CSE 311: Foundations of Computing I

Homework 7 (due Wednesday, Nov 23 at 6:00 PM)

Directions: *Write up carefully argued solutions to the following problems. The first task is to be complete and correct. The more subtle task is to keep it simple and succinct. Your solution should be clear enough that it should explain to someone who does not already understand the answer why it works. You may use any results proven in lecture without proof. Anything else must be argued rigorously. Unless otherwise specified, all answers are expected to be given in closed form.*

1. Strings are Strings [Online] (20 points)

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

(a) [5 Points] The set of all binary strings that end with 0 and have even length, or start with 1 and have odd length.

(b) [5 Points] Binary strings where the number of 0s is congruent to 2 modulo 3.

(c) [5 Points] The set of all binary strings that contain at least two 1's and at most two 0's.

(d) [5 Points] The set of all binary strings that don't contain 001.

You must submit and check your answers to this question using `https://grinch.cs.washington.edu/cse311/regex`. Each part has a (possibly different) limit on the number of submissions. Your last submission prior to this threshold will be graded.

2. Constructing Four Grammars [Online] (30 points)

For each of the following, construct context-free grammars that generate the given set of strings. If your grammar has more than one variable, we will ask you to write a sentence describing what sets of strings you expect each variable in your grammar to generate.

For example, if your grammar were:

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

We would expect you to say "E generates (non-empty) even length binary strings; O generates odd length binary strings; C generates binary strings of length one."

(a) [5 Points] The set of all binary strings that are of odd length and have 1 as their middle character.

(b) [5 Points] $\{1^m 0^n 1^{m+n} : m, n \geq 0\}$

(c) [10 Points] All binary strings that contain at least two 0's and at most two 1's.

(d) [10 Points] $\{1^m 0^n 1^p : m, n, p \geq 0$ and $p \equiv m + n \pmod 2\}$

You must submit and check your answers to this question using
`https://grinch.cs.washington.edu/cse311/cfg`.
Each part has a (possibly different) limit on the number of submissions. Your last submission prior to this threshold will be graded. You must **also** upload your explanation **and** the grammar to Canvas.

3. Relations warmup (12 points)

Consider the set of all twitter users. In each part of this problem we define a relation R on this set. Determine whether R is reflexive, symmetric, antisymmetric, and/or transitive:

(a) [4 Points] $(a, b) \in R$ if there are no common followers of a and b.

(b) [4 Points] $(a, b) \in R$ if there is a user c who is a follower of both a and b.

(c) [4 Points] $(a, b) \in R$ if every user who is a follower of a is also a follower of b.

4. Set Up To Relate (18 points)

Let A be a set. Let R and S be transitive relations on A.

(a) [9 Points] Is $R \cup S$ necessarily transitive? Prove your answer.

(b) [9 Points] Is $R \cap S$ necessarily transitive? Prove your answer.

5. Symmetry and Power (20 points)

Let R be a symmetric relation on a set A. Use induction to show that R^n is symmetric for all integers $n \geq 1$.

6. EXTRA CREDIT: Ambiguity (-NoValue- points)

Consider the following context-free grammar.

$$
\begin{aligned}
\langle \textbf{Stmt} \rangle &\to \langle \textbf{Assign} \rangle \mid \langle \textbf{IfThen} \rangle \mid \langle \textbf{IfThenElse} \rangle \mid \langle \textbf{BeginEnd} \rangle \\
\langle \textbf{IfThen} \rangle &\to \text{if condition then } \langle \textbf{Stmt} \rangle \\
\langle \textbf{IfThenElse} \rangle &\to \text{if condition then } \langle \textbf{Stmt} \rangle \text{ else } \langle \textbf{Stmt} \rangle \\
\langle \textbf{BeginEnd} \rangle &\to \text{begin } \langle \textbf{StmtList} \rangle \text{ end} \\
\langle \textbf{StmtList} \rangle &\to \langle \textbf{StmtList} \rangle \langle \textbf{Stmt} \rangle \mid \langle \textbf{Stmt} \rangle \\
\langle \textbf{Assign} \rangle &\to \texttt{a := 1}
\end{aligned}
$$

This is a natural-looking grammar for part of a programming language, but unfortunately the grammar is "ambiguous" in the sense that it can be parsed in different ways (that have different meanings).

(a) [-NoValue- Points] Show an example of a string in the language that has two different parse trees.

(b) [-NoValue- Points] Give a new grammar for the same language that is **unambiguous** in the sense that every string has a unique parse tree.