CSE 311: Foundations of Computing I

Homework 7 (due Wednesday, May 23 at 11:59 PM)

Directions: *Write up carefully argued solutions to the following problems. 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 results from lecture, the theorems handout, and previous homeworks without proof.*

1. Constructing Four Grammars (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:

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

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

(b) [10 Points] $\{1^m 0^n 1^{m+n} : m, n \geq 0\}$

(c) [10 Points] All strings of the form $x\#y$, with $x, y \in \{0,1\}^*$, such that either x is a subsequence of y^R (i.e., it is y^R with some characters possibly removed) or y is a subsequence of x^R.

2. Set Up To Relate (15 points)

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

(a) [7 Points] Is $R \cup S$ necessarily transitive? Prove your answer.

(b) [8 Points] Is $R \cap S$ necessarily transitive? Prove your answer.

3. Symmetry and Power (25 points)

(a) [5 Points] Suppose that R and S are symmetric binary relations on a set A. Show that $S \circ R$ is not necessarily symmetric.

(b) [10 Points] Let R and S be as above. Suppose we also know that $S \circ R = R \circ S$. Prove that $S \circ R$ is symmetric.

(c) [10 Points] Use induction to show that R^n is symmetric for all integers $n \geq 1$. You can use without proof the fact that relation composition ($\circ$) is associative. (In other words, R^n is the same relation no matter where you place parentheses in $R \circ R \circ \cdots \circ R$.)

4. DFAs [Online] (30 points)

For each of the following, create a *DFA* that recognizes exactly the language given.

(a) [10 Points] The set of all binary strings that contain at least two 1's or at most two 0's.

(b) [10 Points] Binary strings where if we treat them as a binary number, that number is congruent to 2 modulo 5. For example, 00111 is in the language (because $7 \equiv 2 \bmod 5$) but 011 is not.

(c) [10 Points] Consider a binary string w of even length. Let odd(w) be characters at odd position in w and even(w) be characters at even position in w. For example, if $w = 100110$ then odd$(w) = 101$ and even$(w) = 010$. Consider even(w) and odd(w) as binary numbers. Let L be the language

$$L = \{w \in \{0,1\}^* : w \text{ has even length and odd}(w) > \text{even}(w)\}.$$

For example, $00101101 \in L$ because $0110 > 0011$ but $001101 \notin L$ because $010 \not> 011$. Also, $110011 \notin L$.

> You must submit and check your answers to this question using `https://grinch.cs.washington.edu/cse311/fsm`.
> You also must submit documentation in Canvas. For each problem, include a screenshot of your submitted DFA along with, for each state s of your machine, a description of which strings will take the DFA from the start state to s.

5. EXTRA CREDIT: Ambiguity (0 points)

Consider the following context-free grammar.

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

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 distinct meanings).

(a) [0 Points] Show an example of a string in the language that has two different parse trees that are meaningfully different (i.e., they represent programs that would behave differently when executed).

(b) [0 Points] Give **two different grammars** for this language that are both unambiguous but produce different parse trees from each other.