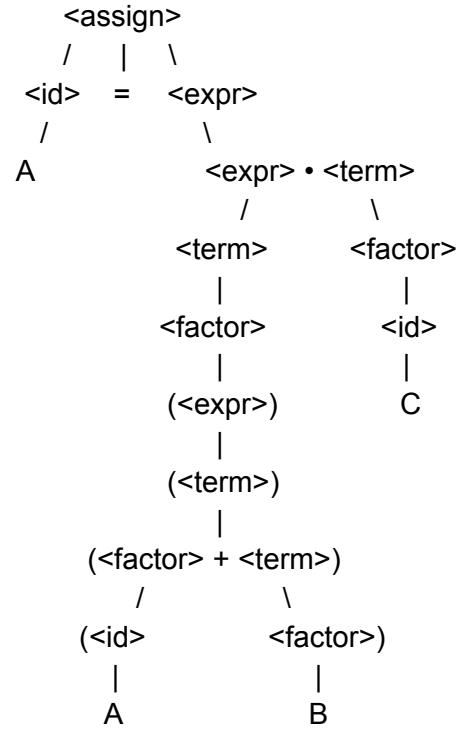
Gretel Rajamoney
1/26/2022
Homework 3

1. (Parse Tree)



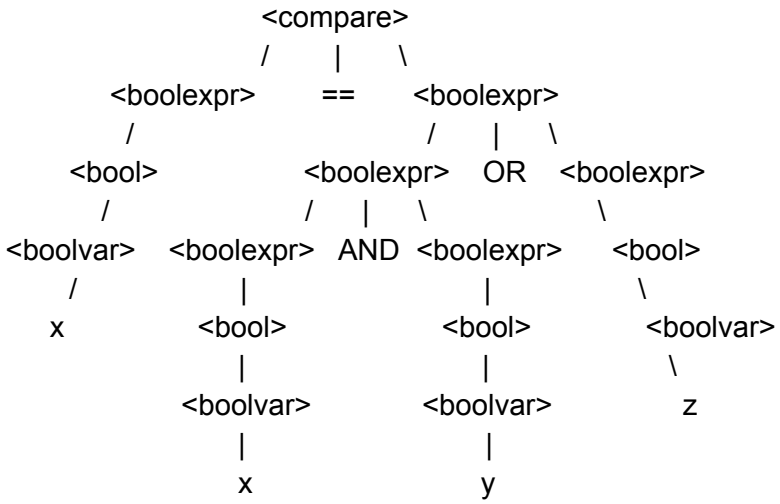
(Leftmost Derivation)

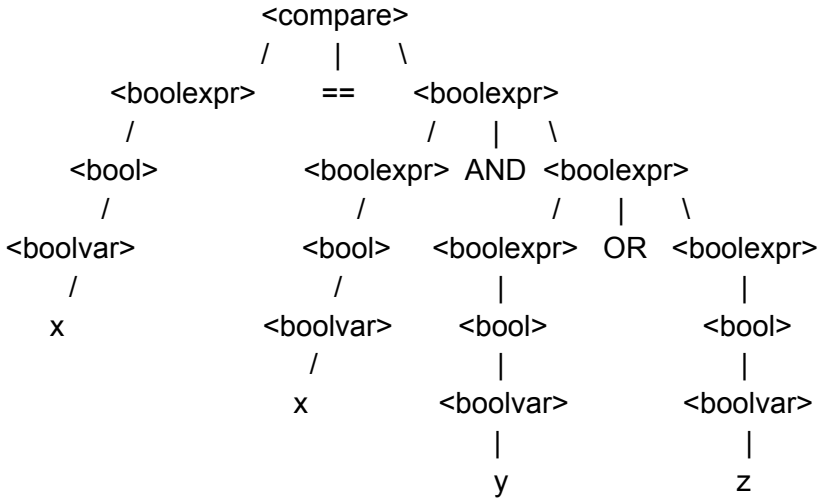
<assign> ---> <id> = <expr>
A = <expr>
A = <expr> • <term>
A = <term> • <term>
A = <factor> • <term>
A = (<expr>) • <term>
A = (<term>) • <term>
A = (<factor> + <term>) • <term>
A = (<id> + <term>) • <term>
A = (A + <term>) • <term>
A = (A + <factor>) • <term>
A = (A + <id>) • <term>
A = (A + B) • <term>
A = (A + B) • <factor>
A = (A + B) • <id>
A = (A + B) • C

2.

```
<assign> ---> <id> = <expr>
<expr>   ---> <expr> • <term>
               | <term>
<term>   ---> <factor> + <term>
               | <factor>
<factor> ---> (<expr>)
               | <id>
               | <id> ++
               | <id> - -
<id>       ---> A
               | B
               | C
```

3.





The grammar is ambiguous because the following can be defined in two ways.

4.

a. Public Class “class_name” extends “class_name” which implements “interface_name”
   <class_head> -> {<modifier>} class <id> [extends class_name] [implements <interface_name> {, <interface_name>}] <modifier> -> public | abstract | final
b. Switch(x)
   {
   Case 1: C = 1; Break;
   Case 2: C = 2; Break;
   Default: C = 0;
   }
   <switch_stmt> -> switch (<expr>) {{case <literal> : <stmt_list> [default : <stmt_list>]}
c. Union E
   {
   Int a;
   Float b;
   Long c;
   Char d;
   Double f;
   } e;
   <union_defn> -> union <var_list> <union_identifier>; <var_list> -> <list_of_data_type specifier> <var> <list_of_data_type specifier> -> int | float | long | char | double
   <union_identifier> -> <var>

5. Completed in the attached Haskell file!