CPSC 351 — Tutorial Exercise #2
DFA Design and Verification — Part One

About This Exercise

The goal of this exercise is to help you to learn to use the design process, for deterministic finite automata, that has been introduced in lectures.

Problems To Be Solved

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

1. Let $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ and let

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

   Design a DFA for $\widehat{L}$ using the ***design process*** described in Lecture #3.

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

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

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

   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!