Consider the subset of the attribute grammar for PL/0 handed out in class which has \texttt{<condition>} as its start symbol. Add type checking rules to the grammar as follows:

1. Augment the grammar with a \texttt{type} attribute which gives the type of a given node in the parse tree (e.g. \texttt{<expression> . type} is the type of the expression). Provide the rules for defining this attribute at appropriate places in the grammar, assuming the existence of a function \texttt{\textbf{type} (ident . lexeme, factor . env)} which returns the type of \texttt{ident} as stored in \texttt{factor . env}.

2. Where appropriate, augment the grammar with type checking rules of the form:
   
   \texttt{if nonterminal[1] . type \neq nonterminal[2] . type then error}

3. Assuming an initial \texttt{<condition> . env} of \{\texttt{<a, const (10)>}, \texttt{<b, var>}\}, draw an attributed parse tree showing clearly the order of evaluation of all attributes, including the \textit{if-then} error checking, for the condition \texttt{a \ast b = 0}. If an error occurs, you may halt processing.