Section 08: Induction, Regular Expressions, CFGs

1. Regular Expressions

(a) Write a regular expression that matches base 10 numbers (e.g., there should be no leading zeroes).

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

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

(d) Write a regular expression that matches all binary strings that do not have any consecutive 0's or 1's.

(e) Write a regular expression that matches all binary strings of the form $1^k y$, where $k \geq 1$ and $y \in \{0,1\}^*$ has at least k 1's.

2. CFGs

Write a context-free grammar to match each of these languages.

(a) All binary strings that start with 11.

(b) All binary strings that contain at most one 1.

(c) All strings over 0, 1, 2 with the same number of 1s and 0s and exactly one 2.
Hint: Try modifying the grammar from Section 8 2c for binary strings with the same number of 1s and 0s (You may need to introduce new variables in the process).

3. Recursively Defined Sets of Strings

For each of the following, write a recursive definition of the sets satisfying the following properties. Briefly justify that your solution is correct.

(a) Binary strings of even length.

(b) Binary strings not containing 10.

(c) Binary strings not containing 10 as a substring and having at least as many 1s as 0s.

(d) Binary strings containing at most two 0s and at most two 1s.

4. More CFGs

Write a context-free grammar to match each of these languages.

(a) All binary strings that end in 00.

(b) All binary strings that contain at least three 1's.

(c) All binary strings with an equal number of 1's and 0's.

(d) All binary strings of the form xy, where $|x| = |y|$, but $x \neq y$.

5. Reversing a Binary Tree

Consider the following definition of a (binary) **Tree**.

Basis Step `Nil` is a **Tree**.

Recursive Step If L is a **Tree**, R is a **Tree**, and x is an integer, then $\texttt{Tree}(x, L, R)$ is a **Tree**.

The sum function returns the sum of all elements in a **Tree**.

$$\begin{aligned} \mathsf{sum}(\texttt{Nil}) &= 0 \\ \mathsf{sum}(\texttt{Tree}(x, L, R)) &= x + \mathsf{sum}(L) + \mathsf{sum}(R) \end{aligned}$$

The following recursively defined function produces the mirror image of a **Tree**.

$$\begin{aligned} \mathsf{reverse}(\texttt{Nil}) &= \texttt{Nil} \\ \mathsf{reverse}(\texttt{Tree}(x, L, R)) &= \texttt{Tree}(x, \mathsf{reverse}(R), \mathsf{reverse}(L)) \end{aligned}$$

Show that, for all **Tree**s T that

$$\mathsf{sum}(T) = \mathsf{sum}(\mathsf{reverse}(T))$$