1. Structural Induction

(a) Consider the following recursive definition of strings $\Sigma^*$ over the alphabet $\Sigma$.

**Basis Step:** $\varepsilon$ is a string

**Recursive Step:** If $w$ is a string and $a \in \Sigma$ is a character, then $wa$ is a string.

Recall the following recursive definition of the function $\text{len}$:

\[
\text{len}(\varepsilon) = 0 \\
\text{len}(wa) = 1 + \text{len}(w)
\]

Now, consider the following recursive definition:

\[
\text{double}(\varepsilon) = \varepsilon \\
\text{double}(wa) = \text{double}(w)aa.
\]

Prove that for any string $x$, $\text{len}($double$(x)) = 2\text{len}(x)$.

**Solution:**

For a string $x$, let $P(x)$ be "len(doub$x$e$(x)) = 2\text{len}(x)". We prove $P(x)$ for all strings $x \in \Sigma^*$ by structural induction.

**Base Case.** We show $P(\varepsilon)$ holds. By definition $\text{len}($double$(\varepsilon)) = \text{len}(\varepsilon) = 0 = 2\text{len}(\varepsilon)$, as desired.

**Induction Hypothesis.** Suppose $P(w)$ holds for some arbitrary string $w$.

**Induction Step.** We show that $P(wa)$ holds for any character $a \in \Sigma$.

\[
\text{len}($double$(wa)) = \text{len}($double$(w)aa) \tag{By Definition of double}
\]
\[
= 1 + \text{len}($double$(wa)) \tag{By Definition of len}
\]
\[
= 1 + 1 + \text{len}($double$(w)) \tag{By Definition of len}
\]
\[
= 2 + 2\text{len}(w) \tag{By IH}
\]
\[
= 2(1 + \text{len}(w)) \tag{Algebra}
\]
\[
= 2($\text{len}(wa)) \tag{By Definition of len}
\]

This proves $P(wa)$.

Thus, $P(x)$ holds for all strings $x \in \Sigma^*$ by structural induction.

(b) Consider the following definition of a (binary) Tree:

**Basis Step:** $\bullet$ is a Tree.

**Recursive Step:** If $L$ is a Tree and $R$ is a Tree then $\text{Tree}(\bullet, L, R)$ is a Tree.

The function leaves returns the number of leaves of a Tree. It is defined as follows:

\[
\text{leaves}(\bullet) = 1 \\
\text{leaves}($\text{Tree}(\bullet, L, R)) = \text{leaves}(L) + \text{leaves}(R)
\]

Also, recall the definition of size on trees:

\[
\text{size}(\bullet) = 1 \\
\text{size}($\text{Tree}(\bullet, L, R)) = 1 + \text{size}(L) + \text{size}(R)
\]

Prove that $\text{leaves}(T) \geq \text{size}(T)/2$ for all Trees $T$. 

Solution:
In this problem, we define a strengthened predicate. For a tree \( T \), let \( P \) be leaves(\( T \)) \( \geq \) size(\( T \))/2 + 1/2. We prove \( P \) for all trees \( T \) by structural induction.

Base Case. We show that \( P(\cdot) \) holds. By definition of leaves(\( , \) ), leaves(\( \bullet \)) = 1 and size(\( \bullet \)) = 1. So, leaves(\( \bullet \)) = 1 \( \geq \) 1/2 + 1/2 = size(\( \bullet \))/2 + 1/2.

Induction Hypothesis: Suppose \( P(L) \) and \( P(R) \) hold for some arbitrary trees \( L \) and \( R \).

Induction Step: We prove that \( P(\text{Tree}(\bullet, L, R)) \) holds.

\[
\text{leaves(} \text{Tree}(\bullet, L, R)\text{)} = \text{leaves}(L) + \text{leaves}(R) \quad \text{[By Definition of leaves]}
\geq (\text{size}(L)/2 + 1/2) + (\text{size}(R)/2 + 1/2) \quad \text{[By IH]}
= (\text{size}(L) + \text{size}(R) + 1)/2 + 1/2
= \text{size}(\text{Tree}(\bullet, L, R))/2 + 1/2 \quad \text{[By Definition of size]}
\]

This proves \( P(\text{Tree}(\bullet, L, R)) \).

Thus, the \( P(T) \) holds for all trees \( T \).

2. Regular Expressions
(a) Write a regular expression that matches base 10 non-negative numbers (e.g., there should be no leading zeroes).

Solution:
\[
0 \cup ((1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)^*)
\]

(b) Write a regular expression that matches all non-negative base-3 numbers that are divisible by 3.

Solution:
\[
0 \cup ((1 \cup 2)(0 \cup 1 \cup 2)^*0)
\]

(c) Write a regular expression that matches all binary strings that contain the substring "111", but not the substring "000".

Solution:
\[
(01 \cup 001 \cup 1^*)^*(0 \cup 00 \cup \varepsilon)111(01 \cup 001 \cup 1^*)^*(0 \cup 00 \cup \varepsilon)
\]
(If you don’t want the substring 000, the only way you can produce 0s is if there are only one or two 0s in a row, and they are immediately followed by a 1 or the end of the string.)