CSE 311: Foundations of Computing I

Homework 6 (due May 22nd at 11:00 PM)

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

1. Bound To Happen (20 points)

The running time of an algorithm as a function of the input size, $T(n)$, satisfies the following equations:

$$
\begin{aligned}
T(n) &= 1 && \text{if } 1 \leq n \leq 8 \\
T(n) &= T(\lfloor n/4 \rfloor) + 2 \cdot T(\lfloor n/8 \rfloor) + 5n && \text{for all } n > 8
\end{aligned}
$$

Use strong induction to prove this *upper bound* on the running time: for all $n \geq 1$, we have $T(n) \leq 10n$.

Hint: The only fact about $\lfloor \ \rfloor$ that you will need is that when $x \geq 1$, $\lfloor x \rfloor$ is an integer and $1 \leq \lfloor x \rfloor \leq x$.

(This running time could arise, for example, if the algorithm operates on an input of size n by, first, calling itself recursively on an input of $1/4$-th that size; then, calling itself recursively on *two* inputs of $1/8$-th that size; and finally, performing additional processing that takes $5n$ steps.)

2. Sum People Have All The Luck (20 points)

Let S be the set defined as follows:

Basis Step: $5 \in S$; $9 \in S$;

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

Show that, for every integer $n \geq 32$, it holds that $n \in S$.

Hint: Strong (not structural) induction is the right tool here since the quantifier is over n (not S).

3. Pulling Some Strings (20 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) [5 Points] Binary strings with odd length.

(b) [5 Points] Binary strings that start with 1 and have odd length.

(c) [5 Points] Binary strings with an even number of 0s.

(d) [5 Points] Binary strings where every occurrence of a 0 is immediately followed by 11.

4. Tied Up With Strings [Online] (21 points)

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

(a) [3 Points] Binary strings with odd length.

(b) [3 Points] Binary strings that start with 1 and have odd length.

(c) [3 Points] Binary strings where every occurrence of a 0 is immediately followed by a 11.

(d) [3 Points] Binary strings where every occurrence of a 00 is immediately followed by a 1.

(e) [3 Points] Binary strings with at least two 0s.

(f) [3 Points] Binary strings with at most two 1s.

(g) [3 Points] Binary strings with at least two 0s **or** at most two 1s.

> Submit and check your answers to this question here:
>
> `https://grinch.cs.washington.edu/cse311/regex`
>
> Think carefully about your answer to make sure it is correct before submitting. You have only 5 chances to submit a correct answer.

5. Tree, Two, One (19 points)

We then define the set **Trees** as follows:

Basis Elements: for any $x \in \mathbb{Z}$, Leaf$(x) \in$ **Trees**.

Recursive Step: for any $x \in \mathbb{Z}$, if $L, R \in$ **Trees**, then Branch$(x, L, R) \in$ **Trees**.

This defines a set of *binary trees* just like those defined in lecture except that we now include data in the nodes.

The following functions count the number of each type of node in the tree:

$$\begin{aligned} \text{leaves}(\text{Leaf}(x)) &= 1 && \text{for any } x \in \mathbb{Z} \\ \text{leaves}(\text{Branch}(x, L, R) &= \text{leaves}(L) + \text{leaves}(R) && \text{for any } x \in \mathbb{Z} \text{ and } L, R \in \textbf{Trees} \end{aligned}$$

$$\begin{aligned} \text{branches}(\text{Leaf}(x)) &= 0 && \text{for any } x \in \mathbb{Z} \\ \text{branches}(\text{Branch}(x, L, R) &= 1 + \text{branches}(L) + \text{branches}(R) && \text{for any } x \in \mathbb{Z} \text{ and } L, R \in \textbf{Trees} \end{aligned}$$

Prove that, for any $T \in$ **Trees**, we have leaves$(T) =$ branches$(T) + 1$.

6. Extra Credit: Sticks And Stones (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 that, 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.