Quiz Section 7: Induction, Regular Expressions

Task 1 – Walk the Dawgs

Suppose that a dog walker takes care of $n \geqslant 12$ dogs. The dog walker is not a strong person, and will walk dogs in groups of 3 or 7 at a time (every dog gets walked exactly once). Prove that the dog walker can always split the n dogs into groups of 3 dogs or 7 dogs.

Task 2 – Seeing double

Consider the following recursive definition of strings.

Basis Step: "" is a string

Recursive Step: If X is a string and c is a character then append(c, X) is a string.

Recall the following recursive definition of the function len:

$$\begin{aligned} \text{len}(\texttt{""}) &= 0 \\ \text{len}(\text{append}(c, X)) &= 1 + \text{len}(X) \end{aligned}$$

Now, consider the following recursive definition:

$$\begin{aligned} \text{double}(\texttt{""}) &= \texttt{""} \\ \text{double}(\text{append}(c, X)) &= \text{append}(c, \text{append}(c, \text{double}(X))). \end{aligned}$$

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

Task 3 – Leafy Trees

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 $\texttt{Tree}(L, R)$ is a **Tree**.

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

$$\begin{aligned} \text{leaves}(\bullet) &= 1 \\ \text{leaves}(\texttt{Tree}(L, R)) &= \text{leaves}(L) + \text{leaves}(R) \end{aligned}$$

Also, recall the definition of size on trees:

$$\begin{aligned} \text{size}(\bullet) &= 1 \\ \text{size}(\texttt{Tree}(L, R)) &= 1 + \text{size}(L) + \text{size}(R) \end{aligned}$$

Prove that $\text{leaves}(T) \geqslant \text{size}(T)/2 + 1/2$ for all Trees T.

Task 4 – Reversing a Binary Tree

Consider the following definition of a **Tree** that has integer values at its nodes in which each node has at most two children.

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

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

Task 5 – 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.

Task 6 – 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".