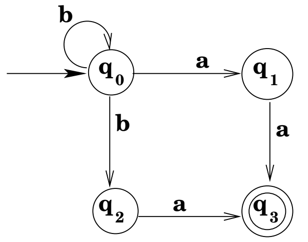
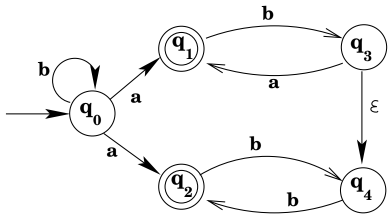
*This homework is due in class on Wednesday, November 19. It covers Chapter 3, Sections 5 and 6. You can work with other people in the class, but you should write up your solutions in your own words to turn in. For the problems on this homework, it is not necessary to explain your work unless you are explicitly asked to do so.*

**1.** Using the algorithm from Section 3.5, convert each of the following NFA's into a DFA that accepts the same language.

**a)**




**b)**




**2.** For each of the following regular expressions over the alphabet $\{a, b\}$, find an NFA that accepts the same language that is generated by the regular expression. Use the algorithm from Section 3.6.

**a)** $(aa + bb)^*$ **b)** $(a + b)aa(a + b)^*$

**3.** Let $M = (Q, \Sigma, q_o, \delta, F)$ be a DFA, and let $S$ be the language accepted by $M$; that is, $S = L(M)$. We have seen that the complement language $\overline{S}$ is accepted by the DFA $M' = (Q, \Sigma, q_o, \delta, \overline{F})$, in which every accepting state in $M$ becomes non-accepting and *vice versa*. Suppose we start with an NFA $N = (Q, \Sigma, q_o, \partial, F)$, with $S = L(N)$. Is it still true that $\overline{S}$ is the language accepted by $N' = (Q, \Sigma, q_o, \partial, \overline{F})$ ? That is, if you interchange the accepting and non-accepting states in an NFA, does the new machine accept the complement of the language accepted by the original machine? Justify your answer.

**4.** Let $L$ be any regular language over an alphabet $\Sigma$. Give an argument to show that $L^R$ is also a regular language.

**5.** Consider the following alphabet, in which each symbol is a stack of three 0's and 1's:

$$\Sigma = \left\{ \begin{bmatrix}0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\1\\1\end{bmatrix}, \begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}1\\0\\1\end{bmatrix}, \begin{bmatrix}1\\1\\0\end{bmatrix}, \begin{bmatrix}1\\1\\1\end{bmatrix} \right\}$$

The addition of two binary numbers can be represented as a string over this $\Sigma$, with each symbol in the string representing one of the columns in the addition. For example,

$$\begin{array}{r} 00100101 \\ \underline{01001011} \\ 01110000 \end{array} \quad \text{is represented as} \quad \begin{bmatrix}0\\0\\0\end{bmatrix} \begin{bmatrix}0\\1\\1\end{bmatrix} \begin{bmatrix}1\\0\\1\end{bmatrix} \begin{bmatrix}0\\0\\1\end{bmatrix} \begin{bmatrix}0\\1\\0\end{bmatrix} \begin{bmatrix}1\\0\\0\end{bmatrix} \begin{bmatrix}0\\1\\0\end{bmatrix} \begin{bmatrix}1\\1\\0\end{bmatrix}$$

A string over $\Sigma$, interpreted as an addition of binary numbers can be either a *correct* addition or an *incorrect* addition. Let $L$ be the language over $\Sigma$ that consists of all strings that represent correct additions. Show that this language is regular by constructing a DFA that accepts the language. If you prefer, you can show that $L^R$ (and therefor also $L$) is regular by constructing a DFA that accepts $L^R$. This shows that the problem of checking whether a binary addition is correct is one that can be solved by a DFA.