Compilation Principle Homework 5

Jinyan XU, 3160101126, Information Security

6.7 Consider the following grammar for simple Pascal-style declarations：

decl → var-list : type

var-list → var-list, id | id

type → integer | real

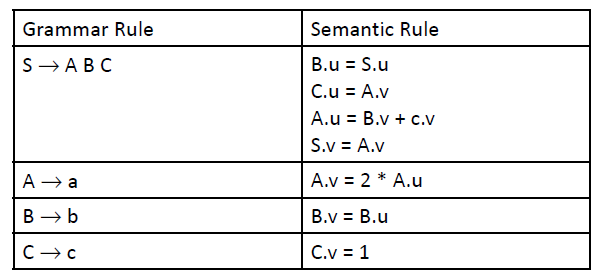
Write an attribute grammar for this grammar.

|  |
| --- |
| **Answer：**  decl → var-list : type var-list.dtype = type.dtype  var-list1 → var-list2, id var-list2.dtype = var-list1.dtype  id.dtype = var-list1.dtype  var-list → id id.dtype = var-list.dtype  type → integer type.dtype = int  type → real type.dtype = float |

6.8 Consider the grammar of 6.7. Rewrite the grammar so that the type of a variable can be purely synthesized attribute, and give a new attribute grammar for the type has this property.

|  |
| --- |
| **Answer:**  decl → id var-list id.dtype = var-list.dtype  var-list1 → , id var-list2 var-list1.dtype = var-list2.dtype  id.dtype = var-list2.dtype  var-list → : type var-list.dtype = type.dtype  type → integer type.dtype = int  type → real type.dtype = float |

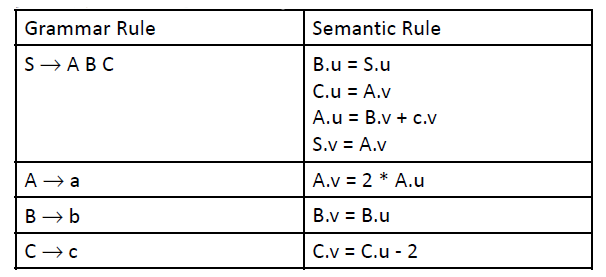
6.13 Consider the following attribute grammar:



(a) Draw the parse tree for string abc, and draw the dependency graph for the associated attributes. Describe a correct order for the evaluation of the attributes.

(b) Suppose that S.u is assigned the value of 3 before attribute evaluation begins. What is the value of S.v when the evaluation has finished?

(c) Suppose the attribute equations are modified as follows:



What value does S.v have after attribute evaluation, if S.u=3 before evaluation begins?

|  |
| --- |
| **Answer:**   1. A correct order for the evaluation of the attributes is: **1, 5, 6, 8, 3, 4, 2, 7**.      1. S.v = 8      1. There has a cycle in the dependency graph, we can’t get a topological sort.     Let’s try to get the value:    x = x/2 – 1, x = -2  So, S.v = -2 |