1) a) False. As a counter-example consider $L_1 = \varepsilon^*$ and $L_2$ to be an arbitrary non-regular language. Then $L_1$ and $L_2; L_1 \cup L_2 = \varepsilon^*$ are regular languages, and $L_2$ is not.

b) False. As a counter-example consider $L_1 = L_2 = \{ \varepsilon \}$, and $L_3 = \emptyset$. As a result, $L_1^* = L_2^* = L_3^* = \{ \varepsilon^* \}$, so $L_1 \subseteq L_2$ and $L_2 \subseteq L_3^*$ but $L_1 \not\subseteq L_3$.

c) True. It is enough to define a finite state automaton, or a regular expression that accepts the language.

Assume that the language $L$ is $L_f = \{ w_1, w_2, \ldots, w_n \}$ where $n$ is the size of the language. Then the following regular expression accepts $L_f$:

$$ R_f = w_1 + w_2 + \ldots + w_n $$

2) a)

b) There are a lot of regular expressions that accept the given language. Some of them are as follows:

- $((x+z)^* (y+z)^* (x+2)^*)^*$
- $((x^* z^* (y+z)^*)^* (x+z+\varepsilon) (y+z+\varepsilon) (x+z+\varepsilon))^*$
- $((x+z+\varepsilon) (y+z+\varepsilon) (x+z+\varepsilon))^*$
3) a) Having \( A_{E} = (Q, \Sigma, \delta_{E}, 0, F) \) as an \( E-NFA \), the NFA which accepts the same language is \( A_{N} = (Q, \Sigma, S_{N}, 0, F) \) with \( S_{N} \) defined as follows.

\[
\forall q \in Q, \forall a \in \Sigma, \quad S_{N}(q, a) = E_{\delta_{E}}(q, a) = E_{\text{Close}(U \delta_{E}(q, a))} \cap E_{\text{Close}(q)}
\]

First we compute \( E_{\text{Close}}(a) \) for all \( a \in \Sigma \):

\[
E_{\text{Close}}(A) = \{ A, B \} \quad E_{\text{Close}}(B) = \{ B \} \quad E_{\text{Close}}(C) = \{ A, B, C \}
\]

Now:

\[
\begin{align*}
S_{N}(A, 0) &= E_{\text{Close}}(\{ A, C \}) = \{ A, B, C \} \\
S_{N}(A, 1) &= E_{\text{Close}}(\{ B \}) = \{ B \} \\
S_{N}(B, 0) &= E_{\text{Close}}(\{ C \}) = \{ A, B, C \} \\
S_{N}(B, 1) &= E_{\text{Close}}(\{ B \}) = \{ B \} \\
S_{N}(C, 0) &= E_{\text{Close}}(\{ A, C \}) = \{ A, B, C \} \\
S_{N}(C, 1) &= E_{\text{Close}}(\{ A, B \}) = \{ A, B \}
\end{align*}
\]

b) \( N \):

[Diagram of the NFA]
3) b) Cont.

transitions over $N_2$:

$$
\begin{align*}
O &\rightarrow A \\
&\rightarrow B \\
&\rightarrow C \\
D &\rightarrow A \\
E &\rightarrow B \\
&\rightarrow C \\
F &\rightarrow B \\
&\rightarrow C \\
G &\rightarrow C
\end{align*}
$$

4) a) we start with the set of all states, in step 0, the set will be separated into two sets of final and non-final states:

\[
\{A, B, C, D, E, F\}
\]

\[
\{A, B, C, F\} \rightarrow \{D, E\}
\]

\[
\{A, C\} \rightarrow \{B, F\}
\]

\[
\{A, C\} \rightarrow \{B\} \rightarrow \{F\}
\]

\[
\{A, C\} \rightarrow \{B\} \rightarrow \{F\} \rightarrow \{D, E\}
\]

\[
\{A, C\} \rightarrow \{B\} \rightarrow \{F\} \rightarrow \{D, E\} \rightarrow \{N \text{ Changes}\}
\]

b) \[
\{\text{abaab, abbbabbab, aaab, aaabab}\} \subseteq \mathcal{L}(A)
\]

\[
\{\text{aaaa, aaaba, ababa, bbbba}\} \cap \mathcal{L}(A) = \emptyset
\]

notice that any string which ends with $a$ is not in $\mathcal{L}(A)$. not in $\mathcal{L}(A)$
3) a) Graph of unit productions:

\[
\begin{align*}
\text{reachability:} & \quad S \Rightarrow^* A, \quad S \Rightarrow^* C, \quad S \Rightarrow^* D \\
& \quad A \Rightarrow^* C, \quad A \Rightarrow^* D \\
& \quad B \Rightarrow^* C
\end{align*}
\]

We get \( G_1 = (V_N, V_T, P_1, S) \) where

\[
V_N = \{S, A, B, C, D\} \quad V_T = \{a, b, c, d\} \quad P \text{ defines as follows:}
\]

\[
S \rightarrow AaBb \mid Da \mid Aa \mid AB \mid aB \mid Ca \mid d \mid dA \\
A \rightarrow Aa \mid AB \mid aB \mid Ca \mid d \mid dA \\
B \rightarrow BD \mid Cd \mid AB \mid aB \mid Ca \\
C \rightarrow AB \mid aB \mid Ca \\
D \rightarrow d \mid dA
\]

b) Generating symbols \( E \) to: \( \{a, b, c, d\} \)

So we Get \( G_2 = (\{S, D\}, \{a, b, c, d\}, P_2, S) \) where \( P_2 \) defines as follows:

\[
\begin{align*}
S & \rightarrow Da \mid Aa \mid d \mid dA \\
A & \rightarrow Aa \mid d \mid dA \\
D & \rightarrow d \mid dA
\end{align*}
\]