Problem 1. Consider the following grammar for boolean expressions:

\[
\begin{align*}
e & \rightarrow b \mid v \mid e \land e \mid e \lor e \mid \neg e \\
b & \rightarrow \text{true} \mid \text{false} \\
v & \rightarrow x \mid y \mid z
\end{align*}
\]

Without making any assumptions about the precedence of the operations, please answer the following about each of the expressions below:

- Is the expression a part of the language defined by the grammar?
- If yes, please draw all possible parse trees for it in the grammar.

The expressions are as follows:
1. \(x \land \text{false}\)
2. \(x \lor y \land z\)
3. \(x \lor x \land y\)
4. \(x \lor \neg \text{true}\)

Is the grammar ambiguous? Please justify your answer.

Problem 2. Which of the following terms are \(\alpha\)-equivalent (i.e., equivalent up to renaming of the bound variables) to \(\lambda x.\lambda y.xyx\)?

1. \(\lambda y.\lambda x.yxx\)
2. \(\lambda x.\lambda x.xxx\)
3. \(\lambda x.\lambda y.yxx\)
4. \(\lambda y.\lambda x.yxx\)

Please explain your answer.

Problem 3. For each of the terms below show its step-by-step evaluation in the call-by-value or the call-by-name \(\lambda\)-calculus. Continue the evaluation until you either reach the normal form, or, if the term doesn't have a normal form, until you can demonstrate that the term diverges. If the evaluation is the same in the two calculi, then you only need to show one of them.

1. \((\lambda x.xxx)(\lambda y.2 + 3)\)
2. \((\lambda x.\lambda y.x)3\)
3. \((\lambda x.xxx)(\lambda x.xxx)\)
4. \((\lambda z. y)((\lambda x.xxx)(\lambda x.xxx))\).

**Problem 4.** Consider the call-by-name \(\lambda\)-calculus. Given a program
\[
(\lambda x.\lambda y.((\lambda z. y) z) x) M,
\]
where \(M\) is any term, can you replace it by \((\lambda x.\lambda y.y\times x) M\) without changing the program’s behavior? Why or why not?

**Problem 5.** Exercise 4.8 on p. 85.

**Problem 6.** Exercise 4.9 on p. 85, Part (a) only.