Problem 1. Give an unambiguous grammar that generates the same language as

\[ S \rightarrow SS \mid (S) \mid 0 \]

Draw the parse tree for 000 in your grammar.

Problem 2. Write a BNF grammar that defines an integer number. Negative integer numbers are preceded by the minus sign. A non-zero number may not start with a zero. \( -0 \) is not a valid number.

Problem 3. Give the finite-state automata and the regular grammar for:

1. All strings over \( \{0,1\} \) containing the string 010.
2. All strings over \( \{0,1\} \) which do not contain the string 010.

Problem 4 Write unambiguous grammar that describes arithmetic expressions over one-letter variables (\( a, b, c \), etc.) which use *, /, +, binary -, and parentheses (do not include unary -, as in \( -x \)). Some valid expressions include: \( a, x+y, a+b+c, (m+n)*k, m+n*k \). The latter expression should be parsed according to the usual precedence rules (i.e., as the sum of the following: \( m \) and the product of \( n \) and \( k \)).

Draw parse trees for the following expressions:

1. \((x*y)+b/(a+c)\)
2. \(a+b*c-d\)

Problem 5. Construct the push-down automaton for the context-free grammar

\[ S \rightarrow 0S0 \mid 1S1 \mid 0 \mid 1 \]

Is it possible to describe this language by a regular grammar? Briefly explain your answer.