CSE 311: Foundations of Computing I

Section : Structural Induction and Regular Expressions

0. Structural Induction

(a) Recall the recursive definition of a list:

$$\textbf{List} = [\,] \mid \text{Int} :: \text{List}$$

And the definition of len on lists:

$$\begin{aligned} \text{len}([\,]) &= 0 \\ \text{len}(x :: L) &= 1 + \text{len}(L) \end{aligned}$$

Consider the following recursive definition:

$$\begin{aligned} \text{stutter}([\,]) &= [\,] \\ \text{stutter}(x :: L) &= x :: x :: \text{stutter}(L) \end{aligned}$$

Prove that $\text{len}(\text{stutter}(L)) = 2\text{len}(L)$ for all Lists L.

(b) Consider the recursive definition of a tree:

$$\textbf{Tree} = \texttt{Nil} \mid \texttt{Tree}(\textbf{Integer}, \textbf{Tree}, \textbf{Tree})$$

And the definition of size on trees:

$$\begin{aligned} \text{size}(\texttt{Nil}) &= 0 \\ \text{size}(\texttt{Tree}(x, L, R)) &= 1 + \text{size}(L) + \text{size}(R) \end{aligned}$$

And the definition of height on trees:

$$\begin{aligned} \text{height}(\texttt{Nil}) &= 0 \\ \text{height}(\texttt{Tree}(x, L, R)) &= 1 + \max(\text{height}(L), \text{height}(R)) \end{aligned}$$

Prove that $\text{size}(T) \leq 2^{\text{height}(T)+1} - 1$ for all Trees T.

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".