CPSC 351 — Tutorial Exercise #4
DFA Design and Verification — Part Two

This exercise is intended to help you to learn how to use the process to design a deterministic finite automaton for a given language, and to prove that your DFA is correct.

Getting Started

This initial problems will probably not be discussed during the tutorial. Please discuss it during office hours with the instructor, if you can, if you have trouble solving it.

1. Let $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$. Consider the deterministic finite automaton M for the language

   $$L = \{\omega \in \Sigma^\star \mid \text{either } \omega \text{ does not include an “a” or } \omega \text{ does not include a “b”}\}$$

   that was developed during Lectures #3 and #4. This corresponded to the picture shown in Figure 1 on page 2.

   In the first problem in the previous tutorial exercise, you were asked to modify this DFA — by changing the set F of accepting states — to produce another DFA, $\widehat{M}$, with the same states, start state and transition function — such that the language of $\widehat{M}$ is

   $$\widehat{L} = \{\omega \in \Sigma^\star \mid \text{either } \omega \text{ includes at least one } \mathtt{a} \text{ or } \omega \text{ include at least one } \mathtt{b} \text{ (or both)}\}.$$

   Describe, as precisely as you can, how you would ***modify the proof*** that $L(M) = L$ (given in a supplement for Lecture #4) to obtain a proof that $L(\widehat{M}) = \widehat{L}$. You should find that *something* has to be changed, but not very much does.

2. You were asked to try to design a DFA for $\widehat{L}$ with a smaller number of steps. If all went well, then you designed a DFA for this language with only two states:

   - A start state, q_{no}, corresponds to the set S_{no} of strings in $\Sigma^\star$ that are *not* in L — so that these strings can only include copies of “c”.
   - Another state, q_{yes}, corresponds to the set $S_{\text{yes}} = \widehat{L}$.

   Proved that the DFA, which you designed, is correct.

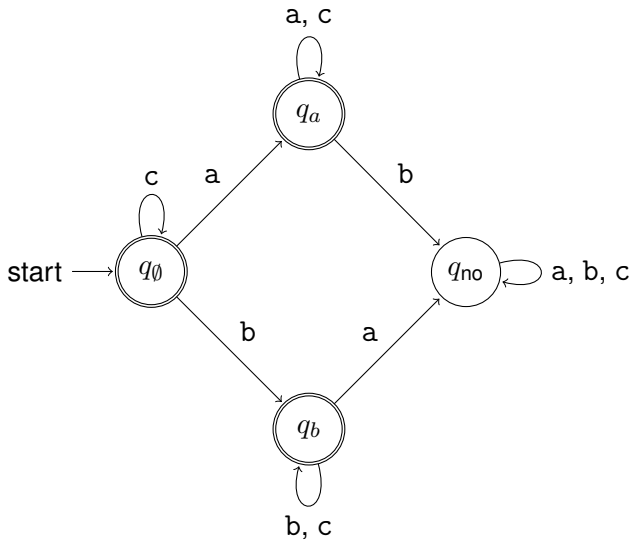

Figure 1: A Deterministic Finite Automaton

More Challenging Problems

At least one of one part of the remaining problem will be discussed the tutorial.

3. Once again, let $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$. *Complete* the design process from Lectures #3 and #4 to design deterministic finite automata for each of the following languages. Then prove that your deterministic finite automaton is correct (that is, that it has the given language).

   (a) $L_1 = \{\omega \in \Sigma^\star \mid \text{The number of } \mathtt{a}\text{'s in } \omega \text{ is divisible by } 4\}$.

   (b) $L_2 = \{\omega \in \Sigma^\star \mid \mathtt{abc} \text{ is a substring of } \omega\}$.