CS 345 — Fall 2001	Homework #1
Programming Languages	Due September 26, 2001

1. Draw the parse tree for "− + 45 ; 5 ; 6" given the following BNF grammar:

⟨*E*⟩ ::= − ⟨*E*⟩ ; ⟨*E*⟩ | + ⟨*E*⟩ ; ⟨*E*⟩ | ⟨*N*⟩

⟨*N*⟩ ::= ⟨*d*⟩ | ⟨*N*⟩⟨*d*⟩

⟨*d*⟩ ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

2. Convert the following prefix expressions into postfix.

(a) ++ 3 4 / c d

(b) − + 007 / Audrey 2 (Assume here that − is a unary operator)

3. Demonstrate that the following grammar is ambiguous.

⟨*E*⟩ ::= ⟨*N*⟩ | − ⟨*E*⟩ | − ⟨*E*⟩⟨*E*⟩

⟨*N*⟩ ::= ⟨*d*⟩ | ⟨*N*⟩⟨*d*⟩

⟨*d*⟩ ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

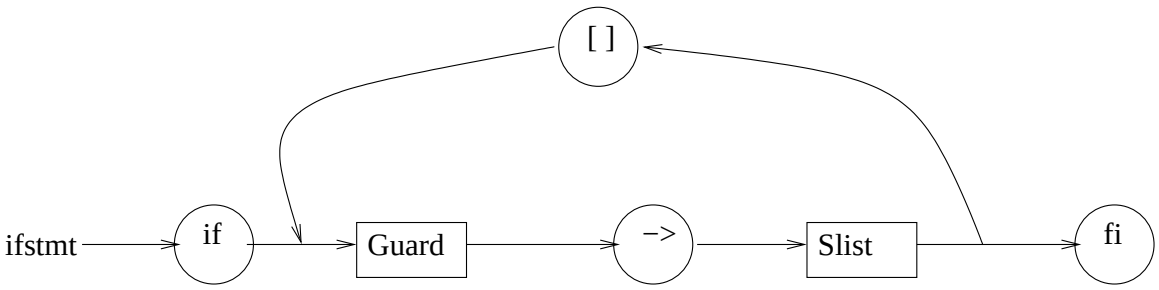
4. Rewrite the following EBNF grammar in BNF, where **Num** and **id** are syntactic terminals.

Guard ::= [∼] *Pred* { (∧ | ∨) [∼] *Pred* }

Pred ::= *Expr* (< | <= | > | >=) *Expr*

Expr ::= **Num** | **id**

5. Convert the following syntax chart into an EBNF production rule:



6. Write a BNF grammar that produces a GCN assignment statement.

Consider only the mathematical operations of addition, subtraction, multiplication, and division. Make sure the grammar forces the normal mathematical rules of left-association and precedence. Assume that the lexer has already removed whitespace, turned variable names into the terminal **id**, and turned numbers into the terminal **num**.

Note that GCN allows simultaneous multiple assignment. Since all assignments occur at the same time, you need not be concerned about which exact assignment order is implied by your grammar. You must make sure, however, that only syntactically legal assignments take place.

Make sure your grammar produces the following examples:

id := **id** + **num** * **id**

id, **id** := (**id** * **num**) / **num**, **num** - **num**

Make sure your grammar does not produce the following examples:

id, **id**, **id** := **num**, **num**

id + **num** := **num**