Theory of Computation
Midterm Exam

Due: Wednesday, Oct. 7th by 11:59pm

1 Multiple Choice Questions

1. Which of the following is the correct type (signature) of the NFA transition function:

   a. $(Q \times (\Sigma \cup \{\varepsilon\})) \to Q$
   b. $(Q \times (\Sigma \cup \{\varepsilon\})) \to \mathcal{P}(Q)$
   c. $(Q \times \Sigma) \to Q$
   d. $(Q \times (\Sigma^* \cup \{\varepsilon\})) \to \mathcal{P}(Q)$

2. A language is regular if and only if

   a) it is the language of a DFA.
   b) it is the language of a NFA.
   c) all of the above.

3. Instead of this exam, I would rather be doing:

   ____

2 Regular Languages

4. Prove that the following language is regular by constructing a DFA that accepts it:

   $L = \{w \in \{!, \square\}^* \mid w = vw'v \text{ where } v, w' \in \{!, \square\}^*, |w'| = 2, \text{ and } |v| = 3\}$

5. Suppose we have two industrial control systems S_1 and S_2 that are working concurrently. We need to forbid some sequence of actions from taking place between the two systems. One such action is allowing S_1 and S_2 to be on at the same time. Consider the alphabet:

   $\Sigma = \{\mathsf{On}(S_1), \mathsf{On}(S_2), \mathsf{Off}(S_1), \mathsf{Off}(S_2)\}$

   A sequence of actions is a word over Σ^*.

   i. Convert the property on sequences given above into a language, and then define an NFA that accepts it.

   ii. A second property is that S_2 must power off after S_1, but S_2 must power on before S_1. Define a language which captures this property (and this property only), and then define a NFA that accepts it.

6. Suppose the language, L, over an alphabet Σ_0 is regular, and $f : \Sigma_0 \to \Sigma_1$, is a function from the alphabet Σ_0 to the alphabet Σ_1.

   The function f can be lifted to words producing the function:

   $$\begin{aligned}&\hat{f} : \Sigma_0^* \to \Sigma_1^*\\ &\hat{f}(\varepsilon) = \varepsilon\\ &\hat{f}(aw) = f(a)\hat{f}(w)\end{aligned}$$

   Show that the language $L_f = \{\hat{f}(w) \in \Sigma_1^* \mid w \in \Sigma_0^*\}$ is regular.

   Hint: You have to use the formal definition of DFAs.

7. Convert the following NFA into it's equivalent DFA using the NFA-to-DFA algorithm: