Homework problems:

1. Convert the following grammar for certain type of list structures,

\[
\begin{align*}
S & \rightarrow (L) \mid a \\
L & \rightarrow N \mid \varepsilon \\
N & \rightarrow S, N \mid S
\end{align*}
\]

into Chomsky normal form.

2. Determine, using the CYK algorithm (“dynamic programming method”, Sipser p. 241, Lewis & Papadimitriou p. 155), whether the strings \(abab\), \(aabb\) and \(bbaab\) are generated by the grammar

\[
\begin{align*}
S & \rightarrow AB \mid BA \\
A & \rightarrow BA \mid a \\
B & \rightarrow AB \mid b
\end{align*}
\]

In the positive cases, give also the respective parse trees.

3. Design pushdown automata recognising the following languages:

(a) \(\{ww^R \mid w \in \{a, b\}^*\}\);

(b) the language described by the grammar in Problem 2 of Tutorial 7.

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

\[
\begin{align*}
V &= \{S, (, ), *, \cup, \emptyset, a, b\} \\
\Sigma &= \{(, ), *, \cup, \emptyset, a, b\} \\
P &= \{S \rightarrow (SS), S \rightarrow S^*, S \rightarrow (S \cup S), \} \\
S & \rightarrow \emptyset, S \rightarrow a, S \rightarrow b
\end{align*}
\]

6. Design a grammar corresponding to the pushdown automaton \(M = (Q, \Sigma, \Gamma, \Delta, s, F)\), where

\[
\begin{align*}
Q &= \{s, q, f\}, \Sigma = \{a, b\}, \Gamma = \{a, b, c\}, F = \{f\}, \\
\Delta &= \{(s, e, c), (q, c), (q, ac), ((q, a), (q, ac)), ((q, a), (q, aa)), ((q, a), (q, c), (q, bc)), ((q, b), (q, b), (q, bb)), ((q, b), (q, c), (f, e)) \}
\end{align*}
\]