CSE 311: Foundations of Computing I

Homework 7 (due December 2nd at 11:00 PM)

Directions: *Write up carefully argued solutions to the following problems. Each solution should be clear enough that it can explain (to someone who does not already understand the answer) why it works. However, you may use results from lecture, the reference sheets, and previous homeworks without proof.*

1. Glitz and Grammar [Online] (15 points)

For each of the following, a construct context-free grammar that generates 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 was:

S → **E** | **O**
O → **EC**
E → **EE** | **CC**
C → 0 | 1

You could say "C generates binary strings of length one, E generates (non-empty) even length binary strings, and O generates odd length binary strings." It is also fine to use a regular expression, rather than English, to describe the strings generated by a variable (assuming such a regular expression exists).

(a) [5 Points] Binary strings matching the regular expression "$1(0 \cup 111)^* \cup 000$".
Hint: You can use the technique described in lecture to convert this RE to a CFG.

(b) [5 Points] All strings of the form $x\#y$, with $x, y \in \{0,1\}^*$ and x a subsequence of y^R.
(Here y^R means the reverse of y. Also, a string w is a subsequence of another string z if you can delete some characters from z to arrive at w.)

(c) [5 Points] All binary strings in the set $\{0^m 1^n 0^{2n+m} : m, n \geq 0\}$.

> Submit and check your grammars here:
>
> https://grin.cs.washington.edu/
>
> Think carefully about your answer to make sure it is correct before submitting. You have only 5 chances to submit a correct grammar.
>
> **Note**: You must also include each grammar and sentences describing each new non-terminal, as described above, with the rest of your assignment.

2. 101 Relations (12 points)

For each of the relations below, determine whether or not it has each of the properties of reflexivity, symmetry, antisymmetry, and/or transitivity. If a relation has a property, simply say so without any further explanation. If a relation does not have a property, state a counterexample, but do not explain your counterexample further.

(a) [2 Points] Define $R \subseteq \mathbb{Z} \times \mathbb{Z}$ by $(a, b) \in R$ iff $a^2 = b^2$.

(b) [2 Points] Define $S \subseteq \mathbb{Z} \times \mathbb{Z}$ by $(a, b) \in S$ iff $|a - b| \leq 3$ (where $|x|$ denotes the absolute value of x, i.e., x if x is nonnegative, and $-x$ if x is negative).

(c) [2 Points] Define $T \subseteq \mathbb{Z} \times \mathbb{Z}$ by $(a, b) \in T$ iff $a \neq b$.

(d) [2 Points] Let $A = \{n \in \mathbb{N} : n > 0\}$ be the set of positive natural numbers. Define $U \subseteq A \times A$ by $(a, b) \in U$ iff $a \mid b$, i.e., a divides b.

(e) [2 Points] Let $B = \mathcal{P}(\mathbb{Z})$. Define $V \subseteq B \times B$ by $(X, Y) \in V$ iff $X \cap [10] \subseteq Y \cap [10]$. (Remember that $[n] = \{1, \ldots, n\}$.)

(f) [2 Points] Let $A = \{n \in \mathbb{N} : n > 0\}$ be the set of positive natural numbers. Define $W \subseteq A \times A$ by $(a, b) \in W$ iff $a \mid 2$ and $b \mid 3$ and ($a = 1$ or $b = 1$).

Hint: Be careful! The definition of W says $a \mid 2$, which is different from the (more common) $2 \mid a$.

3. Get a Prove On (16 points)

Let R and S be relations on a set A. Consider the following claim:

Given that R and S are transitive, it follows that $R \cap S$ is transitive.

(a) [1 Point] What does "$R \cap S$ is transitive" become if we unroll the definition of transitive?

(b) [12 Points] Write a **formal proof** that the claim from (a) holds.

Your two givens should be the claims you get by unrolling the definition of transitive in the statements "R is transitive" and "S is transitive".

Note: For this assignment, we will allow you to introduce multiple quantified variables in a single application of Intro $\forall$. For example, a subproof of the form "Let a and b be arbitrary... $P(a, b)$" would produce $\forall a \, \forall b \, P(a, b)$ in a single application of Intro $\forall$.

(c) [3 Points] Translate the formal proof into an English proof of the original claim.

4. Cat Goes "Meow". Dog Goes "Proof" (16 points)

Let R and S be relations on a set A. Consider the following claim:

Given that R and S are symmetric and $R \circ S = S \circ R$, it follows that $R \circ S$ is symmetric.

(a) [1 Point] What does "$R \circ S$ is symmetric" become if we unroll the definition of symmetric?

(b) [1 Point] What does "$R \circ S = S \circ R$" become if we unroll the definition of "$=$" on sets?

(c) [11 Points] Write a **formal proof** that the claim from (a) holds.

Your three givens should be the claims you get by unrolling the definition of symmetric in the statements "R is symmetric" and "S is symmetric", along with your answer for (b).

Hint: You will, of course, need to use the definition of composition ("$\circ$") in your proof.

Hint: Mechanically applying the givens will let you infer, from $(x, y) \in R \circ S$, that $(y, x) \in S \circ R$ (the composition in the reversed order). You will need that third given to then show that $(y, x) \in R \circ S$.

(d) [3 Points] Translate the formal proof into an English proof of the original claim.

5. Few and Far Machine [Online] (25 points)

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

(a) [5 Points] Binary strings where every occurrence of a 0 is immediately followed by a 1.

(b) [5 Points] Binary strings that start with 0 and have even length.

(c) [5 Points] Binary strings with an even number of 0s.

(d) [5 Points] Binary strings with at least two 0s.

(e) [5 Points] Binary strings with at least two 0s **or** at least two 1s.

Submit and check your answers to this question here:

`https://grin.cs.washington.edu/`

Think carefully about your answer to make sure it is correct before submitting.
You have only 5 chances to submit a correct answer.

6. Ruled With an Iron List (16 points)

Recall the definition of **List** and the functions sum and concat from Homework 6.

Let us now define a new function rev recursively as follows:

$$\begin{aligned} \text{rev}(\texttt{nil}) &::= \texttt{nil} \\ \text{rev}(a :: L) &::= \text{concat}(\text{rev}(L), a :: \texttt{nil}) \quad \forall a \in \mathbb{Z}, \forall L \in \textbf{List} \end{aligned}$$

(a) [4 Points] Write a chain of equalities, citing the appropriate definitions, showing that

$$\text{rev}(1 :: 2 :: 3 :: \texttt{nil}) = 3 :: 2 :: 1 :: \texttt{nil}$$

(b) [12 Points] Use structural induction to prove the following

$$\forall L \in \textbf{List}\, (\text{sum}(\text{rev}(L)) = \text{sum}(L))$$

You can (and should) cite what we proved in Problem 3(c) of Homework 6 in your proof of this claim.

7. Extra Credit: With a Grammar, the Whole World is a Nail (0 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 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.