**CS4308**

**Programming Languages**

**Assignment #2**

1. Write a grammar for the language consisting of strings that have *n* copies of the letter a followed by the same number of copies of the letter b, where *n >* 0. For example, the strings ab, aaabbb, and aaaaabbbbb are in the language, but a, abb, ba, and aaabb are not in the language.
2. Using the grammar below, show a parse tree and a leftmost derivation for each of the following statements:

<assign> -> <id> = <expr>

<id> -> A | B | C

<expr> -> <id> + <expr>

| <id> \* <expr>

| (<expr>)

| <id>

* 1. A = A \* (B + (C \* A))

|  |  |
| --- | --- |
|  | **Solution** |
|  |  |

* 1. B = C \* (A \* C + B)

|  |  |
| --- | --- |
|  | **Solution** |

1. Prove the following grammar is ambiguous:

<assign> -> <id> = <expr>

<id> -> A | B | C

<expr> -> <expr> + <expr>

| <expr> \* <expr>

| (<expr>)

| <id>

I can prove a grammar is ambiguous by creating two leftmost derivations for the same sentence. Consider the sentence, A = B + C \* A. Below are two different leftmost derivations for the same sentence.

1. Consider the following grammar:

<S> -> <A> a <B> b

<A> -> <A> b | b

<B> -> a <B> | a

Which of the following sentences are in the language generated by this grammar?

* 1. baab
  2. bbbab
  3. bbaaaaaS
  4. bbaab

1. Convert the following EBNF to BNF:

<S> -> <A> { b<A> }

<A> -> a[b]<A>