# Problem Set 6

Due: Wednesday, May 17, by 11:59pm

## Instructions

**Solutions submission.** You must submit your solution via Gradescope. In particular:

- Submit a single PDF file in Gradescope containing the written solution to all the regular tasks 1-5 in the homework.
- Submit your solution to Task 6 online to grin.cs as described in the task itself.
- The extra credit is submitted separately in Gradescope

## Task 1 – Happily Ever After [20 pts]

Let $S$ be defined as follows.

**Basis Step:** $3 \in S$; $5 \in S$

**Recursive Step:** if $x, y \in S$, then $x + y \in S$.

Prove that, for all integers $n \geqslant 8$, we have $n \in S$.

*Hint*: Strong induction is the right tool here since the quantifier is not over $S$ (so structural induction does not apply). You will probably want to use multiple base cases until you have enough to be able apply the inductive hypothesis.

## Task 2 – Barking Up the Strong Tree [20 pts]

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

$$\begin{array}{rcll} g(0) & = & 0 & \text{case: } n = 0 \\ g(n) & = & g(n/2) + 1 & \text{case: } n > 0 \text{ and even (so } n/2 > 0 \text{ is an integer),} \\ g(n) & = & g(n-1) & \text{case: } n > 0 \text{ and odd} \end{array}$$

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

Use strong induction to prove that

$$\forall n \geqslant 1 \, (2^{g(n)} \leqslant n < 2^{g(n)+1}).$$

*Hint*: You will need to use the fact that for natural numbers $k$ and $m$ if $k+1$ is odd and $k+1 < 2^{m+1}+1$ then $k+1 < 2^{m+1}$. (This follows since $k+1$ is odd and $2^{m+1}$ is even which means they can't be equal.)

## Task 3 – List your Associates [18 pts]

Recall the definition of lists of numbers from lecture:

**Basis Step**: $\texttt{nil} \in \textbf{List}$

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

and the function concat, which concatenates two lists into a single list is defined recursively as follows:

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

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

Now for your task: Let $R, S \in \textbf{List}$. 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 that $(L + R) + S = L + (R + S)$. In other words, we are asking you to prove that concat is associative.

## Task 4 – Almost Heaven, Fibonacci (Trees) [20 pts]

Recall the set of rooted binary trees from lecture, denoted Tree, defined by recursion as follows.

**Basis:** The tree with one node, denoted by $\bullet$, is a Tree.

**Recursive step:** If $T_1$ and $T_2$ are rooted binary trees, then $\texttt{Tree}(T_1, T_2)$ is a Tree. It represents the tree with a new root node whose left child is $T_1$ and whose right child is $T_2$.

Also recall the definition of the size and height functions on trees:

$$\text{size}(\bullet) := 1$$
$$\text{size}(\texttt{Tree}(T_1, T_2)) := 1 + \text{size}(T_1) + \text{size}(T_2)$$

$$\text{height}(\bullet) := 0$$
$$\text{height}(\texttt{Tree}(T_1, T_2)) := 1 + \max(\text{height}(T_1), \text{height}(T_2))$$

Now let $A$, the *almost balanced binary trees*, be a subset of Tree defined by recursion as follows.

**Basis:** The tree $\bullet$ is in $A$.

**Recursive step:** If $T_1$ is in $A$ and $T_2$ is in $A$ and either $\text{height}(T_1) = \text{height}(T_2)$ or $\text{height}(T_1)$ and $\text{height}(T_2)$ differ by 1, then $\texttt{Tree}(T_1, T_2)$ is in A.

Prove by induction that for all $T \in A$,

$$\text{size}(T) \geqslant f_{\text{height}(T)+1},$$

where $f_m$ denotes the $m$-th Fibonacci number.

(As usual $f_0 = 0$, $f_1 = 1$, and $f_m = f_{m-1} + f_{m-2}$ for $m \geqslant 2$.)

## Task 5 – A Few of My Favorite Strings [12 pts]

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)** Binary strings where every $0$ is immediately followed by a $1$.

**b)** Binary strings that start with 0 and have even length.

**c)** Binary strings with an even number of 0s.

## Task 6 – Hope Strings Eternal [10 pts]

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

**a)** Binary strings where every $0$ is immediately followed by a $1$.

**b)** Binary strings that start with $0$ and have even length.

**c)** Binary strings of length at least $2$ with an even number of 0s that begin and end with the same character.

**d)** Binary strings with at least three 0s.

**e)** 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. (Note: don't include unnecessary spaces.)
You have only 3 chances to submit a correct answer.

## Task 7 – Extra Credit: Completely Stoned

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 $7$.