1. [4+4+4 points] For each of the following regular languages over the alphabet \{0, 1\}, give according to your own choice either a regular expression or a deterministic or non-deterministic finite automaton.

   (a) The language \(A\) consists of all strings that end with 0101.
   (b) The language \(B\) consists of all strings that do not contain two consecutive zeros.
   (c) The language \(C\) consists of all strings where either the number of zeros or the number of ones (but not both) is odd.

2. [12 points] We recursively define the concept of a list as follows.

   - The empty list is a list, and we denote it by \texttt{NIL}.
   - If we are given \(n\) lists \(L_1, \ldots, L_n\) for some \(n \geq 1\), we can combine them to create a new list \((L_1, \ldots, L_n)\).
   - There are no lists except those implies by the previous two rules.

   Thus, we have for example the following lists:

   \[
   (((((\texttt{NIL}))))),
   ([\texttt{NIL}}, [\texttt{NIL}}, [\texttt{NIL}}, [\texttt{NIL}}, [\texttt{NIL}})),
   ([\texttt{NIL}}, [\texttt{NIL}}, [\texttt{NIL}}, [\texttt{NIL}}, [\texttt{NIL}}, [\texttt{NIL}})))).
   ([\texttt{NIL}}, [\texttt{NIL}}, [\texttt{NIL}}, [\texttt{NIL}}, [\texttt{NIL}}, [\texttt{NIL}})).
   \]

   We can consider lists as strings in the alphabet that contains the symbols "\texttt{NIL}"", "," (comma) and parentheses ",(" and ")". Give a context-free grammar for the language consisting of all such lists.

---

Continues on the reverse side!
3. [12 points] By simulating the CYK algorithm, determine whether the string bbaab is in the language generated by the grammar

\[
\begin{align*}
S & \rightarrow \quad X X \mid Y Y \\
X & \rightarrow \quad A B \mid B A \\
Y & \rightarrow \quad A X \mid X A \\
A & \rightarrow \quad A A \mid a \\
B & \rightarrow \quad B B \mid b
\end{align*}
\]

4. [12 points] Prove that if a language \(A\) and its complement \(\overline{A}\) are both Turing-recognizable, then \(A\) is decidable.

5. [4+8 points]

(a) Give short but precise definitions for the classes P and NP. You may assume various basic concepts related to Turing machines etc. as known.

(b) Give two examples of an NP complete problem. What are the implications of NP completeness for solving such problems in practice?