**Amy Jiang**

**Homework#6**

**04/18/2017**

**Data Structures (CS/IS 211)**

**#3**

Write a recursive grammar for the language of strings of one or more letters. The first letter of each string must be uppercase, and all other letters in the string must be lowercase.

Language = { string s : first char of s is uppercase , the rest of s is lowercase }

< word > = < uppercase > | < uppercase > < lowercase > |

< uppercase > = A | B | C | . . . | Z

< lowercase > = a | b | c | . . . | z

**#5**

Consider a language of strings that contain only X’s, Y’s, and Z’s. A string in this language must begin with an X. If a Y is present in a string, it must be the final character of the string.

1. a. Write a recursive grammar for this language.
   * + - 1. < string > = X | < X > < CHAR > | < string > Y | < string > < CHAR >

< X > = X

< CHAR > = X | Z

1. b. Write all possible two-character strings of this language.

XX , XY , XZ

**# 6**

Consider a language of words, where each word is a string of dots and dashes. The following grammar describes this language:

< word > = < dot > | < dash >< word > | < word >< dot >

< dot > = •

< dash > = -

1. a. Write all three-character strings that are in this language.
   * + - 1. • • • , - • • , - - •
2. b. Is the string • • • • - - in this language? Explain.
   1. No, this string is not legal because from < word > , it states that we can only have dashes in the beginning.

**# 10**

Is “+ \* a – b / c + + d e – f g” a prefix expression? Explain in terms of the grammar for prefix expressions.

< prefix > = < identifier > | < operator > < prefix > < prefix >

< operator > = + | - | \* | /

< identifier > = a | b | . . . | z

This is not in prefix mode since when solved there is a single operator not used.

