

Assignment

Name : Ishita Verma

Univ ID : N17385760

NetID : iv447

Chapter 3 : Problem Set

3. Rewrite the BNF of Example 3.4 to give + precedence over * and force + to be right associative.

Eg: 3.4 $\langle \text{assign} \rangle \rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle$

(Grammar)

$\langle \text{id} \rangle \rightarrow A | B | C$

$\langle \text{expr} \rangle \rightarrow \langle \text{expr} \rangle + \langle \text{term} \rangle |$
 $\langle \text{term} \rangle$

$\langle \text{term} \rangle \rightarrow \langle \text{term} \rangle * \langle \text{factor} \rangle | \langle \text{factor} \rangle$

$\langle \text{factor} \rangle \rightarrow (\langle \text{expr} \rangle) | \langle \text{id} \rangle$

Answer:-

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

$\langle \text{id} \rangle \rightarrow A | B | C$

$\langle \text{expr} \rangle \rightarrow \langle \text{expr} \rangle * \langle \text{term} \rangle |$
 $\langle \text{term} \rangle$

$\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle + \langle \text{term} \rangle | \langle \text{factor} \rangle$

$\langle \text{factor} \rangle \rightarrow (\langle \text{expr} \rangle) | \langle \text{id} \rangle$

4. Rewrite the BNF of Example 3.4 to add the ++ and -- Unary Operators of Java.

Answer :-

Answer:

$$\begin{aligned}
 <\text{assign}> &\rightarrow <\text{id}> = <\text{expr}> \\
 <\text{expr}> &\rightarrow <\text{expr}> + <\text{term}> \\
 &\quad | <\text{term}> \\
 <\text{term}> &\rightarrow <\text{id}> | <\text{post-increment}> | \\
 &\quad | <\text{post-decrement}> | <\text{pre-increment}> \\
 &\quad | <\text{pre-decrement}> \\
 <\text{post-increment}> &\rightarrow <\text{id}> ++ \\
 <\text{post-decrement}> &\rightarrow <\text{id}> -- \\
 <\text{pre-increment}> &\rightarrow ++ <\text{id}> \\
 <\text{pre-decrement}> &\rightarrow -- <\text{id}> \\
 <\text{id}> &\rightarrow A | B | C
 \end{aligned}$$

6. Using the grammar in Example 3.2, show a parse tree and leftmost derivation for each of the following statements:

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

Ex 3.2 : $\begin{aligned}
 <\text{assign}> &\rightarrow <\text{id}> = <\text{expr}> \\
 <\text{id}> &\rightarrow A | B | C \\
 <\text{expr}> &\rightarrow <\text{id}> + <\text{expr}> \\
 &\quad | <\text{id}> * <\text{expr}> \\
 &\quad | (<\text{expr}>) \\
 &\quad | <\text{id}>
 \end{aligned}$

Answer a. $A = A * (B + (C * A))$

$\langle \text{assign} \rangle \Rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle$

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

$\Rightarrow A = \langle \text{id} \rangle * \langle \text{expr} \rangle$

$\Rightarrow A = A * \langle \text{expr} \rangle$

$\Rightarrow A = A * (\langle \text{expr} \rangle)$

$\Rightarrow A = A * (\langle \text{id} \rangle + \langle \text{expr} \rangle)$

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

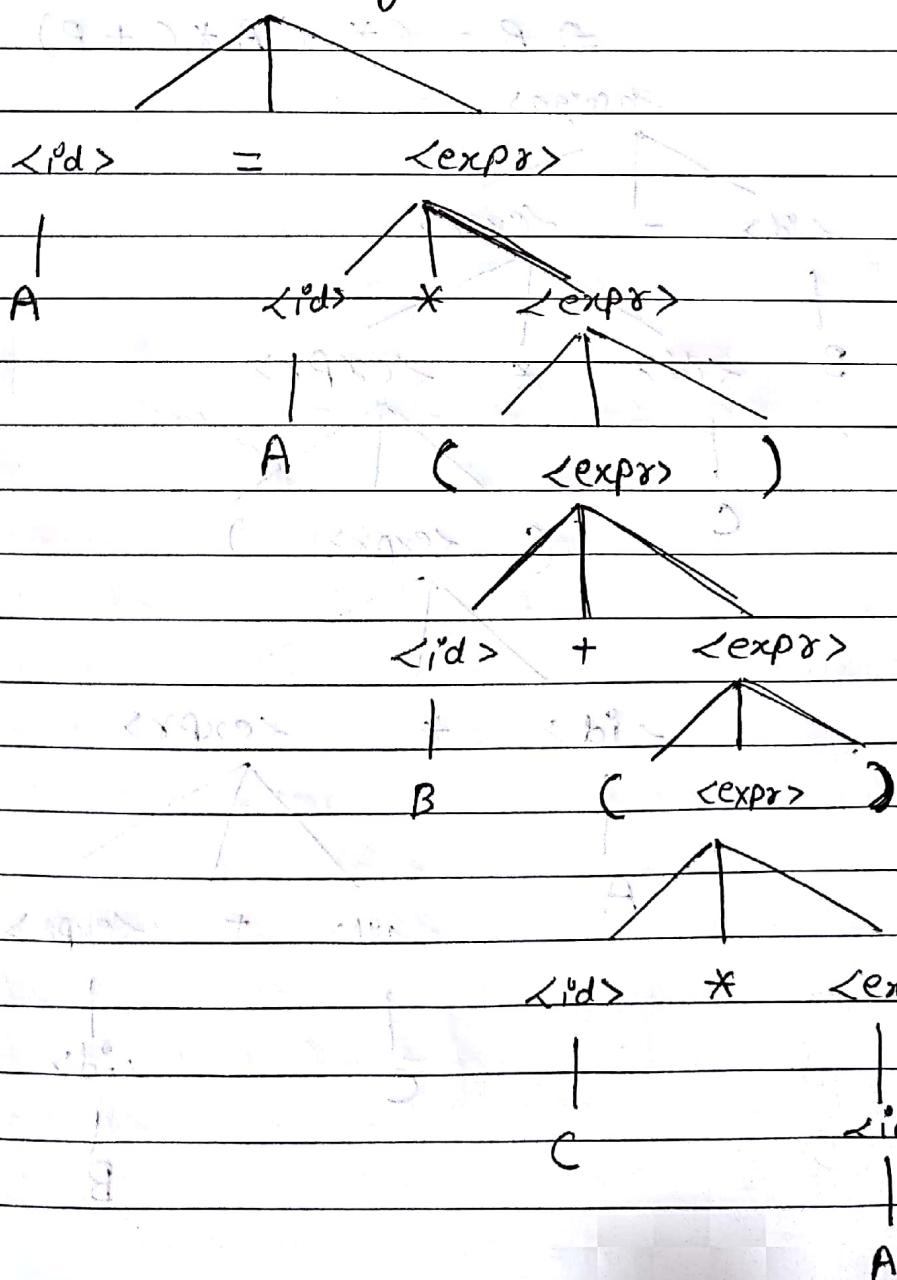
$\Rightarrow A = A * (B + (C * \langle \text{id} \rangle))$

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

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

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

Parse Tree:



b. $B = C * (A * C + B)$

Leftmost

Derivation: $\langle \text{assign} \rangle \Rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle$

$$\Rightarrow B = \langle \text{expr} \rangle$$

$$\Rightarrow B = \langle \text{id} \rangle * \langle \text{expr} \rangle$$

$$\Rightarrow C * \langle \text{id} \rangle * \langle \text{expr} \rangle \Rightarrow B = C * \langle \text{expr} \rangle$$

$$\Rightarrow C * \langle \text{id} \rangle * \langle \text{expr} \rangle \Rightarrow B = C * (C * \langle \text{expr} \rangle)$$

$$\Rightarrow B = C * (C * (\langle \text{id} \rangle * \langle \text{expr} \rangle))$$

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

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

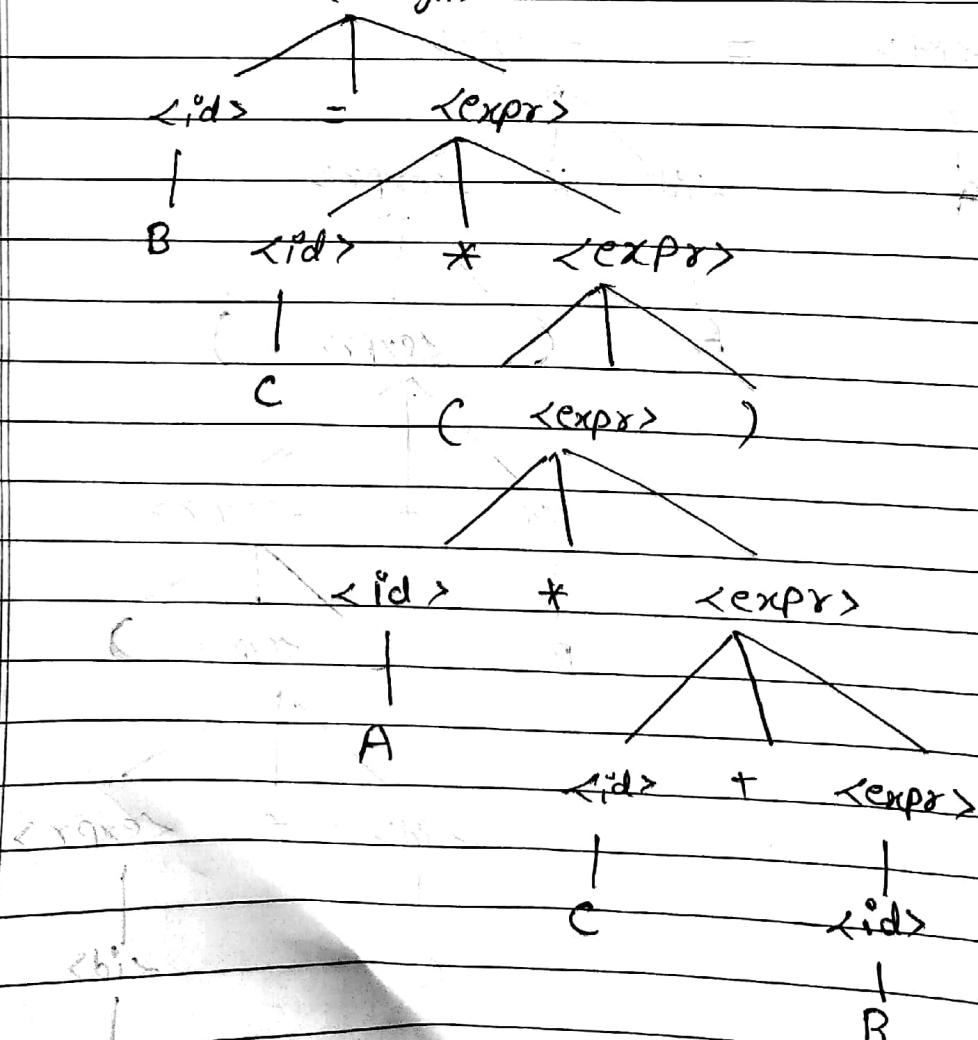
$$\Rightarrow B = C * (C * (A * C + \langle \text{expr} \rangle))$$

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

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

$\langle \text{assign} \rangle$

Parse
Tree:



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

leftmost derivation:-

$\langle \text{assign} \rangle \Rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle$

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

$$A = \langle \text{id} \rangle * \langle \text{expr} \rangle$$

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

$$A = A * (\langle \text{expr} \rangle)$$

$$A = A * (\langle \text{id} \rangle + \langle \text{expr} \rangle)$$

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

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

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

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

Parse Tree:-

$\langle \text{assign} \rangle$



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

A



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

A

$$\text{constant expression} = (\langle \text{expr} \rangle)$$

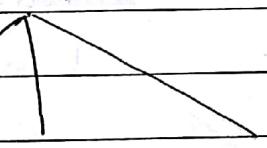
$$\text{constant expression} = A$$

$$A = (\text{constant} + \text{constant}) = A$$

$$\text{constant} + (\text{constant} + \text{constant}) = A$$

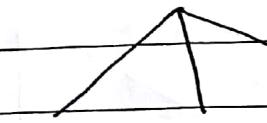
$$\text{constant} + (\text{constant} + A) = A$$

B



$$\langle \text{expr} \rangle = (\langle \text{expr} \rangle)$$

$$\text{constant} + (\text{constant} + A) = A$$



$$\langle \text{expr} \rangle = (\langle \text{id} \rangle)$$

C

$$\text{constant} + (\text{constant} + A) = A$$

$$\text{constant} + (B + A) = A$$

$$\langle \text{expr} \rangle * (\text{constant} + A) = A$$

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

7. Using the grammar in Example 3.4, show a parse tree and leftmost derivation for each of the following statements.

a. $A = (A+B)*C$

b. $A = B+C+A$

c. $A = A * (B+C)$

d. $A = B * C * (A+B)$

Ex. 3.4 $\text{<assign>} \rightarrow \text{id} - \text{expr}$

$\text{<expr>} \rightarrow \text{id} \rightarrow A | B | C$

$\text{<expr>} \rightarrow \text{expr} + \text{term}$

$(\text{expr} + \text{expr}) \mid \text{term}$

$\text{<term>} \rightarrow \text{term} * \text{factor}$

$\mid \text{factor}$

$\text{<factor>} \rightarrow (\text{expr})$

$\mid \text{id}$

a. $A = (A+B)*C$

Leftmost

$\text{<assign>} \Rightarrow \text{id} - \text{expr}$

$\Rightarrow A = \text{expr}$

$\Rightarrow A = \text{term} * \text{factor}$

$\Rightarrow A = \text{factor} * \text{factor}$

$\Rightarrow A = (\text{expr}) * \text{factor}$

$\Rightarrow A = (\text{term} + \text{term}) * \text{factor}$

$\Rightarrow A = (\text{factor} + \text{term}) * \text{factor}$

$\Rightarrow A = (A + \text{term}) * \text{factor}$

$\Rightarrow A = (A + \text{factor}) * \text{factor}$

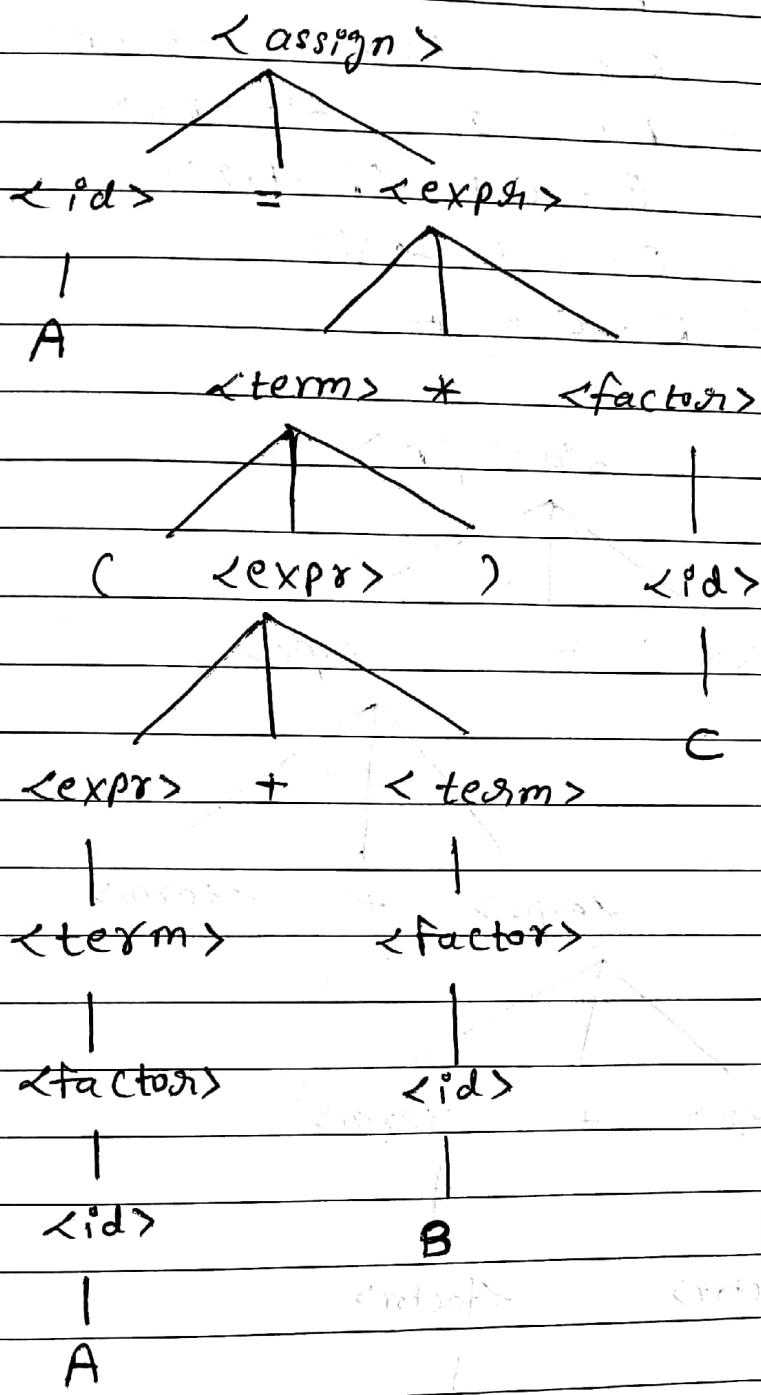
$\Rightarrow A = (A + \text{id}) * \text{factor}$

$\Rightarrow A = (A+B) * \text{factor}$

$\Rightarrow A = (A+B) * \text{id}$

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

parse
Tree:



$$b: A = B + C + A$$

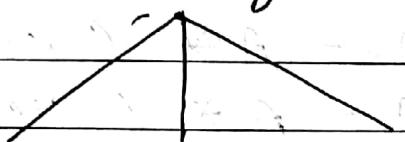
leftmost
Derivation:

$$\begin{aligned}
 & \text{<assign>} \Rightarrow \text{<id>} = \text{<expr>} \\
 & \Rightarrow A = \text{<expr>} \\
 & \Rightarrow A = \text{<expr>} + \text{<term>} \\
 & \Rightarrow A = \text{<expr>} + \text{<term>} + \text{<term>} \\
 & \Rightarrow A = \text{<term>} + \text{<term>} + \text{<term>} \\
 & \Rightarrow A = \text{<factor>} + \text{<term>} + \text{<term>} \\
 & \Rightarrow A = \text{<id>} + \text{<term>} + \text{<term>}
 \end{aligned}$$

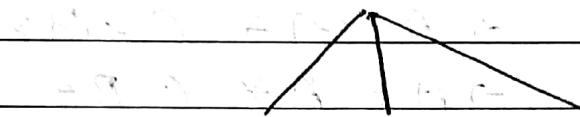
- $\Rightarrow A = B + \langle \text{term} \rangle + \langle \text{term} \rangle$
 $\Rightarrow A = B + \langle \text{factor} \rangle + \langle \text{term} \rangle$
 $\Rightarrow A = B + \langle \text{id} \rangle + \langle \text{term} \rangle$
 $\Rightarrow A = B + C + \langle \text{term} \rangle$
 $\Rightarrow A = B + C + \langle \text{factor} \rangle$
 $\Rightarrow A = B + C + \langle \text{id} \rangle$
 $\Rightarrow A = B + C + A$

Parse
Tree :-

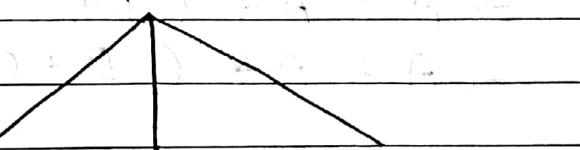
$\langle \text{assign} \rangle$



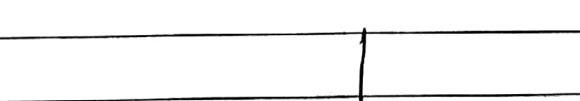
$\langle \text{id} \rangle = \langle \text{expr} \rangle$



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



$\langle \text{expr} \rangle + \langle \text{term} \rangle \langle \text{factor} \rangle$



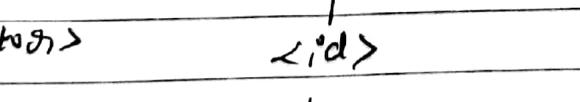
$\langle \text{term} \rangle$



$\langle \text{factor} \rangle$



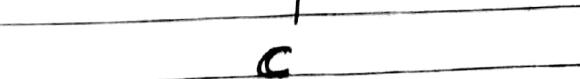
$\langle \text{id} \rangle$



$\langle \text{factor} \rangle$



$\langle \text{id} \rangle$



A



$\langle \text{id} \rangle$



C



B

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

leftmost
Derivation:

$\langle \text{assign} \rangle \Rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle$

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

$\Rightarrow A = \langle \text{term} \rangle * \langle \text{factor} \rangle$

$\Rightarrow A = \langle \text{factor} \rangle * \langle \text{factor} \rangle$

$\Rightarrow A = \langle \text{id} \rangle * \langle \text{factor} \rangle$

$\Rightarrow A = A * \langle \text{factor} \rangle$

$\Rightarrow A = A * (\langle \text{expr} \rangle)$

$\Rightarrow A = A * (\langle \text{expr} \rangle + \langle \text{term} \rangle)$

$\Rightarrow A = A * (\langle \text{term} \rangle + \langle \text{term} \rangle)$

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

$\Rightarrow A = A * (\langle \text{id} \rangle + \langle \text{term} \rangle)$

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

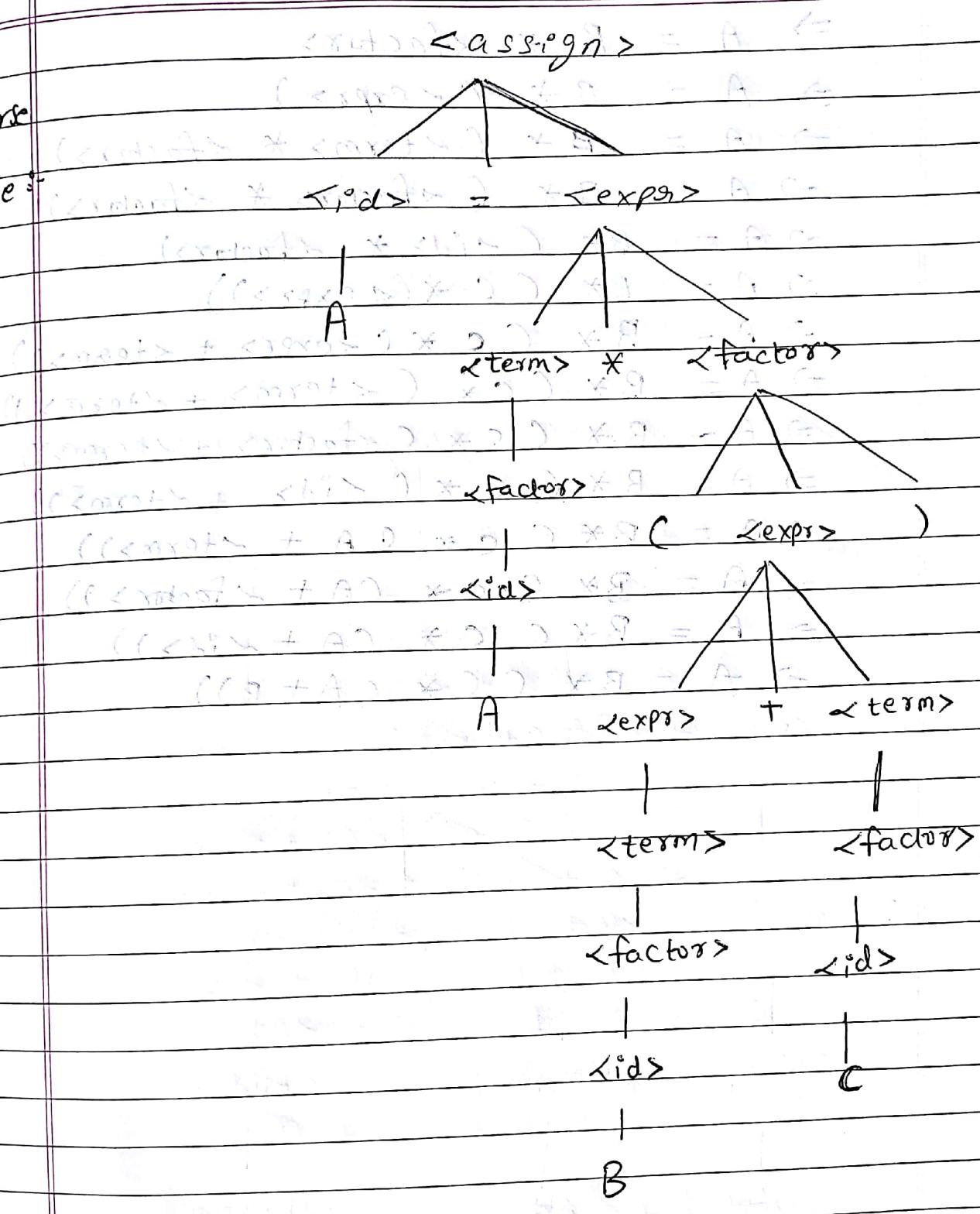
$\Rightarrow A = A * (B + \langle \text{factor} \rangle)$

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

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

parse

Tree:



d. $A = B * (C * (A + B))$

leftmost
derivation:

$$\langle \text{assign} \rangle \Rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle$$

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

$$\Rightarrow A = \langle \text{term} \rangle * \langle \text{factor} \rangle$$

$$\Rightarrow A = \langle \text{factor} \rangle * \langle \text{factor} \rangle$$

$$\Rightarrow A = \langle \text{id} \rangle * \langle \text{factor} \rangle$$

$\Rightarrow A = B * \langle \text{factor} \rangle$

$\Rightarrow A = B * (\langle \text{expr} \rangle)$

$\Rightarrow A = B * (\langle \text{terms} \rangle * \langle \text{factor} \rangle)$

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

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

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

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

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

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

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

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

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

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

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

<assign>

Parse

Tree:

<id>

=

<expr>

A

<term>

* <factors>

<factor>

C

<expr>

<id>

<term>

* <factors>

B

<factor>

C <expr>

<expr> + <term> - <term> + <id>

<expr> + <term>

C

<terms>

<factors>

<factors>

<id>

<ids>

B

A

8. Prove that the following grammar is ambiguous:

$$\begin{aligned} & \langle S \rangle \rightarrow \langle A \rangle \\ & \langle A \rangle \rightarrow \langle A \rangle + \langle A \rangle \mid \langle id \rangle \\ & \langle id \rangle \rightarrow a/b/c \end{aligned}$$

Answer: Let us consider string $a + b + c + a$

Leftmost $a + b + c + a$

Derivation: $\langle S \rangle \rightarrow \langle A \rangle$

(i)

$$\Rightarrow \langle A \rangle + \langle A \rangle$$

$$\Rightarrow \langle id \rangle + \langle A \rangle$$

$$\Rightarrow a + \langle A \rangle$$

$$\Rightarrow a + \langle A \rangle + \langle A \rangle$$

$$\Rightarrow a + \langle id \rangle + \langle A \rangle + \langle A \rangle$$

$$\Rightarrow a + b + \langle A \rangle + \langle A \rangle$$

$$\Rightarrow a + b + \langle id \rangle + \langle A \rangle$$

$$\Rightarrow a + b + c + \langle A \rangle$$

$$\Rightarrow a + b + c + \langle id \rangle$$

$$\Rightarrow a + b + c + a$$

Leftmost $a + b + c + a$

Derivation

(ii)

$$\langle S \rangle \rightarrow \langle A \rangle$$

$$\Rightarrow \langle A \rangle + \langle A \rangle$$

$$\Rightarrow \langle A \rangle + \langle A \rangle + \langle A \rangle$$

$$\Rightarrow \langle A \rangle + \langle A \rangle + \langle A \rangle + \langle A \rangle$$

$$\Rightarrow \langle id \rangle + \langle A \rangle + \langle A \rangle + \langle A \rangle$$

$$\Rightarrow a + \langle A \rangle + \langle A \rangle + \langle A \rangle$$

$$\Rightarrow a + \langle id \rangle + \langle A \rangle + \langle A \rangle$$

$$\Rightarrow a + b + \langle A \rangle + \langle A \rangle$$

$\Rightarrow a + b + \langle id \rangle + \langle A \rangle$

$\Rightarrow a + b + C + \langle A \rangle$

$\Rightarrow a + b + C + \langle id \rangle$

$\Rightarrow a + b + C + a$

Parse Tree 1: $a + b + C + a$

$\langle S \rangle : i f \langle A \rangle \rightarrow \langle A \rangle$

| $\rightarrow \langle A \rangle \rightarrow \langle A \rangle$

$\langle A \rangle \rightarrow \langle id \rangle \rightarrow a$

$\langle A \rangle \rightarrow \langle id \rangle \rightarrow a$

$\langle A \rangle \rightarrow \langle id \rangle \rightarrow a$

$\langle A \rangle \rightarrow \langle id \rangle \rightarrow a$

$\langle A \rangle \rightarrow \langle id \rangle \rightarrow a$

$\langle A \rangle \rightarrow \langle id \rangle \rightarrow a$

$\langle A \rangle \rightarrow \langle id \rangle \rightarrow a$

$\langle A \rangle \rightarrow \langle id \rangle \rightarrow a$

Parse
Tree 2:

$\langle S \rangle : i f \langle A \rangle \rightarrow \langle A \rangle$

$\langle A \rangle \rightarrow \langle id \rangle \rightarrow a$

$\langle A \rangle \rightarrow \langle id \rangle \rightarrow a$

$\langle A \rangle \rightarrow \langle id \rangle \rightarrow a$

$\langle A \rangle \rightarrow \langle id \rangle \rightarrow a$

There are two different parse tree for same string. Therefore, grammar is ~~context-free~~ ambiguous.

$$\text{11. } \langle S \rangle \rightarrow \langle A \rangle a \langle B \rangle b \\ \langle A \rangle \rightarrow \langle A \rangle b \mid b \\ \langle B \rangle \rightarrow a \langle B \rangle \mid a$$

Consider the grammar.

which of the following sentences are in the language generated by this grammar?

- a. baab
- b. bbab
- c. bbaaaaa
- d. bbaab

Answer :- a. baab

$$\begin{aligned} \text{Derivation: } \langle S \rangle &\Rightarrow \langle A \rangle a \langle B \rangle b \\ &\Rightarrow b a \langle B \rangle b \\ &\Rightarrow b a ab. \end{aligned}$$

d. bbaab

$$\begin{aligned} \text{Derivation: } \langle S \rangle &\Rightarrow \langle A \rangle a \langle B \rangle b \\ &\Rightarrow \langle A \rangle b a \langle B \rangle b \\ &\Rightarrow b b a \langle B \rangle b \\ &\Rightarrow bbaab \end{aligned}$$

12. Consider the following grammar:

$$\langle S \rangle \rightarrow a \langle S \rangle c \langle B \rangle \mid \langle A \rangle \mid b$$

$$\langle A \rangle \rightarrow c \langle A \rangle \mid c$$

$$\langle B \rangle \rightarrow d \mid \langle A \rangle$$

Which of the following sentences are in the language generated by this grammar?

- a. ab cd
- b. a ccc bd
- c. a ccc bcc
- d. acd
- e. accc

Answer: a. ab cd

$$\begin{aligned}\text{Derivation: } S &\Rightarrow a \langle S \rangle c \langle B \rangle \\ &\Rightarrow a b c \langle B \rangle \\ &\Rightarrow a b c d.\end{aligned}$$

$$\begin{aligned}e. \text{ acccc} &\\ S &\Rightarrow a \langle S \rangle c \langle B \rangle \\ &\Rightarrow a \langle A \rangle c \langle B \rangle \\ &\Rightarrow a c c \langle B \rangle \\ &\Rightarrow a c c d \text{ (A) } \\ &\Rightarrow a c c c \text{ (B)}\end{aligned}$$

13. Write a grammar for the language consisting of strings that have n copies of the letter a followed by the same number of copies of the letter b, where $n > 0$. For example, the strings ab, aaaaabbb and aaaaaaaabb are in the language, but a, abb, ba and aabb are not.

Answer: Grammar: $S \rightarrow a S b \mid AB$
 $A \rightarrow a$
 $B \rightarrow b$

14. Draw parse trees for the sentences aabb and

aabb, as derived from the grammar of
Problem 13.

Answers: ① a a b b

Derivation:

$$\begin{aligned}
 S &\rightarrow aSb \\
 &\Rightarrow aABb \\
 &\Rightarrow AaBb \\
 &\Rightarrow aab\ b
 \end{aligned}$$

Parse Tree

```

graph TD
    S --> A
    S --> B
    A --> a
    B --> b
  
```

② aaaa bbbb

Derivation : $S \Rightarrow aSb$

$\rightarrow \underline{a a S b b}$

$$\Rightarrow \text{AaASbbB}$$

$$\Rightarrow \text{AaaABbbB}$$

$\Rightarrow \text{aaaaBbbb}$

$\Rightarrow \overline{aaaabbhb}$

Parse Tree:

