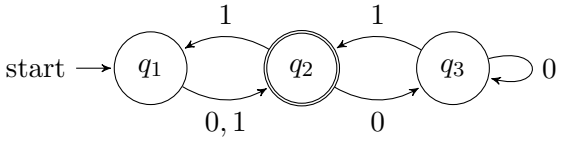
0b. **Practice** (0 points - **NOT GRADED**)

Give DFAs that do the following (easy to hard):

(a) accept all strings, (b) accept all strings not of length exactly 2, (c) accept all strings whose length is one more than a multiple of 3, (d) accept all even length strings which contain an even number of 0's, (e) accept all strings except those containing the substring 1001, (f) accept all binary strings which represent a binary number divisible by 5

0c. **Practice** (0 points - **NOT GRADED**)

Give some strings accepted by this DFA.



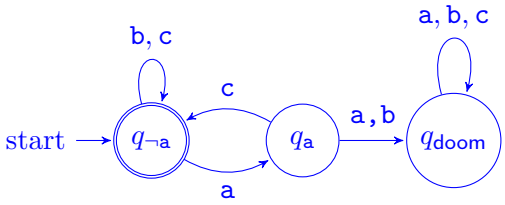
1. **Building DFAs** (20 points - 10 each)

Give DFAs for two of these languages. (Warning: One of them is *not* regular!) Give meaningful names to the states so they are easy to understand.

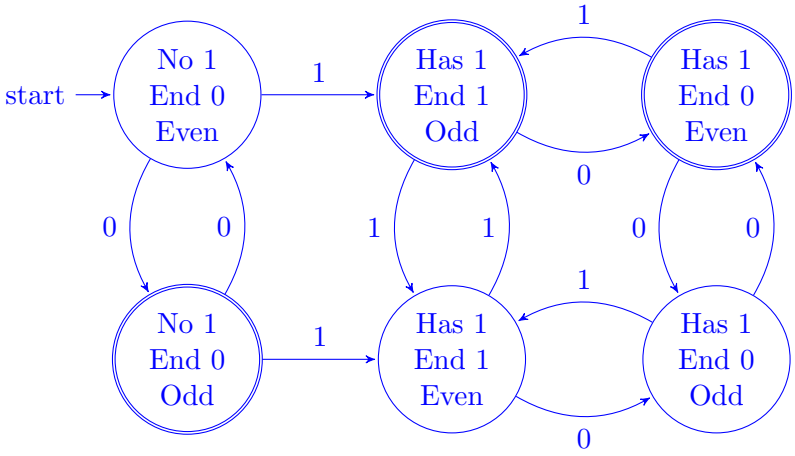
- $L_1 = \{x \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}^* \mid \text{If } x[i] = \mathtt{a}, \text{ then } x[i+1] = \mathtt{c}.\}$.
- $L_2 = \{x \in \{0, 1\}^* \mid$ exactly two of the following are true: x is odd length, x ends with a 0, x contains a $1\}$

- $L_3 = \{x \in \{0,1\}^* \mid x \text{ contains more 1's than 0's } \}$

The following is a DFA for L_1.



The following is a DFA for L_2. (We could have added a separate start state for ε because it doesn't end with 0 or 1, but this already works.)



L_3 is not regular.

2. **Building DFA, Formal Notation** (10 points)

   Let ZERO_7 denote the set of decimals string whose characters multiply to 0 or 1 when taken mod 7, i.e., $\mathsf{ZERO}_7 = \{x \in \{0,1,\ldots,9\}^* \mid \prod_i x[i] \bmod 7 \in \{0,1\}\}$. Prove that ZERO_7 is regular. (Hint: Don't draw a DFA, it'd be too big. Instead, define one explicitly in terms of Q, Σ, δ, q_0, and F.)

   Define $Q = \{0,1,\ldots,6\}$, $\Sigma = \{0,1,\ldots,9\}$, $\delta(q,a) = (q \times a) \bmod 7$, $q_0 = \{1\}$, and $F = \{0,1\}$. Then if $M = (Q, \Sigma, q_0, \delta, F)$ we have $L(M) = \mathsf{ZERO}_7$ so ZERO_7 is regular.

3. **Closure** (5 points) The symmetric difference of A and B, denoted $A \oplus B$, is the set of elements in A or in B, but not in both. For example, if $L = \{\varepsilon, 0\}$ and $L' = \{\varepsilon, 1\}$ then $L \oplus L' = \{0,1\}$. Prove that REG is closed under symmetric difference.

   Note that the symmetric difference of sets A and B can be expressed as

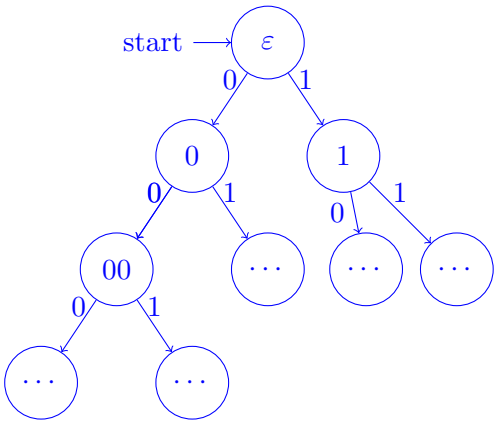
$$A \oplus B = (A \setminus B) \cup (B \setminus A) = (A \cap \overline{B}) \cup (\overline{A} \cap B).$$

Since the set of regular languages is closed under intersection, union, and complement (as shown in class), it follows that regular languages are also closed under symmetric difference.

Alternatively, we could prove this using DFAs. Let L, L' be regular languages and M, M' be DFAs for them. Then we could define D for $L \oplus L'$ as follows (mirroring proofs we did for closure under $\cap$ and $\cup$)

- $D.Q = M.Q \times M'.Q$
- $D.q_0 = (M.q_0, M'.q_0)$
- $D.\Sigma = M.\Sigma = M'.\Sigma$
- $D.\delta((q, q'), x) = (M.\delta(q, x), M'.\delta(q', x))$
- $D.F = (M.F \times (M'.Q \setminus M'.F)) \cup ((M.Q \setminus M.F) \times M'.F))$ (i.e., states (q, q') with $q \in M.F$ or $q' \in M'.F$, but not both).

4. **Infinite DFAs** (10 points) In some senses, the most important characteristic of DFAs is that they only have finite memory ($|Q|$ is finite). Define "DIAs" identically to DFAs except that Q and F are allowed to be infinite. Prove that *every* language $L \subseteq \{0,1\}^*$ is the language of some infinite DIA. (Hint: choose Q carefully to make this easy for yourself.)



Consider a DIA M constructed as an infinite tree as shown above. We have associated a different node to every string in $\{0,1\}^*$ and added transition functions to track what we have read so far. To make it accept a language L, we make states accepting if the string they correspond to is contained in L.

We could formalize this in terms of the formal 5-tuple definition as follows:

- $M.\Sigma = \{0,1\}$
- $M.Q = \{0,1\}^*$
- $M.q_0 = \varepsilon$
- $M.\delta(q, x) = q \circ x$
- $M.F = L$

5. **Teaching** (5 points)

Teach someone (that does not already know what a DFA is) how a simple DFA works, then write a few sentences to answer these questions: Who did you teach? What DFA did you use and how did you motivate it? What did you do well (or poorly) in your instruction?

I taught approximately 100 students enrolled in my 4510B course. I started the lesson by using following DFA which I motived as modeling the behavior of my 14 months old playing with light switched. On the positive side, I like to think that starting a class of with an example based on a cute baby is a good amount of whimsy for welcoming everyone. On the negative side, I had not planned ahead of time how best to show the current state of the DFA while processing the example input so that was a bit of a mess as I flustered my way through a few attempts at that.