Homework problems:

1. Convert the following grammar into Chomsky normal form:

\[
\begin{align*}
S & \rightarrow AB \mid c \\
A & \rightarrow T \mid aA \\
B & \rightarrow TT \mid \varepsilon \\
T & \rightarrow bS
\end{align*}
\]

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

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

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

3. Design pushdown automata recognising the following languages:

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

\(\text{(b)} \ \{w \in \{a, b\}^* \mid w \text{ has as many } a\text{'s as } b\text{'s}\};\)

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 \(M = (Q, \Sigma, \Delta, s, F)\), where

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