Homework 7: Structural Induction, Regular Expressions, CFGs

Due date: Friday December 4 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. What doesn't kill you makes you stronger [25 points]

Consider the following intertwined definitions of recursively defined sets, S, T.

Definition of S
Basis: $0 \in S$
Recursive: If $x \in T$ then $x + 3 \in S$

Definition of T
Basis: $1 \in T$, $3 \in T$
Recursive: If $x \in S$ then $x + 1 \in T$.

Let $P(n)$ be "if n is odd, then $n \in T$" and let $Q(n)$ be "if n is even, then $n \in S$"

(a) Explain why $P(2)$ is true (don't write a formal proof; our answer is one sentence). [2 points]

(b) Imagine you wanted to prove $P(n)$ by induction. Why is even a strong inductive hypothesis not going to be enough to (easily) complete an inductive step? (1-2 sentences) [3 points]

(c) Define $R(n) := P(n) \wedge Q(n)$. Show $R(n)$ holds for all natural numbers $n \geq 4$ by induction on n. [20 points]
Be careful with this proof: you should not use structural induction (you are showing a claim about all integers, not a claim about every member of S or T) Also think carefully about what your inductive step should look like – P and Q are **implications**, that means your inductive hypothesis will be an implication and your inductive step will begin by supposing a hypothesis.

(d) This proof technique is sometimes called "strengthened induction" (not to be confused with strong vs. weak induction). You prove a strengthened claim $P(n) \wedge Q(n)$ for all n in order to show $P(n)$ for all n, becuase $P()$ wasn't nice enough to induct on. Choosing what to add is a tradeoff – the more information in $Q()$ that you add, the more information you have in your inductive hypothesis to assume. But also the more you have to show in your inductive step! You do not have to write anything for this part [0 points]

3. Recursion – See: Recursion [18 points]

For each of the following languages, give a recursive description of the language. Your basis step must explicitly enumerate a finite number of initial elements. Use the shortest description possible (in terms of the number of recursive rules and the number of basis rules). Briefly (1-2 sentences) justify that your description defines the same language. Do not give us a full proof; you do not have to justify why your description is the shortest possible.

(a) Binary strings that start with 0 and have odd length (i.e. an odd number of characters).

(b) Binary strings x such that $\texttt{len}(x) \equiv 1 \pmod 3$ where $\texttt{len}(x)$ is the number of characters in x.

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

4. 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://grinch.cs.washington.edu/cse311/regex

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

Do not include whitespace (spaces, tabs, or other blanks) as the system considers them characters.

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 $2 \pmod 4$.

5. 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:

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

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 label 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