

Programming Languages
Assignment 1

Q1: Using the BNF grammar below show a **parse tree** and a **leftmost derivation** for each of the following 4 statements:

- 1) $A = (A + B) * C$
- 2) $A = B + C + A$
- 3) $A = A * (B + C)$
- 4) $A = B + (C * (A * B))$

BNF grammar

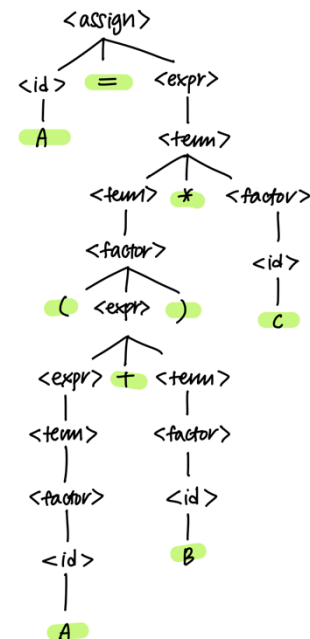
$\langle \text{assign} \rangle \rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle$
 $\langle \text{id} \rangle \rightarrow A \mid B \mid C$
 $\langle \text{expr} \rangle \rightarrow \langle \text{expr} \rangle + \langle \text{term} \rangle \mid \langle \text{term} \rangle$
 $\langle \text{term} \rangle \rightarrow \langle \text{term} \rangle * \langle \text{factor} \rangle \mid \langle \text{factor} \rangle$
 $\langle \text{factor} \rangle \rightarrow (\langle \text{expr} \rangle) \mid \langle \text{id} \rangle$

1) $A = (A + B) * C$

Leftmost derivation:

$\langle \text{assign} \rangle$
 $\rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle$
 $\rightarrow A = \langle \text{expr} \rangle$
 $\rightarrow A = \langle \text{term} \rangle$
 $\rightarrow A = \langle \text{term} \rangle * \langle \text{factor} \rangle$
 $\rightarrow A = \langle \text{factor} \rangle * \langle \text{factor} \rangle$
 $\rightarrow A = (\langle \text{expr} \rangle) * \langle \text{factor} \rangle$
 $\rightarrow A = (\langle \text{expr} \rangle + \langle \text{term} \rangle) * \langle \text{factor} \rangle$
 $\rightarrow A = (\langle \text{term} \rangle + \langle \text{term} \rangle) * \langle \text{factor} \rangle$
 $\rightarrow A = (\langle \text{factor} \rangle + \langle \text{term} \rangle) * \langle \text{factor} \rangle$
 $\rightarrow A = (\langle \text{id} \rangle + \langle \text{term} \rangle) * \langle \text{factor} \rangle$
 $\rightarrow A = (A + \langle \text{term} \rangle) * \langle \text{factor} \rangle$
 $\rightarrow A = (A + \langle \text{factor} \rangle) * \langle \text{factor} \rangle$
 $\rightarrow A = (A + \langle \text{id} \rangle) * \langle \text{factor} \rangle$
 $\rightarrow A = (A + B) * \langle \text{factor} \rangle$
 $\rightarrow A = (A + B) * \langle \text{id} \rangle$
 $\rightarrow A = (A + B) * C$

Parse Tree:



Programming Languages

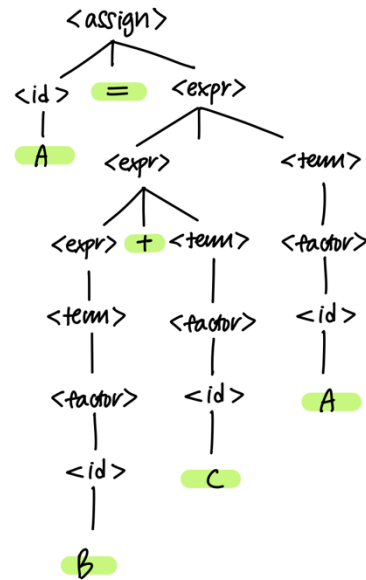
Assignment 1

2) $A = B + C + A$

Leftmost derivation:

$\langle \text{assign} \rangle$
 $\rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle$
 $\rightarrow A = \langle \text{expr} \rangle$
 $\rightarrow A = \langle \text{expr} \rangle + \langle \text{term} \rangle$
 $\rightarrow A = \langle \text{expr} \rangle + \langle \text{term} \rangle + \langle \text{term} \rangle$
 $\rightarrow A = \langle \text{term} \rangle + \langle \text{term} \rangle + \langle \text{term} \rangle$
 $\rightarrow A = \langle \text{factor} \rangle + \langle \text{term} \rangle + \langle \text{term} \rangle$
 $\rightarrow A = \langle \text{id} \rangle + \langle \text{term} \rangle + \langle \text{term} \rangle$
 $\rightarrow A = B + \langle \text{factor} \rangle + \langle \text{term} \rangle$
 $\rightarrow A = B + \langle \text{id} \rangle + \langle \text{term} \rangle$
 $\rightarrow A = B + C + \langle \text{term} \rangle$
 $\rightarrow A = B + C + \langle \text{factor} \rangle$
 $\rightarrow A = B + C + \langle \text{id} \rangle$
 $\rightarrow A = B + C + A$

Parse Tree:

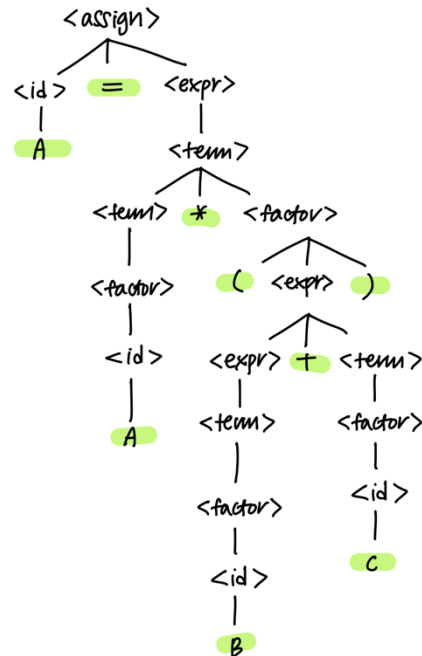


3) $A = A * (B + C)$

Leftmost derivation:

$\langle \text{assign} \rangle$
 $\rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle$
 $\rightarrow A = \langle \text{expr} \rangle$
 $\rightarrow A = \langle \text{term} \rangle$
 $\rightarrow A = \langle \text{term} \rangle * \langle \text{factor} \rangle$
 $\rightarrow A = \langle \text{factor} \rangle * \langle \text{factor} \rangle$
 $\rightarrow A = \langle \text{id} \rangle * \langle \text{factor} \rangle$
 $\rightarrow A = A * \langle \text{expr} \rangle$
 $\rightarrow A = A * \langle \text{expr} \rangle + \langle \text{term} \rangle$
 $\rightarrow A = A * \langle \text{term} \rangle + \langle \text{term} \rangle$
 $\rightarrow A = A * \langle \text{factor} \rangle + \langle \text{term} \rangle$
 $\rightarrow A = A * \langle \text{id} \rangle + \langle \text{term} \rangle$
 $\rightarrow A = A * B + \langle \text{term} \rangle$
 $\rightarrow A = A * B + \langle \text{factor} \rangle$
 $\rightarrow A = A * B + \langle \text{id} \rangle$
 $\rightarrow A = A * B + C$

Parse Tree:



Programming Languages

Assignment 1

4) $A = B + (C * (A * B))$

Leftmost derivation:

$\langle \text{assign} \rangle$

$\rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle$

$\rightarrow A = \langle \text{expr} \rangle$

$\rightarrow A = \langle \text{expr} \rangle + \langle \text{term} \rangle$

$\rightarrow A = \langle \text{term} \rangle + \langle \text{term} \rangle$

$\rightarrow A = \langle \text{factor} \rangle + \langle \text{term} \rangle$

$\rightarrow A = \langle \text{id} \rangle + \langle \text{term} \rangle$

$\rightarrow A = B + \langle \text{term} \rangle$

$\rightarrow A = B + \langle \text{factor} \rangle$

$\rightarrow A = B + (\langle \text{expr} \rangle)$

$\rightarrow A = B + (\langle \text{term} \rangle)$

$\rightarrow A = B + (\langle \text{term} \rangle * \langle \text{factor} \rangle)$

$\rightarrow A = B + (\langle \text{factor} \rangle * \langle \text{factor} \rangle)$

$\rightarrow A = B + (\langle \text{id} \rangle * \langle \text{factor} \rangle)$

$\rightarrow A = B + (C * \langle \text{factor} \rangle)$

$\rightarrow A = B + (C * (\langle \text{expr} \rangle))$

$\rightarrow A = B + (C * (\langle \text{term} \rangle))$

$\rightarrow A = B + (C * (\langle \text{term} \rangle * \langle \text{factor} \rangle))$

$\rightarrow A = B + (C * (\langle \text{factor} \rangle * \langle \text{factor} \rangle))$

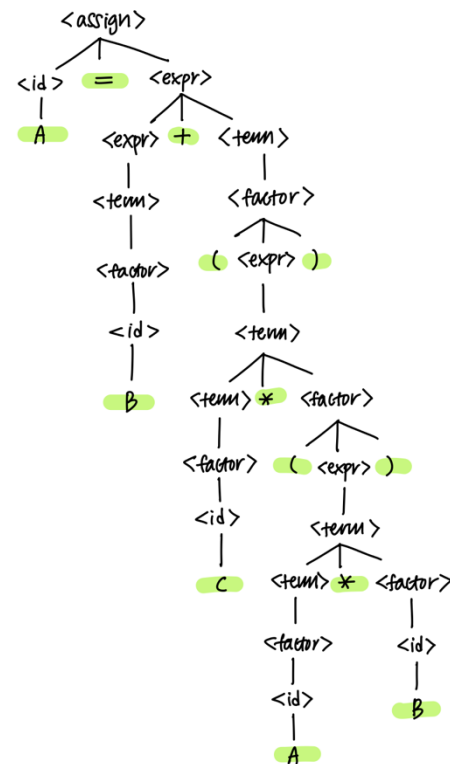
$\rightarrow A = B + (C * (\langle \text{id} \rangle * \langle \text{factor} \rangle))$

$\rightarrow A = B + (C * (\langle A \rangle * \langle \text{factor} \rangle))$

$\rightarrow A = B + (C * (\langle A \rangle * \langle \text{id} \rangle))$

$\rightarrow A = B + (C * (A * B))$

Parse Tree:



Programming Languages
Assignment 1

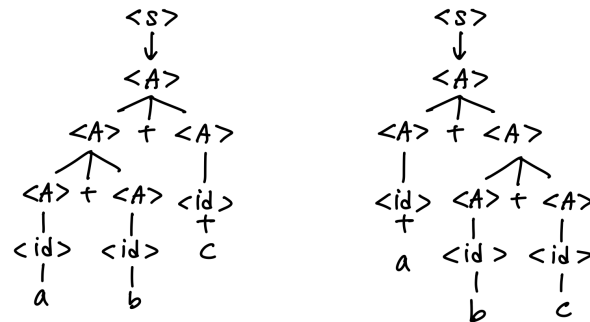
Q2: Prove that the following grammar is ambiguous:

BNF grammar

```
<S> → <A>
<A> → <A> + <A> | <id>
<id> → a | b | c
```

The BNF grammar above is ambiguous because the same expression can be interpreted in multiple ways.

For instances, the expression “a + b + c” can lead to two different parsing trees:



Having multiple parse trees for the same expression indicated that the BNF grammar is ambiguous, as the same expression does not have a single definitive way of being evaluated.

Q3: Modify the grammar below to add a unary minus — operator and ^ power operator that have higher precedence than either + or *.
Thus, the precedence of the operators in the final BNF should be ranked from the **highest** to the **lowest** as follows: (), —, ^, *, +
Also, all the operators have left associativity except the power operator ^ and unary minus —, which have right associativity.

BNF grammar

```
<assign> → <id> = <expr>
<id> → A | B | C
<expr> → <expr> + <term> | <term>
<term> → <term> * <factor> | <factor>
<factor> → ( <expr> ) | <id>
```

Modified grammar:

```
<assign> → <id> = <expr>
<id> → A | B | C
<expr> → <expr> + <term> | <term>
<term> → <term> * <power> | <power>
<power> → <unary> ^ <power> | <unary>
<unary> → - <unary> | <factor>
<factor> → ( <expr> ) | <id>
```