1. Rewrite the following SDT:

\[
A \rightarrow A \{a\} B | A B \{b\} | 0 \\
B \rightarrow B \{c\} A | B A \{d\} | 1
\]

so that the underlying grammar becomes non-left-recursive. Here, \(a, b, c,\) and \(d\) are actions, and \(0\) and \(1\) are terminals.

2. Consider the following post system rules for type checking logical expressions over variables:

\[
\begin{align*}
\rho(v) = \tau & \quad \rho \vdash e : \tau \\
\rho \vdash \#e : \int & \quad \rho \vdash e_1 : \tau \\
\rho \vdash \#e : \int & \quad \rho \vdash e_1 // e_2 : \tau
\end{align*}
\]

Given \(\rho = \{\langle s, \text{string}\rangle\}\) prove that \(\rho \vdash \#("" // s) : \int\) with the method explained in class (and lecture notes).

3. Consider the type declaration

\[
\text{struct BinTree} \\
\{ \\
\quad \text{int val;} \\
\quad \text{struct BinTree *left;} \\
\quad \text{struct BinTree *right;} \\
\};
\]

Draw a graph that shows how the C compiler represents this type internally.

4. Consider the following C code:

\[
\begin{align*}
\text{struct BinTree *} p, q; \\
\ldots \\
p->right->left = q.left;
\end{align*}
\]
Draw a type graph for \( p \) and \( q \) and indicate the types of \( p\rightarrow\text{right} \), \( p\rightarrow\text{right} \rightarrow \text{left} \), and \( q\leftarrow \text{left} \) that shows that the assignment is typed correctly (similar as shown on page 15 of slides for Ch6).

5. Consider the pseudo code program:

```plaintext
procedure P()
  var a : real,
      x : integer;
procedure Q(a : integer)
  var b : real;
  procedure R(b : integer)
  begin
    Q(2*b);
  end
begin
  ...
  R(a-1);
  ...
end
begin
  ...
  x := 2;
  Q(x);
  ...
end
```

(a) In the body of procedure \( R \), which variables and arguments are visible in scope (answer as in “\texttt{varname of procedurename}”)?

(b) Suppose \( P \) calls \( Q \), \( Q \) calls \( R \), and \( R \) calls \( Q \). Now, show the \textit{subroutine frames} on the stack with the slots for locals and arguments and the \textit{access links} (as in slide 14 of Ch.7 notes).

6. Consider Textbook 2nd ed. Figure 6.43 p.411 (Textbook 1st ed. pp.501–502). Assume that \( E_1 \) and \( E_2 \) are just the identifiers of variable names (as in Textbook 1st ed.). Give the parse tree with \textit{truelist} and \textit{falselist} annotations for the example input \( a>b \ &\& \text{true} \ |\ | \ c<d \ |\ | \text{false} \), assuming the emitted code starts at address 100. Give the backpatched code emitted for this example.