Context-Free Grammars

1. CFG Design

a. (from Sipser Problem 2.46) Consider the CFG $C = (V, \Sigma, R, S)$ where $V = \{A, B\}$, $\Sigma = \{0, 1\}$, $S = A$, and R is given by

$$A \to AA \mid B$$
$$B \to 0B1 \mid 01$$

Describe the language generated by C.

b. (Sipser Problem 2.47b) Let $\Sigma = \{0, 1\}$, and let B be the collection of strings that contain at least one 1 in their second half. In other words, $B = \{uv \mid u \in \Sigma^*, v \in \Sigma^* 1 \Sigma^*, \text{ and } |u| \geq |v|\}$. Give a CFG that generates B.

2. Ambiguous Grammars

In this section we will be developing the concept of grammar "ambiguity". Please do *not* refer to the Ambiguity section of the textbook.

a. Consider the CFG C_1 given by $E \to E + E \mid E * E \mid 5$. How many different derivations are there for the string 5+5*5? If this grammar were being used in a real-life application (e.g. in the code of a calculator), why might these multiple derivations be a problem?

b. We want to call a CFG *ambiguous* if it exhibits the problem from part a. Come up with a formal definition of ambiguity. *(You may rely on the formal definitions of "derivation" and/or "parse tree".)*

c. Consider the CFG C_2 given by

$$S \to VN$$
$$V \to eat \mid throw$$
$$N \to apple \mid snowball$$

How many derivations are there for "throw apple"? Why would these multiple derivations *not* pose the same practical problems as the ambiguity we saw in part a? Revise your definition of ambiguity from part b if necessary to make sure that it considers C_1 to be ambiguous but does *not* consider C_2 to be ambiguous.

d. Design an *unambiguous* CFG C_3 such that $L(C_3) = L(C_1)$. *(It just has to generate the same strings; you don't have to worry about whether the parse trees correspond to the correct mathematical order of operations.)*

e. Call a context-free language *inherently ambiguous* if *every* CFG that generates it is ambiguous. Find an inherently ambiguous language, and justify why you think it's inherently ambiguous (though actually proving it is beyond the scope of this class).