CPSC 351 — Tutorial Exercise #3
Additional Practice Problems

About These Problems

These problems will not be discussed during the tutorial, and solutions for these problems will not be made available. They can be used as "practice" problems that can help you practice skills considered in the lecture presentation for Lecture #3, or in Tutorial Exercise #3.

Practice Problems

Each of these problems concern languages over the alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$.

1. Let L_1 be the set of strings $\omega \in \Sigma^\star$ such that ω includes at least four copies of a (which do not have to appear in a row — so that, for example, $\mathtt{aabaca} \in L_1$). Try to design a deterministic finite automaton
$$M_1 = (Q_1, \Sigma, \delta_1, q_{0,1}, F_1)$$
such that $L(M_1) = L_1$.

2. Let L_2 be the set of strings $\omega \in \Sigma^\star$ such that aaaa is a *substring* of ω — that is, $\omega = \mu\,\mathtt{aaaa}\,\nu$ for a pair of strings $\mu, \nu \in \Sigma^\star$. (Thus $\mathtt{aabaca} \notin L_2$.) Try to design a deterministic finite automaton
$$M_2 = (Q_2, \Sigma, \delta_2, q_{0,2}, F_2)$$
such that $L(M_2) = L_2$.

3. Let L_3 be the set of strings $\omega \in \Sigma^\star$ such that ω *ends with* aaaa — so that $\omega = \mu\mathtt{aaaa}$ for some string $\mu \in \Sigma^\star$. Try to design a deterministic finite automaton
$$M_3 = (Q_3, \Sigma, \delta_3, q_{0,3}, F_3)$$
such that $L(M_2) = L_2$.

Note: For *more* practice, you should consider solving each of these problems in *two* ways:

- Try to guess the information that must be remembered correctly, right away, so that only one "attempt" is required. *Try not to go overboard!* If you include more information than necessary then your DFA will be more complicated than necessary, and proving its correctness might become more difficult.
- Always start with as little information as possible — namely, whether the string that has been seen, so far, belongs to the desired language (so that the deterministic finite automaton, that are you are trying to design, would only have two states). If this attempt fails then *try to understand why:* Which transitions are well-defined? Which transitions are *not* well-defined? What extra information is needed?

This will help you to practice different *parts* of the process that you are trying to learn to use.