Grammars

Consider an expression language with two binary operators "$" and "#" and two unary prefix operators "*" and "@". It also includes a single alphabetic symbol "X" along with parenthesis "(" and ")". An EBNF expression grammar is given below where the starting non-terminal is <expr> and terminals are highlighted.

<expr> ::= <term> { $ <term>}
<term> ::= <factor> { # <factor>}
<factor> ::= [@ | *] (X | (<expr>))

Problem

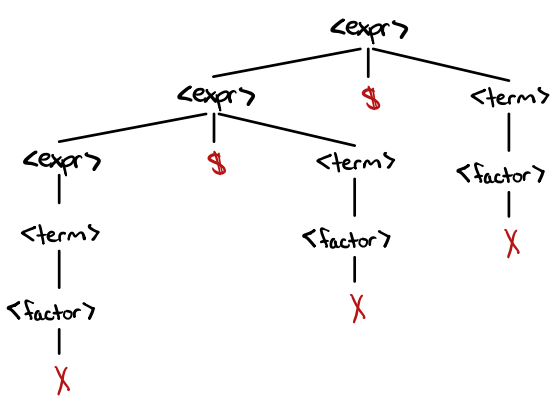
1. Give an equivalent unambiguous BNF grammar for this expression language.

<expr> ::= <term>
<expr> ::= <expr> $ <term>
<term> ::= <factor>
<term> ::= <term> # <factor>
<factor> ::= X | *X | @X
<factor> ::= (<expr>) | *(<expr>) | @(<expr>)

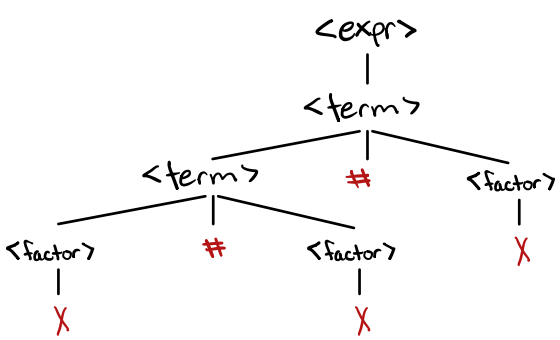
2. Give the associativity of the two binary operators and a precedence table for all four operators in your BNF grammar.

Operator	Associativity	Precedence
*	Right-to-left	1
@	Right-to-left	
$	Left-to-right	2
#	Left-to-right	

X$ X$ X



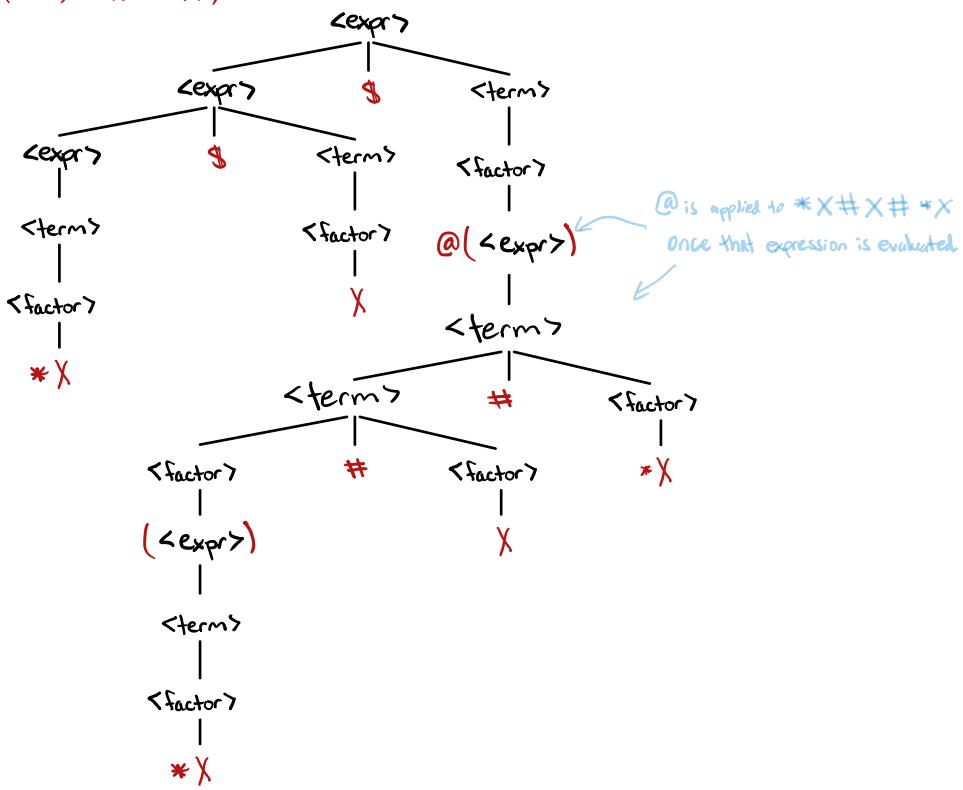
X#X#X



In both these example strings, the parse tree is left-skewed which means the left side of the expression must be evaluated before the right side. Therefore, the “$“ and the “#” operators are left-to-right associative. The right side of the unary operators “*” and “@“ must be evaluated before these operations can be applied, therefore these operators are right-to-left associative. In the parse tree bellow, the unary operators occur higher on the tree since they are applied an evaluated expression. This means the unary operators take precedence over the binary operators.

3. Using your BNF specification, provide a parse tree for the expression `*X$X$@((*X)#X#*X)` .

*X $X $@((*X) # X # *X)



Syntactic Ambiguity

Prove that the following BNF grammar is ambiguous.

```
N = {<S>, <A>, <I>}
T = {a, b, c, x}
P = {<S> ::= <A>
     <A> ::= <A>x<A>
     <A> ::= <I>
     <I> ::= a
     <I> ::= b
     <I> ::= c}
S = <S>
```

A grammar is syntactically ambiguous if it contains a string with one or more distinct parse trees. Given the string “axbxcxcx,” the grammar provided at the left may result in either of the parse trees below or other possible trees. Since there are two or more distinct trees for this given string, those grammar can be declared syntactically ambiguous. In other words, this grammar may imply two meanings for the string “axbxcxcx.”