Prince Kannah
CS-354: Programming Languages
HW4

P209: 4.1

The language can still be recognized through an attribute grammar as follow:

Condition: A.count = B.count = C.count

A.count := 1

A1.count := A2.count+1

B.count := 1

B1.count := B2.count+1

C.count := 1

C1.count := C2.count+1

P286: 6.1

The statements are not contradictory. The associativity rules defines the order in which binary operators are applied within an expression. It does not specify the order in which the operands of a given operator are evaluated. Whereas the precedence rules define which operator has precedence of the other. Both rules however does not specify the order in which the compiler is needed to evaluate the operands of binary operator. The compilers are free to evaluate the operands in either order (left-to-right or right-to-left)

P287: 6.8

It is more than just a coincidence that languages that employ a reference model of variable also tend to use automatic garbage collection. Languages that use the value model of variables copies the objects having the same values. This in essences creates a separate instance for different objects having the same values. However, a language that employs a reference model propagates references to the same object. So, different objects having similar values refer to the same memory location. Because you could have several variables refer to the same location it becomes near impractical to keep track of all of them this forces reference model languages to use garbage collection.

P289: 6.25

```
// Using a while loop
/*
*Here the condition is checked at the beginning of the loop.
*The loop reads the line until there is no value to be read.
*The loop terminates if a blank line is encountered.
*/
line = read_line();
while(!all_blanks(line)){
  consume_line(line);
  line=read_line();
}
```
When compared to either the do-while or while loop the mid-test version is a good alternative. The while and do-while require more memory and is not easily understandable.

<u>P290: 6.26</u>
First we need to check if the code finds the first zero row of a matrix correctly. The outer loop is used to iterate each row the matrix one by one. The inner loop checks if a particular row is all zeros. It then checks each value of a row: if it is non-zero then we jump to the next label, otherwise if everything is zero then mark it as such and break out of the loop. Therefore the code works correctly in finding the first zero row. This same function can be achieved using a boolean flag without the goto statement. The flag can be use to keep track of rows with non-zero then within the inner loop check for non-zero and flip on (set to true) the flag. Then at the end of the outer loop (row iteration) check to see if the flag has been switch and if so, you have found your first zero row.