Theory of Computation
Homework 6.

Due Date: Wednesday, October 15.
Note unusual date. No class Monday, October 13.

Chapter 2:
No. 21. Hint. If you had access to two DFAs, \( M_A \) recognizing \( A \) and \( M_B \) recognizing \( B \), how would you use them to recognize \( \text{Shuffle}(A,B) \)? What are you using these machines to remember? What else are you remembering?

Nos. 26,27,30. See the sample problem and solution overleaf.

Challenge problem: Chapter 2, no. 28.
Sample Problem  Let $L$ be a regular language. Define Remove-One-Char($L$), or ROC($L$) for short, to be the language containing those strings that can be obtained by removing a single character from a string in $L$; more formally:

$$\text{ROC}(L) = \{uv | uav \in L, \text{ where } u, v \in \Sigma^*, a \in \Sigma\}.$$  

Show that ROC($L$) is also regular. It may be helpful to illustrate your construction with a diagram, but you should provide a reasonably precise explanation so that it is completely clear how your construction works. Remember to give a brief justification of why it works, also.

Sample solution. Let $M$ be a DFA recognizing $L$. We will construct NFA $N$ to accept ROC($L$). The intuitive idea is that $N$ will be obliged to follow an edge in $M$ without reading a character exactly once. This is implemented as follows. The graph for $N$ consists of two copies of the graph for $M$, with the first copy joined to the second by $\lambda$-labeled edges. Hence a path in $N$ transits once from the first copy to the second. Having the start vertex for $N$ in the first copy and the recognizing vertices in the second copy ensures that a string is recognized exactly if it omits one character that would be read on the corresponding path in $M$.

The details follow. $N$ comprises two copies of $M$, named $M_1$ and $M_2$. For each vertex $p$ in $M$ there will be a vertex $p_1$ in $M_1$ and a vertex $p_2$ in $M_2$. And for each $a$-edge $(p, q)$ in $M$ (an $a$-edge is an edge labeled by $a$), in $N$ there will be $a$-edges $(p_1, q_1)$ and $(p_2, q_2)$ plus a $\lambda$-edge $(p_1, q_2)$. For each recognizing vertex $r$ in $M$ there will be a recognizing vertex $r_2$ in $N$, and if $s$ is the start vertex for $M$, $s_1$ is the start vertex for $N$.

Suppose $w$ is recognized by $M$. Let $w = uav$ for some character $a \in \Sigma$ and strings $u, v \in \Sigma^*$ (so $w \neq \lambda$). Then there is a recognizing path $P$ for $w$ which can be partitioned into path $P_u$ labeled by $u$, followed by an $a$-edge $(p, q)$, followed by a path $P_v$ labeled by $v$. Consider the following path $P'$ in $N$: It comprises path $P_u$ in $M_1$, followed by edge $(p_1, q_2)$, followed by path $P_v$ in $M_2$. Clearly $P'$ is a recognizing path in $M'$ and $P'$ has label $uav = uv$. This construction works for any of the characters in $w$, and thus the resulting strings $w' = uv$ form the set ROC($w$), which is therefore recognized by $N$. As this applies to any $w$ recognized by $M$, it follows that ROC($L$) $\subseteq$ L($N$).

On the other hand, if $w'$ is recognized by $N$, then there is a recognizing path $P'$ for $w'$ which comprises a path $P_u$ in $M_1$, followed by an edge $(p_1, q_2)$, followed by a path $P_v$ in $M_2$. Let $u$ be the label on $P_u$, $v$ the label on $P_v$, and $a$ the pseudo-label on $(p_1, q_2)$. Consider the path $P$ in $M$ comprising $P_u$, then edge $(p, q)$, then path $P_v$. It has label $w = uav$ and is recognizing. Clearly $w' \in$ ROC($w$). Thus for each $w' \in L(N)$ there is a $w \in L(M)$ with $w' \in$ ROC($w$). In other words $L(N) \subseteq$ ROC($L$).

This shows that $N$ recognizes ROC($L$), and hence ROC($L$) is regular if $L$ is regular.