CSE 351

Programming Languages

Homework Assignment #3

Due Date: 17th of April 2020 @17:00

1. A BNF grammar for expressions is given below:

```
<program> → <type_decls> <stmts>

<type_decls> → <type_decl> <type_decls>

<type_decls> → <type_decl>

<type_decl> → <type> <var_list> ;
      Semantic Rule: <var_list>.actual_type ← <type>.actual_type

<var_list>[1] → <var> , <var_list>[2]
      Semantic Rule: <var_list>[2].actual_type ← <var_list>[1].actual_type
      Semantic Rule: insert(<var>.string,<var_list>[1].actual_type)
      /* insert function updates the type of a variable on the Symbol Table. */

<var_list> → <var>
      Semantic Rule: insert(<var>.string,<var_list>.actual_type)

<type> → int
      Semantic Rule: <type>.actual_type ← int

<type> → float
      Semantic Rule: <type>.actual_type ← float


<stmts> → <stmt> <stmts>


<stmts> → <stmt>


<stmt> → <var> = <expr>
      Predicate: <var>.actual_type == <expr>.actual_type

<expr>[1] → <var> + <expr>[2]
      Semantic Rule: <expr>[1].actual_type ← if ((<var>.actual_type == int) and
                                               (<expr>[2].actual_type == int))
                                                  int
                                             else
                                                  float
                                             endif

<expr> → <var>
      Semantic Rule: <expr>.actual_type ← <var>.actual_type


<var> → A | B | C | D | E
      Semantic Rule: <var>.actual_type ← lookup(<var>.string)
      /* lookup function returns the type from the Symbol Table of a given variable name */
```

a) Draw the parse tree for the following small program.

int A, B;
float D;
A = B + D ;

b) Show the flow of attributes in the parse tree you draw for part a

c) Indicate if any semantic error is found in the program.

2. While coding, it is always hard to match opening and closing curly braces (i.e. { and } symbols) in C and Java. So, we would like to write a unique number next to each curly brace pair, as shown in the example code below.

```
public class Displayer { [1]
    public static void main(String args[]) { [2]
        System.out.println("You'll love Java!");
    } [2]
} [1]
```

Add semantic functions or predicates to achieve this for the given curly braces grammar below.

<curly> → <curly> <curly>

<curly> → { <curly> }

<curly> → { }

3. Add an attribute grammar over the given binary string grammar below, so that we can calculate its decimal value on the root node of any parse tree.

<binary> → <binary> <digit>

<binary> → <digit>

<digit> → 0

<digit> → 1

Draw the parse tree for the 1001 input, and show the attribute flow to calculate its decimal value of 9.