CSC236 Tutorial
week #12, Winter 2015

1. Recall that for NFAs (AKA nondeterministic finite state machines), we allow states to not only have multiple out-arrows labeled with the same symbol, but also to have *no* out-arrows for some symbols. When a computation path reads a symbol b at a state that has no out-arrow labeled b, that computation path dies (or *is rejecting*) and therefore cannot cause the input string to be accepted by the NFA[1].

   Recall the language, from last week's handout, of binary strings containing the substring "0011". Find an NFA computing that language with the following property: It has only 4 arrows between distinct states. The only other arrows are self-loops.

2. The Myhill-Nerode Theorem can be used to prove that a language is not regular. Consider the language $L := \{s \in \{0,1\}^* \mid \text{s has more 0s than 1s}\}$.

   (a). For every $n \geq 1$, use the Myhill-Nerode Theorem to prove that any DFA computing L must have at least n states. If you get stuck, try doing the proof for $n = 1$ or $n = 2$ first.

   (b). Use (a) to conclude that L is not regular.

3. Are there non-regular languages L_1 and L_2 such that $L_1 \cap L_2$ is regular? If the answer is yes, give a counterexample. If the answer is no, prove that $\forall L_1, L_2.\, L_1, L_2 \text{ not regular} \implies L_1 \cap L_2 \text{ not regular}$.

[1] But recall that some other computation path may cause the input string to be accepted.