CS 311: Theory and Implementation of Programming Languages (Fall 2009)
Assignment 1

(Chapter 3)

Problem 6.a: Using the grammar in Example 3.2, show a parse tree and a leftmost derivation for this statement:

a) A = A * (B + (C * A))

Problem 13: Write a grammar for the language consisting of strings that have n copies of the letter followed by the same number of copies of the letter b, where n > 0. For example the strings *ab*, *aaaabbbb*, and *aaaaaaaabbbbbbbb* are in the language but *a*, *abbm*, *ba* and *aaabb* are not.

Problem 19.a,b: Compute the weakest precondition for each of the following assignment statements and postcondiotions:

a) a = 2 * (b – 1) – 1 {a > 0}
b) b = (c + 10) / 3 {b > 6}

Problem 20.a: Compute the weakest precondition for each of the following sequences of assignment statements and their postcondiotions:

a) a = 2 * b + 1;
   b = a - 3
   {a > 0}

Problem 23: Write an attribute grammar whose BNF basis is that of Example 3.6 in Section 3.4.5, but whose language rules are as follows: Data types cannot be mixed in expressions, but assignment statements need not have the same types on both sides of the assignment operator.

Problem 25: Prove the following program is correct:

```
{n > 0}
count = n;
sum = 0;
while count <> 0 do
        Sum = sum + count;
        count = count – 1;
end
{sum = 1 + 2 + … + n}
```