PLC: Micro Assignment 4 [12.5 points]

Due Thursday, Feb. 26, within first 3 minutes of class. You can volunteer to present one of your answers for 1 extra credit point. Preference will be given to students who have not presented yet.

1. This question concerns the following grammar, with start symbol *eqs* and nonterminals *eqs*, *eq*, *lhs*, and *rhs*. The symbols like *EqsNext* to the left of the colons are the names of the productions.

$$
\begin{array}{llll}
EqsNext & : & eqs & \to & eq\ eqs \\
EqsStart & : & eqs & \to & eq \\
Eq & : & eq & \to & lhs\ \text{'='}\ rhs\ \text{'\textbackslash n'} \\
Lhs & : & lhs & \to & (\text{'a'}\ |\ \text{'b'})+ \\
Rhs & : & rhs & \to & (\text{'a'}\ |\ \text{'b'})+ \\
\end{array}
$$

(a) Is this grammar regular or context-free? [1 point]

(b) Draw a step-by-step derivation (do not combine steps) of the following input string, where you should assume each equation is followed by a newline ('\n'). You can just expand *lhs* (similarly *rhs*) to a string of 1 or more a's and b's in one step. [5 points]

```
aa = bb
ba = ab
```

(c) Draw the derivation tree corresponding to the derivation you wrote above. You can just write the strings of a's and b's for *lhs* and *rhs* directly below them in the tree [4 points].

(d) Draw the parse tree corresponding to the derivation tree you wrote above. To get the parse tree, replace each node you expanded with the name of the rule used to expand it. Also, drop sequences of terminals that can be reconstructed based on what rule was applied (e.g., '=' and '\n' for a *Eq* node) [2 points].

(e) Is this grammar ambiguous? [0.5 points]