3. Rewrite the BNF of Example 3.4 to give + precedence over * and force + to be right associative.

4. Rewrite the BNF of Example 3.4 to add the ++ and −− unary operators of Java.

7. Using the grammar of Example 3.4, show a parse tree and a leftmost derivation for each of the following statements:
   a. \( A = (A + B) * C \)
   b. \( A = B + C + A \)
   c. \( A = A * (B + C) \)
   d. \( A = B * (C * (A + B)) \)

12. Consider the following grammar:

   \[
   \begin{align*}
   \langle S \rangle & \rightarrow a \langle S \rangle c \langle B \rangle | \langle A \rangle | b \\
   \langle A \rangle & \rightarrow c \langle A \rangle | c \\
   \langle B \rangle & \rightarrow d | \langle A \rangle
   \end{align*}
   \]

   Which of the following sentences are in the language generated by this grammar?
   a. abcd
   b. acccbd
   c. accbccc
   d. acd
   e. accc

24. Compute the weakest precondition for each of the following sequences of assignment statements and their postconditions
   a. \( a = 2 * b + 1; \)
      \( b = a - 3 \)
      \{b < 0\}
   b. \( a = 3 * (2 * b + a); \)
      \( b = 2 * a - 1 \)
      \{b > 5\}