Problem. 

(a) The given method does not provide a valid definition to extend $\varepsilon$ to $\varepsilon: Q \times S^* \rightarrow Q$.

Basis: $\forall q \in Q$, $\varepsilon(q, \varepsilon)$ is defined.

Induction: Invalid - can not reach a basis as mentioned in class.

What is $\varepsilon(q, a)$ for $q \in Q$ and $a \in \Sigma$?

$\varepsilon(q, a) = \varepsilon(q, \varepsilon a) = \varepsilon(\varepsilon(q, \varepsilon a), a) = \varepsilon(q, a) !$

(b) The given method provides a valid definition to extend $\varepsilon$ to $\varepsilon: Q \times S^* \rightarrow Q$ via induction on its 2nd argument - you may check its basis and induction do give rise to a well-defined function $\varepsilon$ on $Q \times S^*$.

Denote this function by $\delta$: $Q \times \Sigma \rightarrow Q$ as its extension (following the method) $\delta_i: Q \times S_i^* \rightarrow Q$.

We show that $\delta$ (the extension, $\varepsilon$ defined in lectures) and $\delta_i$ agree on $Q \times S_i^*$ (equal as functions: $Q \times S_i^* \rightarrow Q$),

i.e., $\forall q \in Q \forall x \in S_i^* \delta(q, x) = \delta_i(q, x)$.

i.e., $\forall q \in Q \forall x \in S_i^*$ $\exists (q, x)$.

Proof:
We prove the statement above, \( \forall n \geq 0 \) \( P(n) \), by

induction on \( n \) (effectively, induction on \( 1 \times 1 \)).

Base \( n = 0 \): Prove \( P(0) \), i.e., \( \forall x \in \mathbb{E} \, \forall q \in \mathcal{Q} \)

\[
\delta_{0}(q,x) = \delta(q,x)
\]

When \( n = 1 \), \( \forall q \in \mathcal{Q} \),

\[
\delta_{1}(q,x) = \begin{cases} 
3(q,x) = 3 & \text{by the basis of \( \delta \)} \\
\varepsilon_{1}(q,x) = 3 & \text{their inductive definition of \( \delta \) and \( \varepsilon \)}
\end{cases}
\]

Induction step: Prove \( \forall n \geq 0 \) \( P(n) \land P(n+1) \Rightarrow P(n+1) \).

Let \( n \geq 0 \) be arbitrary.

Assume \( P(n) \land P(n+1) \).

Induction hypothesis.

We prove \( P(n+1) \), i.e., \( \forall x \in \mathbb{E} \, \forall q \in \mathcal{Q} \)

\[
\delta_{n+1}(q,x) = \delta_{n+1}(q,x)
\]

Let \( x \in \mathbb{E} \) be arbitrary, and consider another \( q \in \mathcal{Q} \).

We recall \( x = ay \) for some \( a \in \mathbb{E} \) and \( y \in \mathbb{E}^{+} \), then

\[
\delta_{n}(q,x) = \delta_{n}(q,ay)
\]

\[
= \delta_{n}(\delta(q,a),y) \quad \text{inductive definition of} \ \delta_{n}
\]

\[
= \delta_{n}(\varepsilon(q,a),y) \quad \text{by induction hypothesis} \ P(n)
\]

\[
= \delta_{n}(\varepsilon(q,a),y)
\]

Check this equality via the (inductive) definition of \( \delta_{n} \) and \( \varepsilon \) with the steps of \( \delta(q,a) \).
\[ \frac{\hat{\delta}(\hat{\delta}(q, a), y)}{\hat{\delta}(q, a), y} \]

check this equality via the inductive definition of \( \hat{\delta} \)

\[ \frac{\hat{\delta}(\hat{\delta}(q, y), a, y)}{\hat{\delta}(q, y), a, y} \]

check this equality via the inductive definition of \( \hat{\delta} \)

\[ \frac{\hat{\delta}(\hat{\delta}(q, a), y)}{\hat{\delta}(q, a), y} \]

\[ \hat{\delta}(q, ay) = \hat{\delta}(q, x) \]

\[ \frac{\text{this equality requires a proof}}{\text{the property of } \hat{\delta}} \]

\[ \forall u, v \in \Sigma, a \in Q, q \in \Sigma^* \]

\[ \hat{\delta}(q, uv) = \hat{\delta}(\hat{\delta}(q, u), v) \]

by an induction on \( |v| \).

\[ \text{Note: Every step in above derivation is necessary!} \]

This completes the inductive step.

Combining the base and induction step, we prove

\[ \hat{\delta}(q, z) \in \Pi \]

by induction.
(c) Idea:

The basis of the "method" yields a basis for the extended extension.

But can the "method" provide a valid definition to extend $\leq$ to $\leq^*$: $xy \leq^* \rightarrow \phi$?

The given inductive definition is in an unusual form:

$$\forall q \in \phi \forall x, y \in \Sigma^* \exists f(q, xy) = \phi(f(q, y)).$$

1. Does (1) give a well-defined function for $f$?

For $x, y \in \Sigma^*$, $|xy| > 1$,

clearly $\exists w \in \Sigma^*$ other than $x, y$ such that $xy = wz$.

We question if the definition in (1) is dependent on the choice of $x, y \in \Sigma^*$, that is, if for some such $x, y, w, z$ with $xy = wz$, but $f(f(q, x), y) \neq f(f(q, w), z)$ then the definition in (1) would not be a valid definition because it would give different answers for $f(q, xy)$ and $f(q, wz)$, and they are supposed to be the same.

However, you can show that this cannot happen, and this definition is valid.
Problem 3. For all positive integers \( p \geq 3 \), we denote by \( <x> \) the integer corresponding to the string \( x \in \{0,1,\ldots,p-1\}^* \).

For all positive integers \( k \geq 1 \), note that for all \( x \in \{0,1,\ldots,p-1\}^* \) and \( a \in \{0,1,\ldots,p-1\} \),
\[
<x>a = p<x> + a \quad (a \text{ integer}).
\]

Hence,
\[
<x>a \mod k = (p<x> + a) \mod k
\]
\[
= \left\lfloor p \left( <x> \mod \frac{k}{2} \right) + a \right\rfloor \mod k.
\]

This suggests that we can use the states of a deterministic finite automaton DFA \( M \) to remember the mod-\( k \) remainder of \( <x> \) for consumed input \( x \).

The following DFA \( M = (Q, \Sigma, \delta, q_0, A) \) accepts no language \( L_{3,2} \) for \( p = 2 \) and \( k = 3 \). Can you generalize the construction for arbitrary \( k \geq 1 \) and \( p \geq 2 \)?

\[ Q = \{ q_0, q_1, q_2 \}, \quad \Sigma = \{0,1,3 \} \]

\[ \delta : Q \times \Sigma \to Q \] is defined as:
\[
\delta(q_i, a) = q_{(2i+a) \mod 3} \quad \text{for} \quad i \in \{0,1,2\},
\]

and \( A = \{ q_0 \} \)

\[ M : \]
\[
\begin{array}{ccccccc}
\quad & \quad & \quad & \quad & \quad & \quad & \quad \\
\quad & 1 \quad & \quad & \quad & \quad & \quad & \quad \\
\quad & \quad & \quad & \quad & \quad & \quad & \quad \\
\quad & \quad & \quad & \quad & \quad & \quad & \quad \\
\quad & \quad & \quad & \quad & \quad & \quad & \quad \\
\quad & \quad & \quad & \quad & \quad & \quad & \quad \\
\end{array}
\]

\[
L(M) = L_{3,2} \quad (\text{generalize the construction}) \quad \text{for} \quad L_{k, p}.
\]
The proof of Theorem 3 follows from Proposition 3 (i.e., $L(M) = L(N)$).

We can set up an induction argument (or proof).

Similar to the one in Lecture 4, the one in Problem 2 (b).
Problem 4.
Since \( L \) is regular, we get \( L = L(M) \) for some deterministic finite automaton \( M = (Q, \Sigma, \delta, q_0, F) \).

We construct a nondeterministic finite automaton \( N = (Q', \Sigma, \delta', q_0', F') \) that accepts \( K \) (hence, \( K \) is regular).

The idea of constructing such \( N \) is as follows:

- \( N \) starts in the state \((0, q_0)\) and simulates \( M \) for some number of steps.

At some point, which is nondeterministically chosen, \( N \) allows an \( \varepsilon \)-transition from a state of the form \((0, q)\) to either the state \((1, \varepsilon(q, 0))\) or \((1, \varepsilon(q, 1))\) before guessing \( a \) or \( b \) (guessing \( a \)).

Intuitively, \( N \) is consuming no symbol from its input (while "hypothesizing" that \( M \) has consumed some symbol \( a \) (which is either 0 or 1). Then \( N \) simply continues simulating \( M \) on the remainder of the input string.

Also, there is no \( \varepsilon \)-transition leading out of the states of the form \((L, q)\).
The 5-tuple definition of $\mathbf{N}_5$:

$\mathcal{O}_2 = \{0, 13 \times 0\}$

$\mathcal{O}_2 = (0, 9_0)$

$\delta_{2}: \mathcal{O}_2 \times \Sigma \rightarrow \mathcal{P}(\mathcal{O}_2)$ is defined as:

$\forall q \in \mathcal{O}_2 \forall a \in \Sigma : \delta_{2}(0, q, a) = \{(0, s(q, a))\}$

$\forall q \in \mathcal{O}_2 \forall a \in \Sigma : \delta_{2}(0, q, a) = \{(1, 5(q, 0))\}$

$\forall q \in \mathcal{O}_2 \forall a \in \Sigma : \delta_{2}(1, q, a) = \{(1, 8(q))\}$

$\forall q \in \mathcal{O}_2 \delta_{2}(1, q, \varepsilon) = \emptyset$
Problem 5. For each positive integer \( n \geq 1 \), consider the language \( L_n = \{ \sum 1^{n-2i} \} \) over the alphabet \( \Sigma = \{0, 1, 2\} \).

A deterministic finite automaton with \( n \) states — states 0, 1, \ldots, \( n-2 \) for counting the number \( 1 \)'s encountered so far, and an additional state 9, into which it enters so soon seeing \( 2 \)’s

a 0 or more than \( n-2 \) 1’s, is a DFA with \( n \) states that accepts \( L_n \).

On the other hand, any two strings of the form \( 1^i \cdot 1^j \) for \( 0 \leq i < j \leq n-1 \) are "distinguished" by \( 1^{n-2-i} \).

any DFA must have at least \( n \) states.

Follow a similar argument (applying Pumping Principle) in class to show non-regularity of a language.