Theory of Computation, Homework 1

1.4 g.)
Let $M_1$ recognizes $L_1 = \{ \omega \mid \omega \text{ has even length} \}$, where $M_1 = (Q_1, \{a, b\}, \delta_1, q_1, F_1 = q_1)$, and $M_2$ recognizes $L_2 = \{ \omega \mid \omega \text{ has a odd number of a's} \}$, where $M_2 = (Q_2, \{a, b\}, \delta_2, q_2, F_2)$. DFAs for $L_1$ and $L_2$ are constructed as follows, where $Q_1 = \{q_1, q_{12}, F_1\}$ and $Q_2 = \{q_2, F_2\}$.

Construct $M$ to recognize $L_1 \cap L_2$, where $M = (Q, \{a, b\}, \delta, q_0, F)$.

1. $Q = \{(r_1, r_2) \mid r_1 \in Q_1, r_2 \in Q_2\}$
2. $\Sigma = \{a, b\}$
3. $a \in \Sigma, \delta((r_1, r_2), a) = (\delta_1(r_1, a), \delta_2(r_2, a))$
4. $q_0 = (q_1, q_2)$
5. $F = \{(r_1, r_2) \mid r_1 \in F_1 \text{ and } r_2 \in F_2\}$
L₁=11

L₂=111

L=¬(11∪11)
5-state DFA recognizing no pairs of 1s with an odd number of symbols in between.

1. Step 1: 4-state NFA recognizing a pair of 1 with an odd number of symbols in between.

2. Step 2: Convert the NFA to the corresponding DFA.

3. Step 3: Complement and minimize the states of the DFA above.
1.16
a.)
\[ Q' = \{\phi, \{1\}, \{2\}, \{1,2\}\} \]

<table>
<thead>
<tr>
<th>( \delta )</th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>{1}</td>
<td>{1,2}</td>
<td>{2}</td>
</tr>
<tr>
<td>{2}</td>
<td>\phi</td>
<td>{1}</td>
</tr>
<tr>
<td>{1,2}</td>
<td>{1}</td>
<td>{1,2}</td>
</tr>
<tr>
<td>\phi</td>
<td>\phi</td>
<td>\phi</td>
</tr>
</tbody>
</table>

\( q_0' = \{1\} \)
\( F' = \{\{1\}, \{1,2\}\} \)
b.)

\[ Q' = \{ \phi, \{1,2\}, \{2,3\}, \{1,3\} \} \]

<table>
<thead>
<tr>
<th>( \delta )</th>
<th>( a )</th>
<th>( b )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E({1}) = {1,2} )</td>
<td>{1,3}</td>
<td>( \phi )</td>
</tr>
<tr>
<td>{1,3}</td>
<td>{2,3}</td>
<td>{2,3}</td>
</tr>
<tr>
<td>{2,3}</td>
<td>{1,2}</td>
<td>{2,3}</td>
</tr>
<tr>
<td>( \phi )</td>
<td>( \phi )</td>
<td>( \phi )</td>
</tr>
</tbody>
</table>

\( q_0' = E(\{1\}) = \{1,2\} \)

\( F' = \{ \{1,2\}, \{2,3\} \} \)

---

1.28

a.)

Diagram of a nondeterministic automaton with states labeled by \( \{1,2\}, \{1,3\}, \{2,3\}, \phi \) and transitions labeled by \( a, b, \epsilon \).
1.31

I'll show $A^R$ is regular by constructing its corresponding NFA from the DFA of $A$.

Say if $A$ is regular, there must exist an DFA recognizing $A$.

Let the DFA be $M = (Q, \Sigma, \delta, q_0, F)$ and the corresponding NFA $N = (Q', \Sigma, \delta', q_0', F')$.

step1.) a new start state $q_0'$ is added, connecting each state in $F$ with transitions on $\epsilon$.

Therefore, $Q' = Q \cup q_0'$.

step2.) Let $F' = \{q_0\}$

step3.) $\delta'$ is the reverse of $\delta$, that is, if $\delta'(q, a) = \{q' : \delta(q', a) \neq \epsilon = q}\}$. If $q = q_0'$,

$\delta'(q, \epsilon) = F$. Except the cases, $\delta'(q, a) = \phi$. 

...
Now, support $M$ accepts a string $w_1w_2...w_n$ with state transitions $q_0, q_1, q_2, ..., q_n$. The reverse of the string $w_nw_{n-1}...w_1$ will be accepted by $N$ with state transitions $q'_0, q_n, q_{n-1}, ..., q_0$ because $\delta(q_i, w_{i+1}) = q_{i+1}$ if and only if $q_i \in \delta'(q_{i+1}, w_{i+1})$ for $0 \leq i < n$ and $q_n \in \delta'(q'_0, \epsilon)$. As a result, a corresponding $NFA$ is constructed and recognize $A^R$. So $A^R$ is also regular if $A$ is regular.

1.37

Suppose the binary string is read from the most significant digit to the least significant digit. If the input number is a multiple of $n$, then the number mod $n$ should be 0. So as long as we can construct a DFA $M = (Q, \{0, 1\}, \delta, q_0, F = \{q_0\})$ which verify if the remainder of the input number divided by $n$ is 0, the language $C_n$ is regular.

step 1.) Let $Q = \{q_0, q_0, ..., q_{n-1}\}$ represent the current remainder of the input digits so far mod $n$. That is, each state corresponds to a possible value of the remainder from 0 to $n - 1$.

step 2.) Start and accept states are both $q_0$ meaning the remainder is 0 when the string is accepted.

step 3.) Each time a new digit is read and $q_i$ is the current state, the remainder $r$ is recalculated by computing $2i \mod n$ on input 0 or $(2i + 1) \mod n$ on input 1. The state is then changed from $q_i$ to $q_r$. Formally, $\delta(q_i, 0) = q_{(2i \mod n)}$ and $\delta(q_i, 1) = q_{(2i + 1 \mod n)}$.

Obviously, after all the string is read, the final state will indicate the remainder of the input number mod $n$, identifying if the string belongs to $C_n$. Hence, we find a DFA recognizing language $C_n$. So $C_n$ is regular.

1.41

To show that the class of regular languages is closed under perfect shuffle, one way is to construct a DFA $M = (Q, \Sigma, \delta, q, F)$ recognizing the language. Let $M_A = (Q_A, \Sigma, \delta_A, q_A, F_A)$ and $M_B = (Q_B, \Sigma, \delta_B, q_B, F_B)$ be two DFAs recognizing $A$ and $B$ respectively. $M$ will be a combination of $M_A$ and $M_B$.

step 1.) Let $Q = Q_A \times Q_B \times \{S_A, S_B\}$, where $S_A$ indicates the next symbol to read is in $A$ while $S_B$ indicates that in $B$.

step 2.) Let $q = (q_A, q_B, S_A)$. Assume the first symbol to read is in $A$.

step 3.) Let $F = F_A \times F_B \times \{S_A\}$. If the string is accepted, both $M_A$ and $M_B$ should be in their respective accept states and the next symbol to read is assumed to be in $A$.

step 4.) Let $\delta((x, y, S_A), a) = (\delta_A(x, a), y, S_B)$ and $\delta((x, y, S_B), b) = (x, \delta_B(y, b), S_A)$. Depending on the turn states $S_A$ and $S_B$, $\delta$ chooses $\delta_A$ or $\delta_B$ to apply the transition on the input symbol according the states of $M_A$ and $M_B$.

Apparently, a string in the perfect shuffle of $A$ and $B$ will be accepted by $M$ because both $M_A$ and $M_B$ will reach their accept states respectively.
1.46 a.)
Assume \( L = \{0^n1^m0^n \mid n, m \geq 0\} \) is regular. Let \( n = p \) and \( m = 1 \), where \( p \) is the pumping length of \( L \). Then there must exist \( x, y, z \) such that \( 0^p1 \ 0^p = x \ y^jz \). Since \( |y| > 0 \) and \( |xy| \leq p \), \( y = 0^k \) for \( 0 < k \leq p \). However, \( xy^2z = 0^{p+k}1 \) \( 0^p \notin L \) because there are more 0s before 1 than those 0s after 1. Therefore, \( L \) is not regular.