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

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

Note: The following result might have been introduced in a course that you have taken before this, and it can be used when completing this exercise.

> ***Division Theorem:*** Let n be any integer and let m be an *positive* integer (so that $m \geq 1$). Then there exists a ***unique*** pair of integers q and r such that $0 \leq r \leq m - 1$ and $n = q \cdot m + r$. (Then n is *congruent* to r, modulo m.)

1. Let $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$. *Complete* the design process (which you started when completing the previous tutorial exercise) 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) $L_2 = \{\omega \in \Sigma^\star \mid \mathtt{abc} \text{ is a substring of } \omega\}$.

2. Use a significant technical result, introduced in the preparatory material for DFA design and verification, to prove that the deterministic finite automata that you designed, above, are *correct* — that is, they have the languages that they are supposed to.

   Note: You *do not* need to use mathematical induction to solve this problem.