# Homework 6: Strong Induction, Structural Induction, Regular Expressions, CFGs

Due date: Wednesday March 3 at 11:59 PM (Seattle time, i.e. GMT-8)

If you work with others (and you should!), remember to follow the collaboration policy.
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.

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

## 1. Manhattan Walk [20 points]

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

**Basis Step:** $(0, 0) \in S$
**Recursive Step:** if $(a, b) \in S$ then $(a, b + 1) \in S$, $(a + 2, b + 1) \in S$

Prove that $\forall (a, b) \in S, a \leq 2b$.

**Hint** Remember that with structural induction you must show $P(s)$ for every element $s$ that is added by the recursive rule – you will need to show $P()$ holds for two different elements in your inductive step.

## 2. Running Times [20 points]

You wrote a piece of recursive code. On an input of size $n$, your function takes $T(n)$ time to run, where:

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

In the definition above, $\lfloor x \rfloor$ is the "floor" function, it returns the greatest integer at most $x$.
For example: $\lfloor 3.2 \rfloor = 3$, $\lfloor 3.7 \rfloor = 3$, $\lfloor 3 \rfloor = 3$.

Show that for all $n \in \mathbb{N}$ with $n \geq 1$, $T(n) \leq 20n$

Hint 1: Notice that while $T()$ is defined with equality, you are only proving an inequality.
Hint 2: The only fact about the floor function you will need is $\lfloor x \rfloor$ is an integer and $x - 1 < \lfloor x \rfloor \leq x$.

## 3. str000ng induction would be a good choice [20 points]

Let $0^n$ mean a string of $n$ zeros. Let $S$ be the set of strings defined as follows:

**Basis Steps:** $0^3 \in S$, $0^5 \in S$
**Recursive Step:** If $0^x, 0^y \in S$ then $0^x \cdot 0^y \in S$ where $\cdot$ is string concatenation.

Show that, for every integer $n \geq 12$ the set $S$ contains the string $0^n$.

**Caution:** Structural Induction is not the best tool for this problem. Structural induction shows $\forall_{x \in S}(P(x))$. You're analyzing what the elements of $S$ are in this problem, not proving a predicate holds for all elements of $S$.

## 4. The Apple Doesn't Fall Far From The... `Tree` [20 points]

In CSE 143, you saw a recursive definition of trees. That definition looks a little different from what we saw in class.

The following definition is analogous to what you saw in 143. We'll call them `JavaTrees`.

**Basis Step:** `null` is a `JavaTree`.
**Recursive Step:** If $L, R$ are `JavaTrees` then $(\texttt{data}, L, R)$ is also a `JavaTree`.

Show that for all `JavaTrees`: if they have $k$ copies of `data` then they have $k+1$ copies of `null`.

Remark: You're effectively showing here that a binary tree with $k$ nodes has $k+1$ `null` child pointers.

## 5. Constructing Regular Expressions (Online) [20 points]

For each of the following languages, construct a regular expression that matches exactly the given set of strings. You should submit (and check!) your answers online at https://grin.cs.washington.edu/

Think carefully before entering your regular expression; you only have 5 guesses. Because these are auto-graded, we will not award partial credit.

You **must** also take a screenshot of your final submission and include that in your gradescope submission.

(a) Binary strings where every occurance of a 1 is immediately followed by a 0.

(b) Binary strings where no occurence of 00 is immediately followed by a 1.

(c) The set of all binary strings that contain at least one 1 and at most two 0's.

(d) The set of all binary strings that begin with a 1 and have length congruent to $2 \pmod 4$.

## 6. Context Is Everything. Except for Context-Free Grammars [15 points]

For each of the following languages, construct a context-free grammar that generate the given set of strings. Make sure to tell us which nonterminal is the start symbol. If your grammar has more than one nonterminal, write a sentence describing what sets of strings you expect each variable in your grammar to generate.

For example, if your grammar were:

$$\begin{aligned} S &\to E|O \\ E &\to EE|CC \\ O &\to EC \\ C &\to 0|1 \end{aligned}$$

We would expect you to say something like "$E$ generates non-empty even length binary strings; $O$ generates odd length binary strings; $C$ generates binary strings of length one." You do not need to write a description for the start symbol (as that will be described by the original problem statement).

(a) The set of all binary strings that contain at least one 1 and and most two 0's.

(b) $\{1^m 0^n 1^{m+n} : m, n \geq 0\}$

(c) Binary strings with an odd number of 0's

## 7. Feedback

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?
- Did you have any difficulty using Grin?
- Any other feedback for us?
- Reminder to please list your collaborators!