Problem 1 – Attribute Grammars and Syntax-Directed Translation Schemes

Assume the following partial grammar:

\[
\begin{align*}
\text{varcl} & ::= \text{idlist} : \text{type} \\
\text{idlist} & ::= \text{idlist}, \text{ID} | \text{ID} \\
\text{type} & ::= \text{integer} | \text{real} | \text{double}
\end{align*}
\]

1. Write an attribute grammar that computes the attribute \text{type} for each identifier, i.e., for each occurrence of an \text{ID} node in a subtree with \text{<varcl>} as its root. State for each attribute that you are using whether it is synthesized or inherited.

2. Show the parse tree for the input string
   \text{a, b, c, d : double}
   with all attribute instances and the final values of these attributes, i.e., show the decorated tree.

3. Is your attribute grammar S-attributed or L-attributed?

4. Write a syntax-directed translation scheme that stores the types of the variables in a symbol table. You may use pseudo code in your embedded actions. Assume that each \text{ID} has a pre-defined synthesized attribute \text{name} that contains its lexeme. The routine \text{insert(id, type)} inserts an identifier of a particular type into the symbol table. Use YACC-like notation (e.g. \$\$\$.name or \$1\.type).

Problem 2 – Type Systems

Assume a type system with the following inference rules

\[
\begin{align*}
\text{Rule}_{A1}: & \quad E \vdash e_1 : \text{integer} \quad E \vdash e_2 : \text{integer} \\
& \quad \frac{}{E \vdash (e_1 + e_2) : \text{integer}} \\
\text{Rule}_{A2}: & \quad E \vdash e : \text{integer} \\
& \quad \frac{}{E \vdash \&e : \text{pointer(integer)}} \\
\text{Rule}_{A3}: & \quad E \vdash e : \text{pointer(integer)} \\
& \quad \frac{}{E \vdash *e : \text{integer}}
\end{align*}
\]
Assuming that variable \( a \) and constant 3 are of type integer, use the inference rules to determine the types of the following expressions. Note: if a proof does not exist, the type system reports a type error.

1. \&a
2. \((\&a + 3)\)
3. \*a
4. \&3
5. \((\*\&a + 3)\)
6. \&\&a