

**Concepts of Programming Languages, CSCI 305, Fall 2021**  
**Attribute Grammars, Section 4.2, pages 184-187, Nov. 15**

Exercise 4.6, page 211 Add attributes rules to the following grammar to accumulate into the root of the tree a count of the maximum depth to which parentheses are nested in the program string. For example, given the string  $f1(a, f2(b * (c + (d - (e - f))))))$ , the *stmt* at the root of the tree should have an attribute with a count of 3 (the parentheses surrounding argument lists don't count).

1.  $stmt \rightarrow assignment$
2.  $stmt \rightarrow subr\_call$
3.  $assignment \rightarrow id := expr$
4.  $subr\_call \rightarrow id (arg\_list)$
5.  $expr \rightarrow primary\ expr\_tail$
6.  $expr\_tail \rightarrow op\ expr$
7.  $expr\_tail \rightarrow \epsilon$
8.  $primary \rightarrow id$
9.  $primary \rightarrow subr\_call$
10.  $primary \rightarrow ( expr )$
11.  $op \rightarrow + | - | * | /$
12.  $arg\_list \rightarrow expr\ args\_tail$
13.  $args\_tail \rightarrow ,\ arg\_list$
14.  $args\_tail \rightarrow \epsilon$

Sample answer:

1.  $stmt \rightarrow assignment$   
 $stmt.count := assignment.count$
2.  $stmt \rightarrow subr\_call$   
 $stmt.count := subr\_call.count$
3.  $assignment \rightarrow id := expr$   
 $assignment.count := expr.count$
4.  $subr\_call \rightarrow id (arg\_list)$   
 $subr\_call.count := arg\_list.count$
5.  $expr \rightarrow primary\ expr\_tail$   
 $expr.count := \max(primary.count, expr\_tail.count)$
6.  $expr\_tail \rightarrow op\ expr$   
 $expr\_tail.count := expr.count$
7.  $expr\_tail \rightarrow \epsilon$   
 $expr\_tail.count := 0$
8.  $primary \rightarrow id$   
 $primary.count := 0$

9.  $\text{primary} \rightarrow \text{subr\_call}$   
     $\text{primary.count} := \text{subr\_call.count}$
10.  $\text{primary} \rightarrow ( \text{expr} )$   
     $\text{primary.count} := \text{expr.count} + 1$
11.  $\text{op} \rightarrow + \mid - \mid * \mid /$
12.  $\text{arg\_list} \rightarrow \text{expr arg\_tail}$   
     $\text{arg\_list.count} := \max(\text{expr.count}, \text{arg\_tail.count})$
13.  $\text{args\_tail} \rightarrow , \text{arg\_list}$   
     $\text{args\_tail.count} := \text{arg\_list.count}$
14.  $\text{args\_tail} \rightarrow \varepsilon$   
     $\text{args\_tail.count} := 0$