Homework 7: Structural Induction, Regexes

In general, you are graded on both the clarity and accuracy of your work. Your solution should be clear enough that someone in the class who had not seen the problem before would understand it.

We sometimes describe approximately how long our explanations are. These are intended to help you understand approximately how much detail we are expecting. You are allowed to have longer explanations, but explanations significantly longer than necessary may receive deductions.

To help with formatting of English proofs, we've published a style guide on the website containing some tips. **Unless otherwise noted in a problem, all proofs must be English proofs.** You should not have numbered steps (e.g., you should not be doing an inference proof.)

Finally, be sure to read the grading guidelines for more information on what we're looking for.

You must use structural induction for problems 1 and 2.

1. Walk the walk, talk the talk [20 points]

Let S be a subset of $\mathbb{Z} \times \mathbb{Z}$ defined recursively as:

Basis Step: $(-1, 0) \in S$ and $(0, -1) \in S$
Recursive Step: if $(a, b) \in S$ and $(c, d) \in S$, then $(a, b) + (c, d) = (a + c, b + d) \in S$.

We claim that $S = (\mathbb{Z}_{\leq 0} \times \mathbb{Z}_{\leq 0}) \setminus \{(0, 0)\}$; that is, S is the set of integer coordinates in the lower left quadrant (excluding the origin). It is easy to show that any such point is in S: if (x, y) is in this quadrant, then $x, y \leq 0$, meaning that:

$$(x, y) = |x| \cdot (-1, 0) + |y| \cdot (0, -1) = (-1, 0) + \cdots + (-1, 0) + (0, -1) + \cdots + (0, -1)$$

where we add $|x|$ copies of $(-1, 0)$ and $|y|$ copies of $(0, -1)$. This point is in S by the additive recursive step. To show that points not in the lower left quadrant are not in the set S, we will prove the following claim by structural induction.

Prove: For every $(a, b) \in S$, we have $a \leq 0$ and $b \leq 0$, but also $a < 0$ or $b < 0$.

2. The Leaves Don't Fall Far From The... `Tree` [20 points]

In this problem, we'll use a definition of trees that looks a little different from the one we saw in class.

Basis Step: $(\texttt{null}, \bullet, \texttt{null})$ is a `tree`.
Recursive Step: If L, R are `trees` then $(L, \bullet, R)$ is also a `tree`

We will also use the following recursively defined function for nodes:

$$\texttt{nodes}((\texttt{null}, \bullet, \texttt{null})) = 1$$
$$\texttt{nodes}((L, \bullet, R)) = 1 + \texttt{nodes}(L) + \texttt{nodes}(R) \qquad \text{for two arbitrary } \texttt{trees } L, R$$

Show that every tree has an odd number of nodes. In other words, show that for all trees t, $\texttt{nodes}(t)$ is odd.

3. Clawing for Glory [23 points]

You and your friend are staring at a holiday limited edition claw machine, and you both want the one and only Pikachu plushie in there. It is holiday season so the claw machine takes in both red and green coins.

After a series of observations, you realize the game is based on luck, not skill—the machine remembers how many red and green coins it has received since someone last won a toy. You notice that every time someone captures a plushie successfully, the claw machine refreshes to its initial state (r, g), with r representing the number of red coins it needs and g for the number of green coins it needs. On the play when this state reaches $(0, 0)$, the player wins a plushie. In other words, the claw machine is set to letting one plushie out after it has received r red coins and g green coins (having more than the required amount for a color of coins while the other color's requirement is unmet would not have a plushie come out, and these surplus coins do not get carried over after the machine reboots).

The claw machine is currently in the state (k, k) for some k (i.e., it needs the same number of coins of each type). You and your friend take turns to play. In their turn, each player can either put in red coins or green coins (i.e. only one color of coins each round, no mix and matching). The player gets to pick how many coins of a single color they put in (but the state of the machine has to go down by at least 1). More formally, assuming you are at state (a, b) you can move to any of the states: $(a, b - i)$ where $1 \leq i \leq b$ or $(a - j, b)$ where $1 \leq j \leq a$

You need to be the one to put in enough coins of a single color to make the claw machine's state go down to (0, 0) so you can get your favorite Pikachu plushie!
You generously allow your friend to go first.

(a) Using induction, prove that in any claw machine where the default initial state is (a, a) for $a > 0$ you (the player that goes second) can always win the game and get the plushie.

Be sure to explicitly and clearly define a predicate $P()$! We **strongly** recommend your predicate includes the phrase "It is not my turn" or "the second player can" or something similar. The predicate you define should only take in one input. [20 points]

(b) Describe your winning strategy (i.e. describe how you should put in coins of a certain color in order to win the plushie, assuming that you go **SECOND**). A strategy would be something like "If my friend puts in i green coins then I will..." [3 points]

4. Bijections? [12 points]

Determine if the following functions are (1) one-to-one and (2) onto. For each claim: provide a proof if true, otherwise give a counterexample (you must also explain why the counterexample works).

(a) $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R},\ f(x, y) = x + y$. [6 points]

(b) $g : \mathbb{Z} \to \{-1, 1\} \times \mathbb{N},\ g(x) = (\text{sgn}(x), |x|)$. [6 points]
Here, we define the function:

$$\text{sgn}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases}$$

5. Stringing Things Together [12 points]

For each of the following, write a recursive definition of the set of strings satisfying the given properties. Briefly justify (2-4 sentences per part) that your solution is correct. You do not need to mention the exclusion rule.
All problems have relatively simple solutions (e.g., at most 4 basis steps and recursive steps). We may deduct for solutions which are not simple (but you do **not** need the simplest solution).

(a) Binary strings with an odd number of 0s.

(b) Binary strings where every 0 is followed by an even number of 1s.

(c) Binary strings in the form $x0y$ where x, y are binary strings and y is the reversal of x.

6. Constructing Regular Expressions (Online) [15 points]

For each of the following languages, construct a regular expression that matches exactly the given set of strings.

You will submit (and check!) your answers online at https://grin.cs.washington.edu/. Think carefully before entering a submission; you only have 10 guesses. Because these are auto-graded, we will not award partial credit.

Grin may not be available immediately when the homework is released; it will be ready by Monday Nov. 18.

(a) Binary Strings that begin with a 0 and have an odd length.

(b) Binary strings with at least one 0 or at most two 1's.

(c) Binary strings that start with 0, end with 1 do not contain 00.

7. Feedback [1 point]

Please keep track of how much time you spend on this homework and answer the following questions. This can help us calibrate future assignments and future iterations of the course, and can help you identify which areas are most challenging for you.

- How many hours did you spend working on this assignment?
- Which problem did you spend the most time on?
- Any other feedback for us?