#1 The Basics

Answer each part for the following context-free grammar $G = (V, \Sigma, R, S)$ with start symbol S.

$$
\begin{aligned}
S &\to XSX \mid U \\
U &\to aTb \mid bTa \\
T &\to XTX \mid X \mid \epsilon \\
X &\to a \mid b
\end{aligned}
$$

1. What are the elements of V (the variables of G)?
2. What are the elements of Σ (the set of terminals in G)?
3. Give three strings in $L(G)$.
4. Give three strings in Σ^* that are *not* in $L(G)$.
5. Consider the string XXX. Show via a derivation that $XXX \overset{*}{\Rightarrow} aba$.
6. Consider the string T. Show via a derivation that $T \overset{*}{\Rightarrow} aba$.
7. Consider the string T. Show via a derivation that $T \overset{*}{\Rightarrow} XX$.
8. Consider the string R. Show via a derivation that $R \overset{*}{\Rightarrow} aabbb$.
9. Give a description in English of $L(G)$.

#2 Conditional Ambiguity

In C-like programming languages, conditional statements frequently have the following form:

```
STMT         -> ASSIGN | IF-THEN | IF-THEN-ELSE
IF-THEN      -> if condition then STMT
IF-THEN-ELSE -> if condition then STMT else STMT
ASSIGN       -> a:=1
```

This grammar has $V =$ { `STMT, IF-THEN, IF-THEN-ELSE, ASSIGN` } and

$\Sigma =$ { `if, condition, then, else, a:= 1` }.

(The 'skip' statement is a statement that does nothing. For our purposes, it acts as a "base case" to the *stmt* procedure. You could substitute any "real" base statement such as a call to a function, e.g., `printf()` instead.)

1. Give a string generated by this grammar that has two valid parse trees. In other words, demonstrate the ambiguity of this grammar.
2. Which of the two parse trees is the "preferred" interpretation?
3. Change the grammar above to remove the ambiguity. You may try to preserve the original syntax of the conditional statement, but you may also add additional syntax to remove the ambiguity.