1. (12 points) For strings $x, y \in \{0, 1\}^*$ such that $|x| = |y|$, the Hamming distance between $x$ and $y$ is the number of bits in which they differ. If $|x| \neq |y|$, then the Hamming distance between $x$ and $y$ is $\infty$. So the Hamming distance between 01101 and 01000 is 2, and between 01101 and 010001 it is $\infty$. For any language $L$ over $\{0, 1\}^*$ and any $k \in \mathbb{N}$, we define $L_k$ as the set of all strings within a Hamming distance at most $k$ from some string in $L$. So if $L = \{\varepsilon, 1, 1001\}$, then $L_0 = \{\varepsilon, 1, 1001, 0001, 1101, 1011, 1000\}$ and $L_1 = \{\varepsilon, 1, 0, 1001, 0011, 1101, 0111, 0011, 1111, 1100, 1010\}$. If $L = 1^*$, then $L_0 = 1^*(0 + \varepsilon)1^*$. Show that for any regular language $L$ over $\{0, 1\}^*$ and any $k \in \mathbb{N}$, $L_k$ must be regular.

(Hint: If $L$ is regular, then how do you know that $L_0$ is regular? Before generalizing to $L_k$, think of how to show that $L_1$ must be regular.)

2. (7 points) For the NFA with $\varepsilon$-transitions $\{(q_0, q_1, q_2), \{0, 1\}, \Delta, \{q_0, q_1, q_2\}, \{q_0, q_2\}\}$, where

$$
\begin{array}{cccc}
\Delta & 0 & 1 & \varepsilon \\
q_0 & \emptyset & \{q_0, q_1\} & \emptyset \\
q_1 & \{q_2\} & \emptyset & \{q_2\} \\
q_2 & \{q_1, q_2\} & \{q_0\} & \emptyset
\end{array}
$$

use the algorithm from class and our text to construct the regular expression $\alpha_{q_0, q_1} + \alpha_{q_1, q_2}$ describing the language defined by the set of paths from $q_0$ to $q_1$ or $q_2$ only possibly passing through $q_0$ or $q_1$ as intermediate states. You don't need to simplify the regular expressions produced by the algorithm, although you may want to simplify as you proceed in order to make subsequent expressions less cumbersome.

3. (12 points) Describe an algorithm that accepts as input an arbitrary regular expression $\alpha$ and an arbitrary NFA $N$, and outputs either that $L(\alpha) \subseteq L(N)$ or produces a string $z$ such that $z \in L(\alpha)$ and $z \notin L(N)$.

4. (6 points) Is the family of regular languages closed under infinite unions? That is, if $L_0, L_1, L_2, \ldots$ are all regular, must $\bigcup_{k \geq 0} L_k$ be regular? Justify your answer.
CS3133
Solutions to HW#4

1. If $L$ is regular, then there is a DFA $M = (Q, \Sigma, \delta, q_0, F)$ to accept $L$. We construct NFA

$$N = \left( \bigcup_{0 \leq j \leq k} \bigcup \{ q_{i,j} \} \right) \cup \Delta \{ q_{0,0} \}, \bigcup \bigcup \{ q_{i,j} \}$$

to accept $L_k$. Intuitively $N$ has $k+1$ "copies" of each $q_i \in Q$. Sitting in state $q_{i,j}$ "means" that the part of string $z$ which has been consumed differs in $j$ places from a string which would have taken $M$ from $q_0$ to $q_i$.

To define $\Delta$, we first note that for each $q_i \in Q$, if $\delta(q_i,0) = \delta(q_i,1)$, then

$$\Delta(q_{i,j},0) = \Delta(q_{i,j},1) = \{ q_{\delta(q_i,0),j} \} \text{ for } 0 \leq j < k.$$ But if $\delta(q_i,0) \neq \delta(q_i,1)$, then for $0 \leq j < k$ we set $\Delta(q_{i,j},0) = \{ q_{\delta(q_i,0),j}, q_{\delta(q_i,0),j+1} \}$ and $\Delta(q_{i,j},1) = \{ q_{\delta(q_i,1),j}, q_{\delta(q_i,1),j+1} \}$. That is, we admit the possibility that the current bit being consumed is one of the at most $k$ bits which differ from a string in $L$. Finally, for each $q_i \in Q$, $\Delta(q_{i,k},0) = \{ q_{\delta(q_i,0),k} \}$ and

$$\Delta(q_{i,k},1) = \{ q_{\delta(q_i,1),k} \}.$$

2. For each expression, whenever there is a second equality it's a simplification which makes future expressions less cumbersome.

$$\alpha_{q_0} = 1 + \varepsilon$$

$$\alpha_{q_1} = \varepsilon$$

$$\alpha_{q_2} = 0 + \varepsilon$$

$$\alpha_{q_3} = 1$$

$$\alpha_{q_4} = \emptyset$$

$$\alpha_{q_5} = \emptyset$$

$$\alpha_{q_6} = 0 + \varepsilon$$

$$\alpha_{q_7} = 1$$

$$\alpha_{q_8} = 0$$

$$\alpha_{q_9} = 1 + \varepsilon + (1 + \varepsilon)(1 + \varepsilon)^* (1 + \varepsilon) = 1$$

$$\alpha_{q_{10}} = 1 + (1 + \varepsilon)(1 + \varepsilon)^* 1 = 1\varepsilon$$

$$\alpha_{q_{11}} = \emptyset + (1 + \varepsilon)(1 + \varepsilon)^* \emptyset = \emptyset$$

$$\alpha_{q_{12}} = \varepsilon + \emptyset (1 + \varepsilon)^* 1 = \varepsilon$$

$$\alpha_{q_{13}} = \emptyset + \emptyset (1 + \varepsilon)^* 1 = \emptyset$$
\[ \alpha_{q_0 q_2} = 0 + \varepsilon + \emptyset (1 + \varepsilon)^* \emptyset = 0 + \varepsilon \]
\[ \alpha_{q_2 q_0} = 0 + \varepsilon + 1 (1 + \varepsilon)^* \emptyset = 0 + \varepsilon \]
\[ \alpha_{q_2 q_1} = 1 + 1 (1 + \varepsilon)^* (1 + \varepsilon) = 1^*1 \]
\[ \alpha_{q_0 q_1} = 0 + 1 (1 + \varepsilon)^* 1 = 0 + 1^*11 \]
\[ \alpha_{q_0 q_1} = 1^*1 + 1^*1 \varepsilon^* \varepsilon = 1^*1 \]
\[ \alpha_{q_0 q_2} = \emptyset + 1^*1 \varepsilon^* (0 + \varepsilon) = 1^*1 (0 + \varepsilon) \]
\[ \alpha_{q_0 q_1} + \alpha_{q_0 q_2} = 1^*1 + 1^*1 (0 + \varepsilon) = 1^*1 (0 + \varepsilon). \]

3. Our algorithm will first construct DFAs \( M_\alpha \) and \( M_N \) such that \( L(M_\alpha) = L(\alpha) \) and \( L(M_N) = L(N) \). Since regular languages are closed under set difference, there is a DFA \( M^* \) to accept \( L(M_\alpha) - L(\alpha) \). By testing for membership in \( L(M^*) \) all strings of length less than or equal to the number of states in \( M^* \), we can decide that if none of them belong to \( L(M^*) \) then \( L(\alpha) \subseteq L(N) \). If a string \( z \in L(M_\alpha) - L(\alpha) \) is found, then it is printed.

4. The family of regular languages is not closed under infinite union. For any \( k \geq 0 \), the language \( L_k = \{0^k1^k \} \) is finite, and hence it is regular. But we have seen that \( \bigcup_{k \geq 0} L_k = \{0^k1^k \mid k \geq 0 \} \) is not regular.