Recursive Functions, Finite-state Machines

Question 1:

Devise a function $p(n)$ such that $p(n)$ is the number of different ways to create postage of n cents using 3-, 4-, and 5-cent stamps.

Proof:

Define $a(n)$ to be the number of ways n-cents can be created with only 3-cent coins.
Let $b(n)$ be the number of ways n-cents can be created with with 4 and 3-cent coins ONLY, using at least one 4-cent coin.
Lastly, let $c(n)$ be the number of ways n-cents can be created with 3, 4, and at least one 5-cent coin.
Clearly, $p(n) = a(n) + b(n) + c(n)$ since each component function counts mutually exclusive combinations.

Now, $a(n) = \begin{cases} 1 & \text{if } 3|n \\ 0 & \text{otherwise.} \end{cases}$
Also, $b(n) = a(n-4) + b(n-4)$ since if $b(n)$ counts combinations with at least one 4-cent coin, removing a 4-cent coin either leaves at least one 4-cent coin ($b(n-4)$) or zero 4-cent coins ($a(n-4)$)
By a similar argument, $c(n) = a(n-5) + b(n-5) + c(n-5)$.

Now we consider base cases for the recursively defined $b(n)$ and $c(n)$.
Trivially, $b(n) = 0 = c(0)$ when $n < 3$.
Now we list values for $n \in \mathbb{N} \cap [3, 7]$ (which only have 1 possible combination each):

n	3-cent coins	4-cent coins	5-cent coins
3	1	0	0
4	0	1	0
5	0	0	1
6	2	0	0
7	1	1	0

Thus, $b(4) = b(7) = 1$ and $c(5) = 1$, and $c(n) = b(n) = 0$ for any other value of $n \leq 7$.
Now consider $c(n)$ for $n > 7$. Since we've defined cases for $n \in \mathbb{N} \cap [1, 7]$, the recursive calls of $c(n)$ will eventually reach a defined base case (since $n \mod 5 < 7$)
A similar argument can be said for $b(n)$.
Thus, $p(n)$ is well defined and is sure to terminate for all $n \in N$.

We know that $p(0) = p(1) = 0$ and $p(n) = 1$ for $n \in \mathbb{N} \cap [3, 7]$. So, all that's left to do is to use strong induction to prove $p(n-1) < p(n)$ for $n > 7$. It suffices to show that one case since we assume under the inductive hypothesis that $p(n*) < p(n-1)$ for all $n* < n-1$.

We will start with the following corollary (which I checked in python, but couldn't do so rigorously):

$$b(n-6) + c(n-6) \leq b(n-5) + c(n-4) \tag{1}$$

Also, note that for any consecutive i, j, k: $a(i) + a(j) + a(k) = 1$ since exactly one of them is divisible by 3.

$$\begin{aligned}
b(n-6) + c(n-6) &\leq b(n-4) + c(n-5) && \text{(from (1))} \\
b(n-5) + b(n-6) + c(n-6) &\leq b(n-4) + b(n-5) + c(n-5) \\
& && \text{(adding } b(n-5) \text{ to both sides)} \\
b(n-5) + b(n-6) + c(n-6) + & \quad\quad b(n-4) + b(n-5) + c(n-5) + \\
a(n-4) + a(n-5) + a(n-6) &\leq a(n-3) + a(n-4) + a(n-5) \\
a(n-1) + b(n-1) + c(n-1) &\leq a(n) + b(n) + c(n) \\
p(n-1) &\leq p(n) && \text{as wanted.}
\end{aligned}$$

Question 2:

Let $\Sigma = \{a, b, c\}$ and $L_{R4} = \{x \in \Sigma^*;\ |x| = 4\ \wedge x = x^R\}$. Prove that any DFA that accepts L_{R4} has at least nine states.

Proof:

Suppose for sake of contradiction that there exists a DFA $= \{Q, \Sigma, \delta, Q_0, F\}$ where $|Q| < 9$ and accepts L_{R4}.

Let $x \in L_{R4}$. By definition, $x = pqqp$ where $p, q \in \{a, b, c\}$. Since $|\Sigma| = 3$, there are $3 \times 3 = 9$ unique options for prefix pq (by multiplication principle).

Since there are less than 9 total states, there must be at least two unique combinations of pq that send the DFA into the same state (by pigeon-hole principle). Say two of these prefixes are p_0q_0 and p_1q_1.

Since they are indistinguishable prefixes, any string of the form p_0q_0w and p_1q_1w will end up in the same state. Let $w = q_0p_0$ and we can see that both $p_0q_0q_0p_0$ and $p_1q_1q_0p_0$ will reach the same state, say $S \in Q$

If S is an accepting state, we reach a contradiction since $p_0q_0 \neq p_1q_1$ and therefore $p_0q_0q_1p_1 \neq (p_0q_0q_1p_1)^R = p_1q_1q_0p_0$ (the DFA accepts a string not in L_{R4})

If S is a non-accepting state, we still reach a contradiction since $p_0q_0q_0p_0$ is clearly in L_{R4} (the DFA does not accept all strings in L_{R4}).

Thus, any DFA that accepts L_{R4} must have at least 9 states.

In general, if $L_{Rn} = \{x \in \Sigma^* \mid x = x^R \text{ and } |x| = n\}$ where $|\Sigma| = 3$, then you will need at least $3^{\lfloor n/2 \rfloor}$ states since those are the number of unique prefixes (just before the half-way point) in L_{Rn}.

Question 3:

Let $\Sigma = \{0,1,2,3,4,5,6,7,8,9\}$, $L_n = \{x \in \Sigma^* |\ x$ represents a number equivalent to n mod 3 in base 10$\}$. Construct M_0 that accepts $L_0 \cup \{\varepsilon\}$, M_1 that accepts L_1 and M_2 that accepts L_2, making sure each machine has exactly 3 states. Then show $\mathbf{Rev}(L_n) = L_n$

Proof:

$$M_0 = \left\{ \begin{array}{ll} Q = \{R_0, R_1, R_2\} & , \\ \Sigma = \{0,1,2,3,4,5,6,7,8,9\} & , \\ \delta = \begin{array}{|c|c|c|c|} \hline \delta & R_0 & R_1 & R_2 \\ \hline 0,3,6,9 & R_0 & R_1 & R_2 \\ \hline 1,4,7 & R_1 & R_2 & R_0 \\ \hline 2,5,8 & R_2 & R_0 & R_1 \\ \hline \end{array} & , \\ Q_0 = \{R_0\} & , \\ F = \{R_0\} & \end{array} \right\}$$

$$M_1 = \left\{ \begin{array}{ll} Q = \{R_0, R_1, R_2\} & , \\ \Sigma = \{0,1,2,3,4,5,6,7,8,9\} & , \\ \delta = \begin{array}{|c|c|c|c|} \hline \delta & R_0 & R_1 & R_2 \\ \hline 0,3,6,9 & R_0 & R_1 & R_2 \\ \hline 1,4,7 & R_1 & R_2 & R_0 \\ \hline 2,5,8 & R_2 & R_0 & R_1 \\ \hline \end{array} & , \\ Q_0 = \{R_0\} & , \\ F = \{R_1\} & \end{array} \right\}$$

$$M_2 = \left\{ \begin{array}{ll} Q = \{R_0, R_1, R_2\} & , \\ \Sigma = \{0,1,2,3,4,5,6,7,8,9\} & , \\ \delta = \begin{array}{|c|c|c|c|} \hline \delta & R_0 & R_1 & R_2 \\ \hline 0,3,6,9 & R_0 & R_1 & R_2 \\ \hline 1,4,7 & R_1 & R_2 & R_0 \\ \hline 2,5,8 & R_2 & R_0 & R_1 \\ \hline \end{array} & , \\ Q_0 = \{R_0\} & , \\ F = \{R_2\} & \end{array} \right\}$$

Let $x \in L$ where $L \in \{L_0, L_1, L_2\}$ and consider x^R. Let $d^*(Q_0, x)$ be the **final state of x**. If we want a DFSM machine M' that takes x^R, simply take the corresponding machine that accepts L (we'll call it M) and:

(1) Choose the final state of x to be the new starting state.

(2) If the final state of x is an accepting state, let $F := \{R_0\}$ (the original starting state).

(3) Invert of all transitions (i.e. If it were a diagram, switch all the arrows backward.)

All these can be done since there are no dead states and all transitions are well-defined.

It is clear that this new machine accepts only x^R since it mimics the backward traversal of x in M.

Now, notice that you get the exact same machine, (albeit, maybe with different labels). Since the labels themselves are arbitrary, and the choice of L is arbitrary, we can see that $M = M'$ and since M' accepts x^R, then it follows that M_0, M_1, M_2 also accepts x^R if they accept x.

Therefore $L_n = \mathbf{Rev}(L_n)$ as wanted.