Homework problems:

1. Convert the following grammar for certain type of list structures,
   \[ S \rightarrow (L) \mid a \]
   \[ L \rightarrow N \mid \varepsilon \]
   \[ N \rightarrow S, N \mid S \]
   into Chomsky normal form.

2. Determine, using the CYK algorithm (“dynamic programming method”, Lewis & Papadimitriou p. 155), whether the strings \textit{bbaab}, \textit{ababab} and \textit{aabba} are generated by the grammar
   \[ S \rightarrow AB \mid BA \mid a \mid b \]
   \[ A \rightarrow BA \mid a \]
   \[ B \rightarrow AB \mid b \]
   In the positive case, give also the respective parse tree(s).

3. Design pushdown automata recognising the following languages:
   (a) \(\{wcw^R \mid w \in \{a, b\}^*\}\);
   (b) \(\{ww^R \mid w \in \{a, b\}^*\}\).

Demonstration problems:

4. Design an algorithm for testing whether a given a context-free grammar \(G = (V, \Sigma, P, S)\), generates a nonempty language, i.e. whether any terminal string \(x \in \Sigma^*\) can be derived from the start symbol \(S\).

5. Design a pushdown automaton corresponding to the grammar \(G = (V, \Sigma, P, S)\), where
   \[ V = \{S, (, ), *, \cup, \emptyset, a, b\} \]
   \[ \Sigma = \{ (, ), *, \cup, \emptyset, a, b\} \]
   \[ P = \{S \rightarrow (SS), S \rightarrow S*, S \rightarrow (S \cup S), S \rightarrow \emptyset, S \rightarrow a, S \rightarrow b\} \]

6. Design a grammar corresponding to the pushdown automaton \(M = (Q, \Sigma, \Gamma, \Delta, s, F)\), where
   \[ Q = \{s, q, f\}, \quad \Sigma = \{a, b\}, \quad \Gamma = \{a, b, c\}, \quad F = \{f\}, \]
   \[ \Delta = \{((s, e, e), (q, c)), ((q, a, c), (q, ac)), ((q, a, a), (q, aa)), ((q, a, b), (q, b)), ((q, a, c), (q, bc)), ((q, b, b), (q, bb)), ((q, b, a), (q, e)), ((q, e, c), (f, e))\} \]