Problem – Attribute Grammars and Syntax-Directed Translation Schemes

Assume the following partial grammar:

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

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{<vardcl>} 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).