CPSC 351 — Tutorial Exercise #3
DFA Design and Verification — Part One

This exercise is intended to help you to learn how to use the process to design a deterministic finite automaton for a given language.

Getting Started

This initial problem 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.

   Modify this DFA — by changing the set F of accepting states — to produce another DFA, $\widehat{M}$, with the same set of states, start state and transition function — such that the language of $\widehat{M}$ is

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

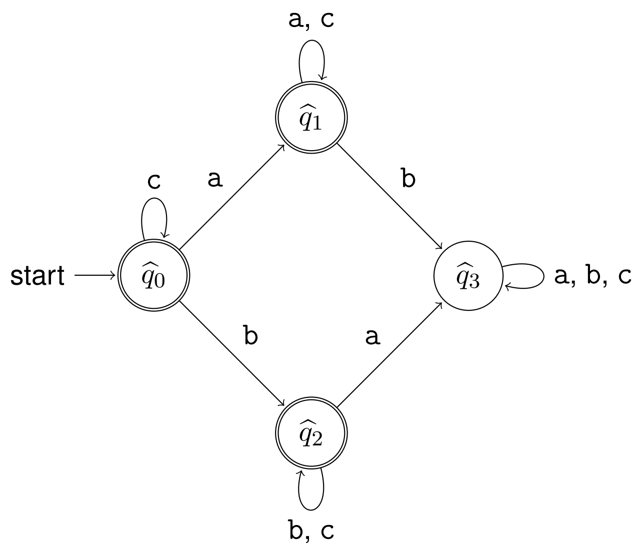
$$\begin{tikzcd}
 & \widehat{q}_1 \arrow[loop above, "\texttt{a, c}"] \arrow[dr, "\texttt{b}"] & \\
\text{start} \arrow[r] & \widehat{q}_0 \arrow[loop above, "\texttt{c}"] \arrow[u, "\texttt{a}"] \arrow[d, "\texttt{b}"'] & \widehat{q}_3 \arrow[loop right, "\texttt{a, b, c}"] \\
 & \widehat{q}_2 \arrow[loop below, "\texttt{b, c}"] \arrow[ur, "\texttt{a}"'] &
\end{tikzcd}$$

Figure 1: A Deterministic Finite Automaton

More Challenging Problems

At least one of the following problems will be considered during the tutorial.

2. Once again, let $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ and let $\widehat{L}$ be the language defined in the previous question. Design a DFA for $\widehat{L}$ using the ***design process*** described in Lecture #3. (You should obtain a different deterministic finite automaton than the one you found when solving the first problem.)

   You should find that you are able to complete the process, this time!

3. Once again, let $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$. Try to use the design process from Lecture #3 to design deterministic finite automata for each of the following languages.

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

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

   If you started under the assumption that the DFA only has to remember whether the string processed so far, belongs to the desired languages, then ***the process should fail, both times***. *Precisely* how does it fail?

Note: The goal, here, is for you to become familiar with the design process that is now being introduced. It is OK if you are not able to *complete* the design process to solve a problem, if you are learning more about the design process because of this!