CS 430 Grammar Lab

Part 1: Grammars and Ambiguity

Consider the following context-free grammar, written using the variant of BNF we've used in class.

```
A → V = E
E → E ^ E
  | E ~ E
  | V
V → a | b | c ... y | z
```

1. List all non-terminals.

2. How many productions are there total?

3. List all terminals.

4. Write a leftmost derivation for the statement "x = a ~ b ^ c" and draw the corresponding parse tree.

5. Is the grammar ambiguous? If it is, prove it by providing an example sentence with its leftmost derivations or parse trees.

Part 2: Associativity and Precedence

Consider the following grammar, written using the variant of BNF we've used in class.

```
A → V = E
E → E ^ F
  | E & F
  | F
F → F ~ G
  | G
G → ! G
  | V
V → a | b | c ... y | z
```

6. Is the & operator left or right associative? Give an example sentence and parse tree.

7. List all operators in order of precedence from higher to lower. If two operators have the same precedence, list them on the same line.

8. Modify the grammar to allow chained assignments (e.g., "x = y = a ~ b ^ c"). List only the productions for any non-terminals that you add or modify. Is the = operator left or right associative?