Context-free grammars

1. Write unambiguous grammars for the following languages:

   - The set of all strings of a's and b's that are palindromes.
   - Strings that match the pattern $a * b*$ and have more a's than b's.
   - Strings with balanced parenthesis and square braces. Example:

     `( [ [ ] (  ( ) [ ( ) ] [ ] ) ] )`

   - The set of all strings of a's and b's such that every a is immediatelly followed by at least one b.
   - The set of all strings of a's and b's with an equal number of a's and b's.
   - The set of all strings of a's and b's with an different number of a's and b's.
   - Blocks of statements in Pascal or MH, where the semicolons (`';'`) separate the statements:

     `( statement;  ( statement ;  statement  ) ; statement )`

   - Blocks of statemens in C, where the semicolons (`';'`) follow each statement:

     `{ statement;  { statement;  statement; }  statement; }`

2. Specify the previous grammars in ANTLR notation, modifying the grammar when necessary.

Parsing

1. Consider the following grammar:

$$S \rightarrow S\ S\ +\ \mid\ S\ S\ *\ \mid\ a$$

   and the string $aa + a*$.

   - Give a leftmost derivation for the string.
   - Give a rightmost derivation for the string.
   - Give a parse tree for the string.
   - Is the grammar ambiguous or unambiguous? Justify your answer.
   - Describe the language generated by this grammar.

2. Given the following grammar:

$$\begin{array}{rcl} S & \rightarrow & E\ \$ \\ E & \rightarrow & T + E \mid T \\ T & \rightarrow & \texttt{num} * T \mid \texttt{num} \end{array}$$

   design a table-driven top-down parser.

3. Given the following grammar:

$$\begin{array}{rcl} E & \rightarrow & T\ (\texttt{'+'}\ T) * \\ T & \rightarrow & F\ (\texttt{'*'}\ F) * \\ F & \rightarrow & \texttt{'('}\ E\ \texttt{')'} \mid \texttt{id} \end{array}$$

   design a recursive-descent parser (ANTLR style).

4. Consider the following grammar:

$$\begin{array}{rcl} G & \rightarrow & S\ \$ \\ S & \rightarrow & AM \\ M & \rightarrow & S \mid \varepsilon \\ A & \rightarrow & aE \mid bAA \\ E & \rightarrow & aB \mid bA \mid \varepsilon \\ B & \rightarrow & bE \mid aBB \end{array}$$

(a) Describe the language generated by the grammar.

(b) Give a parse treee for the string *abaa*\$.

(c) Is it an LL(1) grammar ? Build the parsing table and identify the conflicts.

5. Design a bottom-up parser and a top-down parser for the following grammar:

$$\begin{aligned} P &\rightarrow E\ \$ \\ E &\rightarrow \texttt{atom} \mid \texttt{'}\ E \mid \texttt{(}EE_s\texttt{)} \\ E_s &\rightarrow EE_s \mid \varepsilon \end{aligned}$$

Give the leftmost derivation and the parse tree for the string `(cdr '(a b c))$`.