Homework 7: Structural Induction, Regexes

Version 2: Posted 11/17, 9:30 PM. We corrected the hint in problem 1 (to apply only to members of the set S, not all integers).

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 induction for problems 1 and 2.

1. Finding Pie [20 points]

Garfield is trying to find Jon's secret safe full of pies! Jon moves the safe often, so Garfield has been tracking Jon's movements in order to predict where the safe will move next. From the existing data Garfield has compiled, he has formed a recursively defined set of points that Jon may arrive at. Perhaps by proving some facts about this set, Garfield may begin to find the hidden pie safe...

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

Basis Step: $(3, 3) \in S$ and $(3, 9) \in S$
Recursive Step: if $(a, b) \in S$ and $(c, d) \in S$, then $(a, 2b) \in S$ and $(a + c, d) \in S$.

Prove: For every $(a, b) \in S$, we have $3|ab$.

Hint: You may use the following fact **AFTER** you have applied your inductive hypothesis (during your inductive step): For $x, y \in S$: if $3|xy$ then $3|x$ and $3|y$.

2. Vector's Berries [23 points]

You and your friend Vector Algebra (yes, that is their name) decided to get a change of scenery and go to a field to pick some berries (if you're in CSE2 right now, take a break and go pick some berries). You each bring one basket, and a dream: to pick every strawberry and blueberry in the area!

However, this is no ordinary field: it's actually a highly experimental secret vector field, and the berries aren't actually berries... they're unit vectors! The strawberries correspond to the unit vector in the x direction $(1, 0)$, while the blueberries correspond to the unit vector in the y direction $(0, 1)$. Being math enthusiasts, you and your friend were delighted: there's no greater joy than mathematics! Right?

But as if math wasn't exciting enough, you and your friend decided to up the stakes: with a game! You will each take turns. At the start of a turn, you select a type of berry. During your turn, you may take as many berries as you like; but you can only take berries of **the type of berry that you selected**. The person to take the last berry wins **eternal glory** and gets to skip linear algebra (since they clearly understand vector fields so well)! In linear algebra terms, you can take any scalar multiple of your chosen unit vector...

With that in mind, you and your friend decided that it would make sense to represent your current picks as an ordered pair (s, b), where s is the number of strawberries left in the field and b is the number of blueberries left in the field. In other words, we can represent the state of the entire field with that ordered pair and take a look at it at a glance. So when the state of the field reaches $(0, 0)$, the person who was picking berries wins the linear algebra credit.

Formally, imagine that the field is currently in the state (k, k) for some k (i.e., it needs the same number of berries of each type), and you and your friend are taking turns picking. In their turn, each player can only take one type of berry, but can take as many as they want (i.e. only one type of berry each round, no mixing and matching). The player gets to pick how many berries of a certain type they take, and they **must** take at least one per turn. More formally, assuming you are at state (s, b) you can move to any of the states: $(s, b - i)$ where $1 \le i \le b$ or $(s - j, b)$ where $1 \le j \le s$

You need to be the one to take the last berry to take the field to (0, 0)! Then you can skip an entire class!!!!!! What a dream come true. You kindly allow Vector Algebra (yes, that is their name, again) to go first.

(a) Using induction, prove that in any vector field where the default initial state is (a, a) for $a > 0$ you (the player that goes second) can always win the game and skip linear algebra.

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 pick berries of a certain color in order to win, assuming that you go **SECOND**). A strategy would be something like "If my friend picks i strawberries then I will..." [3 points]

3. 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) $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R},\ f((x, y)) = x^2 - y$. [6 points]

(b) $g : \mathbb{Z} \to \{0, 1, 2\} \times \mathbb{N},\ g(x) = (x\%3, x^2)$. [6 points]
Where $\%$ is the mod operator. For the problem, the result should always be nonnegative (e.g., $-11\%12 = 1$ rather than -11)

4. Stringing Things Together [12 points]

For each of the following, write a recursive definition of the set of strings satisfying the given properties (you do not need to mention the exclusion rule). Briefly justify (2-4 sentences per part) that your solution is correct.
All problems have relatively simple solutions (e.g., at most 6 basis steps and recursive steps). We may deduct for solutions which are very not simple (but you do **not** need the simplest solution).

(a) String (containing only digits 0, 1, 2) with an even number of 2s

(b) Binary strings with no consecutive 1s.

(c) Binary strings with an equal number of 0s and 1s.

5. 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.

(a) Binary Strings that have even length and end with 1.

(b) Binary strings with less than three 0's or at least one 1.

(c) Binary strings that do not contain 1001.
Hint: Look at the last example on lecture 22, slide 13, and see if you can adapt it. You'll need to think carefully about what kind of 'building blocks' you are allowed while still avoiding 1001. Don't forget to check edge cases (like very short strings).

6. 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?