CSE 311: Foundations of Computing I

Homework 6 (due November 21st at 11:00 PM)

Directions: *Write up carefully argued solutions to the following problems. Each solution should be clear enough that it can explain (to someone who does not already understand the answer) why it works. However, you may use results from lecture, the reference sheets, and previous homeworks without proof.*

1. Barking Up the Strong Tree (20 points)

Consider the function $g(m)$ defined for $m \in \mathbb{N}$ recursively as follows:

$$\begin{aligned} g(0) &::= 0 && \text{case: } m = 0 \\ g(2j) &::= g(j) + 1 && \text{case: } m > 0 \text{ and even} \\ g(2j+1) &::= g(2j) && \text{case: } m > 0 \text{ and odd} \end{aligned}$$

The first line gives the definition of $g(m)$ for $m = 0$, the second line gives the definition for even (non-zero) m, and the third line gives the definition for odd m. Those three cases are mutually exclusive and exhaustive, so they define g completely.

Use strong induction to prove that

$$\forall n \in \mathbb{N}\, (g(n) \leq n)$$

Hint: Since g is defined separately for even and odd values, you may need to split your inductive step into two cases (even and odd) as well.

2. Live Strong and Prosper (20 points)

Consider the function $f(n, x)$, taking inputs $n \in \mathbb{N}$ and $x \in \mathbb{R}$, defined recursively as follows:

$$\begin{aligned} f(0, x) &::= x && \text{case: } n = 0 \\ f(2j, x) &::= f(j, x+1) && \text{case: } n > 0 \text{ and even} \\ f(2j+1, x) &::= f(2j, x) && \text{case: } n > 0 \text{ and odd} \end{aligned}$$

As in Problem 1, these three cases are mutually exclusive and exhaustive, so they define f completely.

Use strong induction on n (not x) to show that

$$\forall n \in \mathbb{N}, \forall x \in \mathbb{R}\, (f(n, x) \geq x)$$

For your induction argument, let $P(n)$ be "$\forall x \in \mathbb{R}\, (f(n, x) \geq x)$". Note that this $P(n)$ has a quantifier in it, so each proof of $P(k)$ should be using Intro $\forall$, translated to English.

Recall the definition of lists of numbers from lecture:

Basis Step: $\texttt{nil} \in$ **List**
Recursive Step: for any $a \in \mathbb{Z}$, if $L \in$ **List**, then $a :: L \in$ **List**.

For example, the list $[1, 2, 3]$ would be created recursively from the empty list as $1 :: (2 :: (3 :: \texttt{nil}))$. We will consider "::" to associate to the right, so the simpler expression $1 :: 2 :: 3 :: \texttt{nil}$ means the same thing.

The next two problems use two recursively-defined functions. The first is concat, which concatenates two lists into a single list. It is defined recursively as follows:

$$\begin{aligned} \mathsf{concat}(\texttt{nil}, R) &::= R && \forall R \in \textbf{List} \\ \mathsf{concat}(a :: L, R) &::= a :: \mathsf{concat}(L, R) && \forall a \in \mathbb{Z}, \forall L, R \in \textbf{List} \end{aligned}$$

For example, we get $\mathsf{concat}([1, 2], [3]) = \mathsf{concat}(1 :: 2 :: \texttt{nil}, 3 :: \texttt{nil}) = 1 :: 2 :: 3 :: \texttt{nil}$ from these definitions.

The second function is sum, which adds up the numbers in a list. It is defined as:

$$\begin{aligned} \mathsf{sum}(\texttt{nil}) &::= 0 \\ \mathsf{sum}(a :: L) &::= a + \mathsf{sum}(L) && \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \end{aligned}$$

For example, from these definitions, we get $\mathsf{sum}([1, 2, 3]) = 6$.

3. List Me By a Mile (20 points)

(a) [2 Points] Write a chain of equalities, citing the appropriate definitions, showing that

$$\text{concat}(1 :: 2 :: \texttt{nil}, 3 :: \texttt{nil}) = 1 :: 2 :: 3 :: \texttt{nil}$$

(b) [2 Points] Write a chain of equalities, citing the appropriate definitions, showing that

$$\text{sum}(1 :: 2 :: 3 :: \texttt{nil}) = 6$$

(c) [16 Points] Let $R \in$ **List**. Use structural induction to prove the following

$$\forall L \in \textbf{List}\ (\text{sum}(\text{concat}(L, R)) = \text{sum}(L) + \text{sum}(R))$$

4. List Me, List Me, Now You Gotta Kiss Me (18 points)

Let $R, S \in$ **List**. Use structural induction on L to prove that

$$\forall L \in \textbf{List}\ (\text{concat}(\text{concat}(L, R), S) = \text{concat}(L, \text{concat}(R, S))$$

If we write concat as "+", then this says $(L + R) + S = L + (R + S)$. In other words, concat is associative.

5. A Few of My Favorite Strings (12 points)

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

(a) [4 Points] Binary strings where every 0 is immediately followed by a 1.

(b) [4 Points] Binary strings that start with 0 and have even length.

(c) [4 Points] Binary strings with an even number of 0s.

6. Hope Strings Eternal [Online] (10 points)

For each of the following, construct regular expressions that match the given set of strings:

(a) [2 Points] Binary strings where every 0 is immediately followed by a 1.

(b) [2 Points] Binary strings that start with 0 and have even length.

(c) [2 Points] Binary strings with an even number of 0s

(d) [2 Points] Binary strings with at least three 0s.

(e) [2 Points] Binary strings with at least three 0s **or** at most two 1s.

> Submit and check your answers to this question here:
>
> https://grin.cs.washington.edu/
>
> Think carefully about your answer to make sure it is correct before submitting. You have only 5 chances to submit a correct answer.

7. Extra Credit: Stone By the Company He Keeps (0 points)

Consider an infinite sequence of positions $1, 2, 3, \ldots$ and suppose we have a stone at position 1 and another stone at position 2. In each step, we choose one of the stones and move it according to the following rule: Say we decide to move the stone at position i; if the other stone is not at any of the positions $i+1, i+2, \ldots, 2i$, then it goes to $2i$, otherwise it goes to $2i+1$.

For example, in the first step, if we move the stone at position 1, it will go to 3 and if we move the stone at position 2 it will go to 4. Note: no matter how we move the stones, they will never be at the same position.

Use induction to prove that, for any given positive integer n, it is possible to move one of the stones to position n. For example, if $n = 7$ first we move the stone at position 1 to 3. Then, we move the stone at position 2 to 5 Finally, we move the stone at position 3 to 7.