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 (Grammar)

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

Answer:-

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

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

Answer :-

Answer: <assign> → <id> = <expr>
<expr> → <expr> + <term>
| <term>
<term> → <id> | <post_increment> |
<post_decrement> | <pre_increment>
| <pre_decrement>
<post_increment> → <id> ++
<post_decrement> → <id> --
<pre_increment> → ++ <id>
<pre_decrement> → -- <id>
<id> → A | B | C

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

a. A = A * (B + (C * A))
b. B = C * (A * C + B)
c. A = A * (B + (C))

Ex 3.2: <assign> → <id> = <expr>
<id> → A | B | C
<expr> → <id> + <expr>
| <id> * <expr>
| (<expr>)
| <id>

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

<assign> ⇒ <id> = <expr>

⇒ A = <expr>

⇒ A = <id> * <expr>

⇒ A = A * <expr>

⇒ A = A * (<expr>)

⇒ A = A * (<id> + <expr>)

⇒ A = A * (B + <expr>)

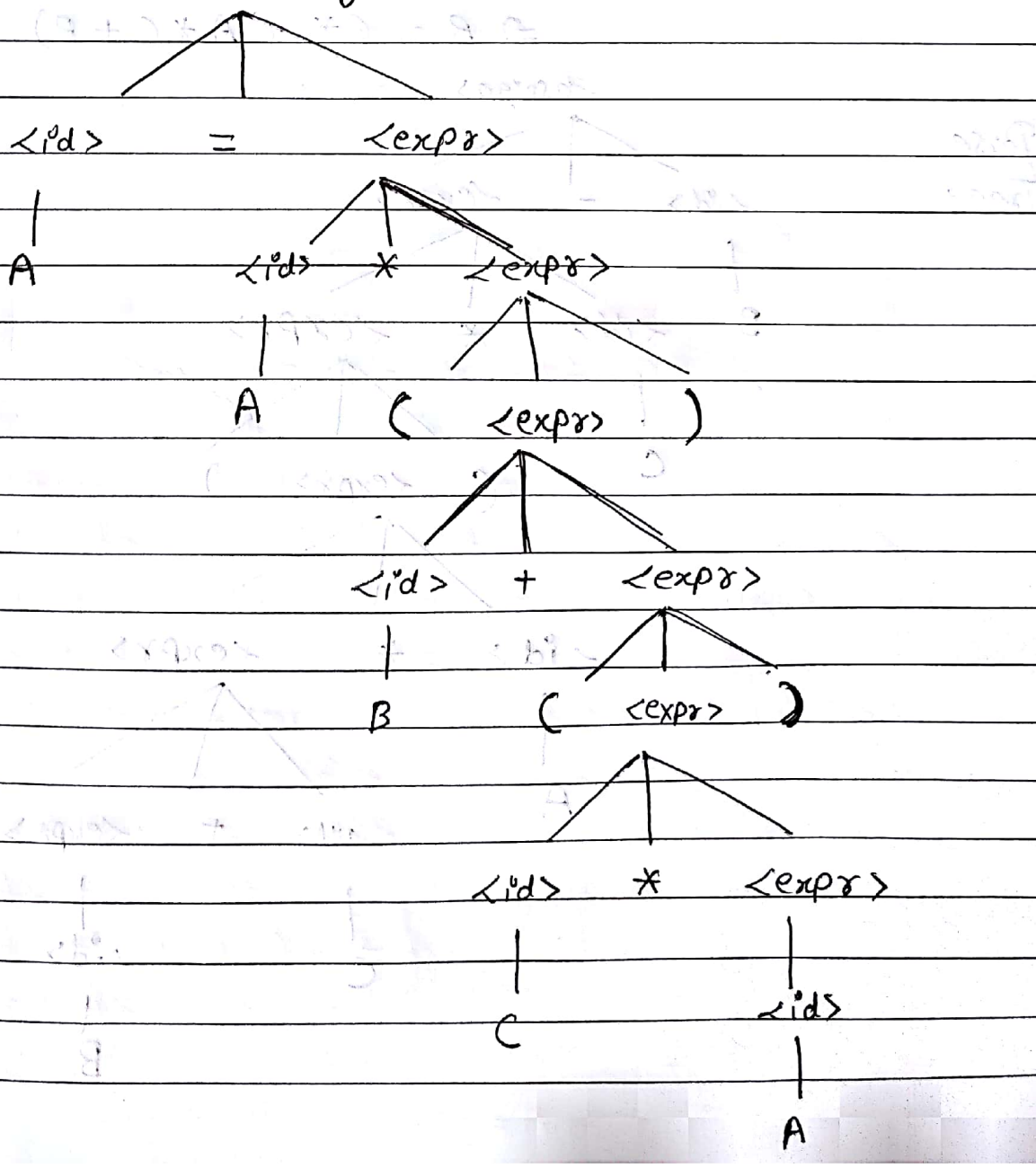
⇒ A = A * (B + (<expr>))

⇒ A = A * (B + (<id> * <expr>))

⇒ A = A * (B + (C * <id>))

⇒ A = A * (B + (C * A))

Parse Tree:

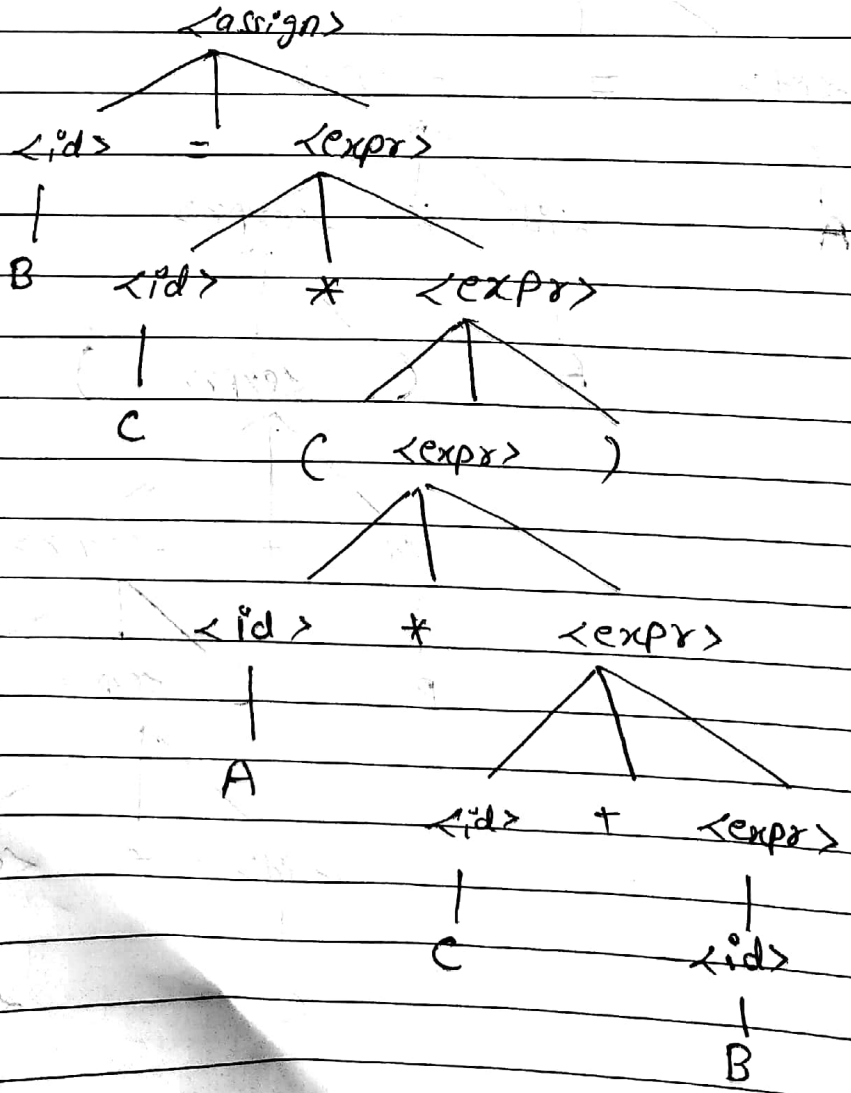


b. B = C * (A * C + B)

Leftmost Derivation:

<assign> ⇒ <id> = <expr>
⇒ B = <expr>
⇒ B = <id> * <expr>
⇒ B = C * <expr>
⇒ B = C * (<expr>)
⇒ B = C * (<id> * <expr>)
⇒ B = C * (A * <expr>)
⇒ B = C * (A * <id> + <expr>)
⇒ B = C * (A * C + <expr>)
⇒ B = C * (A * C + <id>)
⇒ B = C * (A * C + B)

Parse Tree:



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

Leftmost derivation:-

$\langle assign \rangle \Rightarrow \langle id \rangle = \langle expr \rangle$

$A = \langle expr \rangle$

$A = \langle id \rangle * \langle expr \rangle$

$A = A * \langle expr \rangle$

$A = A * (\langle expr \rangle)$

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

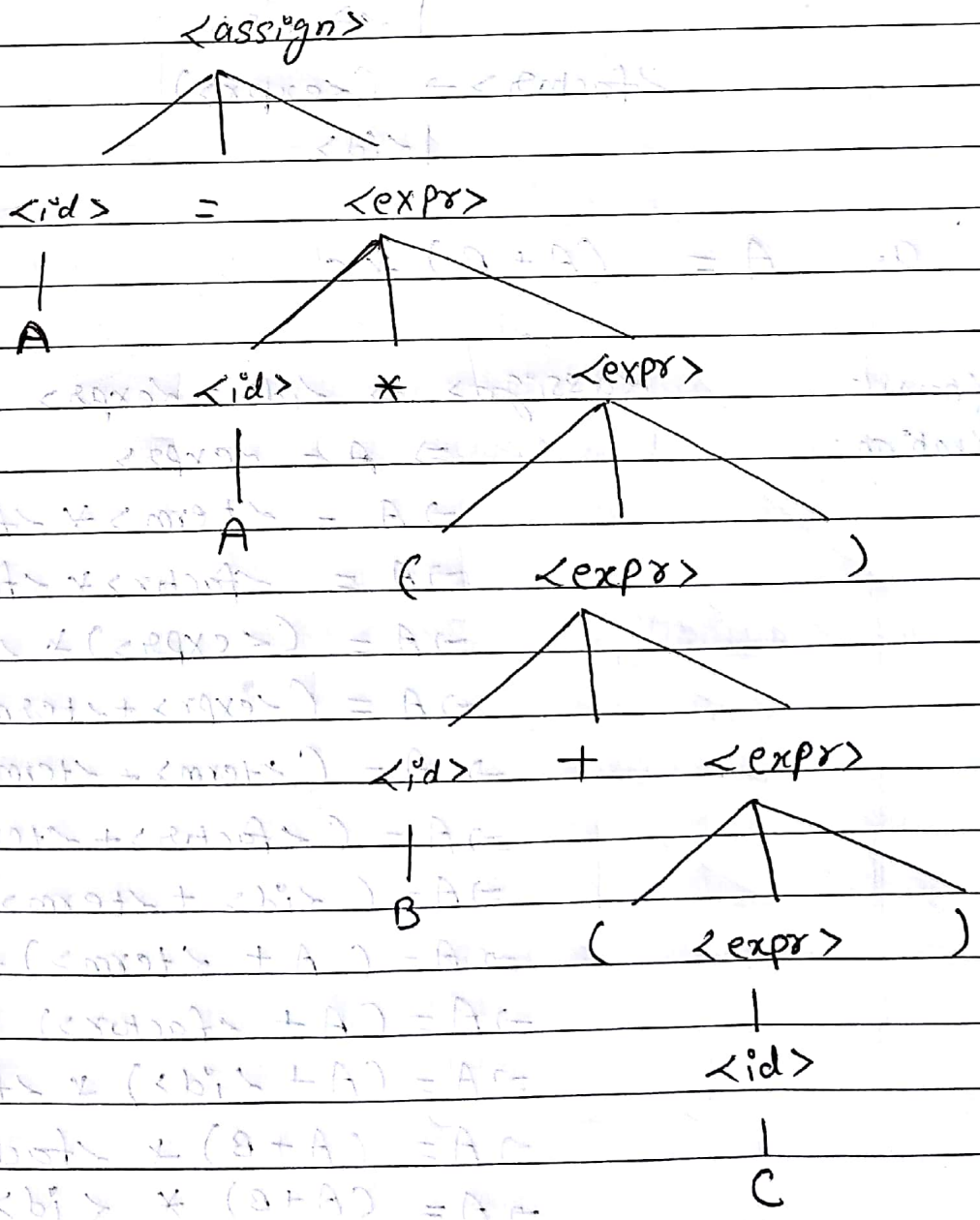
$A = A * (B + \langle expr \rangle)$

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

$A = A * (B + (\langle id \rangle))$

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

Parse Tree:-



Date / /
Page No.
Shivalal

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 <assign> → <id> = <expr>

<id> → A | B | C

<expr> → <expr> + <term>
| <term>

<term> → <term> * <factor>
| <factor>

<factor> → (<expr>)
| <id>

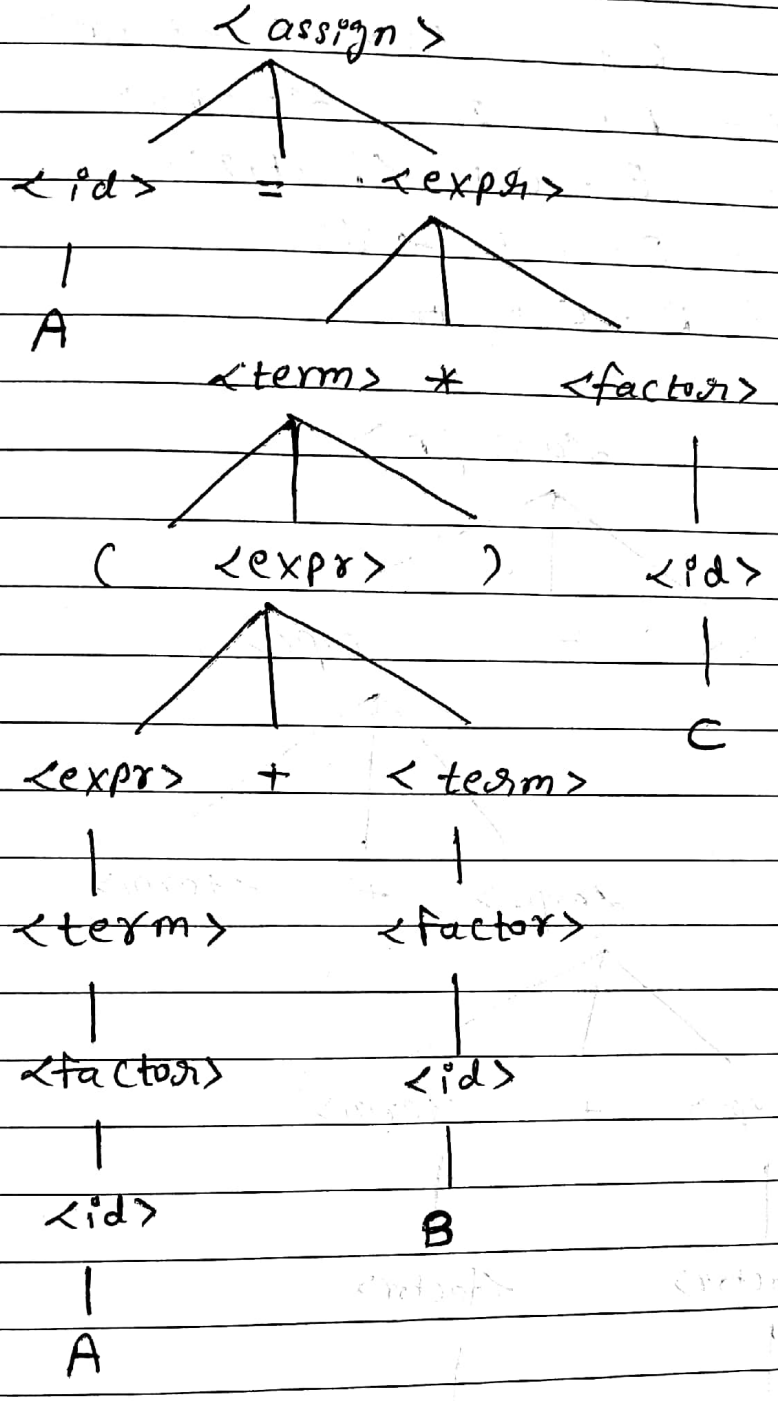
a. A = (A+B) * C

Leftmost Derivation:

<assign> ⇒ <id> = <expr>
⇒ A = <expr>
⇒ A = <term> * <factor>
⇒ A = <factor> * <factor>
⇒ A = (<expr>) * <factor>
⇒ A = (<expr> + <term>) * <factor>
⇒ A = (<term> + <term>) * <factor>
⇒ A = (<factor> + <term>) * <factor>
⇒ A = (<id> + <term>) * <factor>
⇒ A = (A + <term>) * <factor>
⇒ A = (A + <factor>) * <factor>
⇒ A = (A + <id>) * <factor>
⇒ A = (A + B) * <factor>
⇒ A = (A + B) * <id>

⇒ A = (A+B) * C

Parse Tree:



b. A = B + C + A

Leftmost Derivation:

<assign> ⇒ <id> = <expr>
⇒ A = <expr>
⇒ A = <expr> + <term>
⇒ A = <expr> + <term> + <term>
⇒ A = <term> + <term> + <term>
⇒ A = <factor> + <term> + <term>
⇒ A = <id> + <term> + <term>

$\Rightarrow$ A = B + <term> + <term>

$\Rightarrow$ A = B + <factor> + <term>

$\Rightarrow$ A = B + <id> + <term>

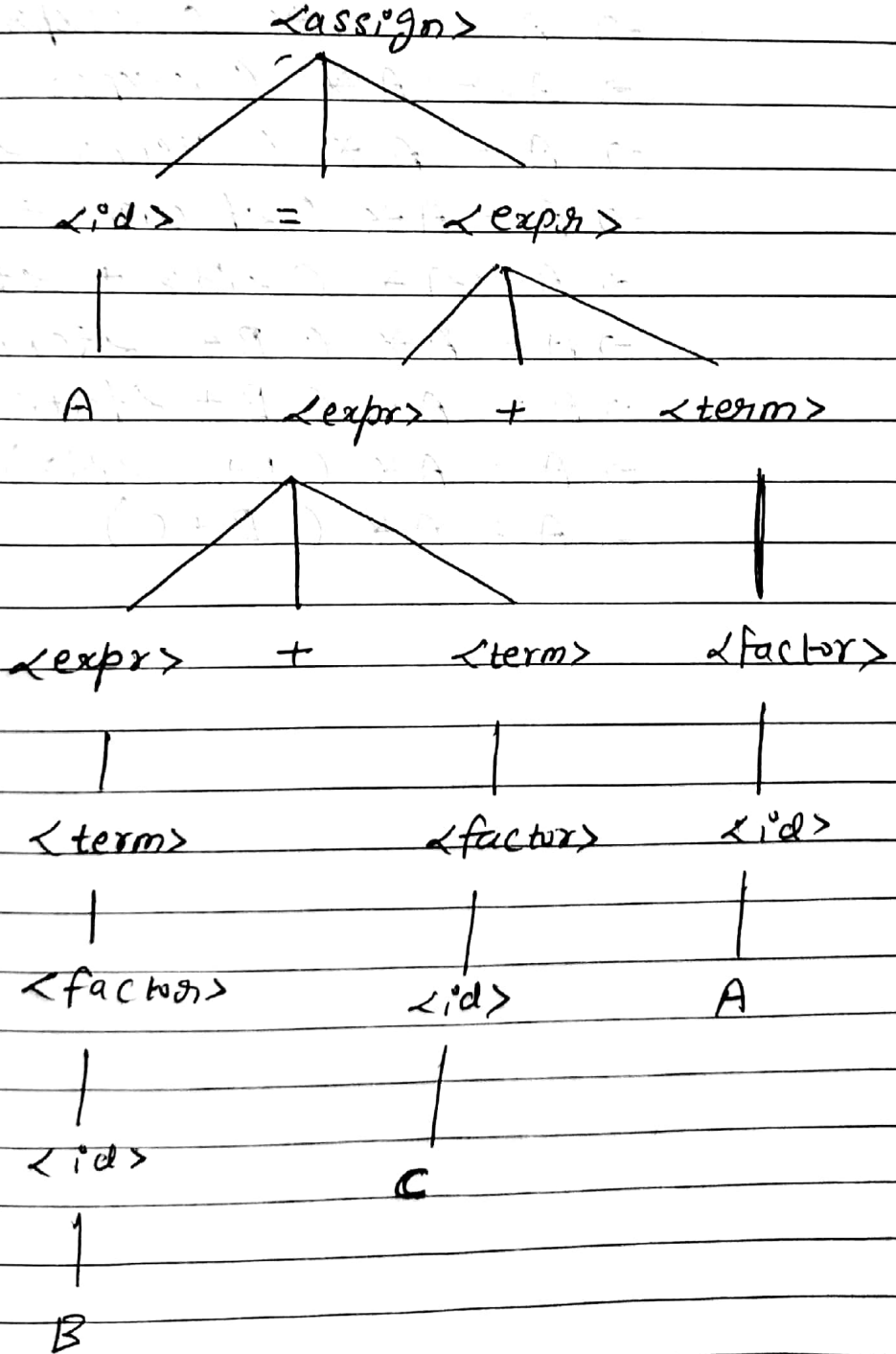
$\Rightarrow$ A = B + C + <term>

$\Rightarrow$ A = B + C + <factor>

$\Rightarrow$ A = B + C + <id>

$\Rightarrow$ A = B + C + A

Parse Tree :-



Date / /
Page No.

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

Leftmost Derivation:

$\langle assign \rangle \Rightarrow \langle id \rangle = \langle expr \rangle$

$\Rightarrow A = \langle expr \rangle$

$\Rightarrow A = \langle term \rangle * \langle factor \rangle$

$\Rightarrow A = \langle factor \rangle * \langle factor \rangle$

$\Rightarrow A = \langle id \rangle * \langle factor \rangle$

$\Rightarrow A = A * \langle factor \rangle$

$\Rightarrow A = A * (\langle expr \rangle)$

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

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

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

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

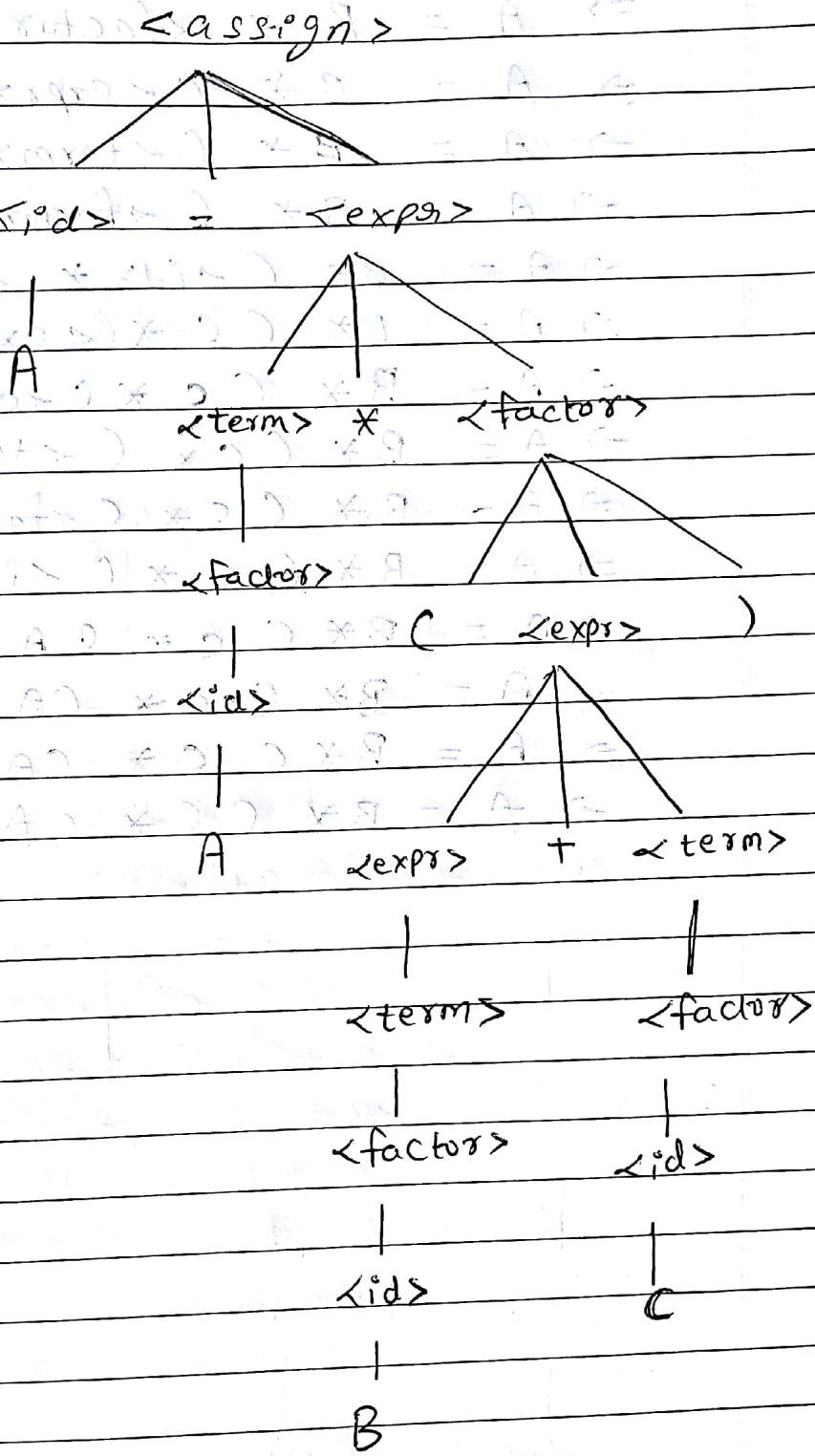
$\Rightarrow A = A * (B + \langle term \rangle)$

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

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

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

Parse Tree :-



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

Leftmost Derivation :

<assign> ⇒ <id> = <expr>
⇒ A = <expr>
⇒ A = <term> * <factor>
⇒ A = <factor> * <factor>
⇒ A = <id> * <factor>

$\Rightarrow A = B * \langle factor \rangle$

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

$\Rightarrow A = B * (\langle term \rangle * \langle factor \rangle)$

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

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

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

$\Rightarrow A = B * (C * (\langle expr \rangle + \langle term \rangle))$

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

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

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

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

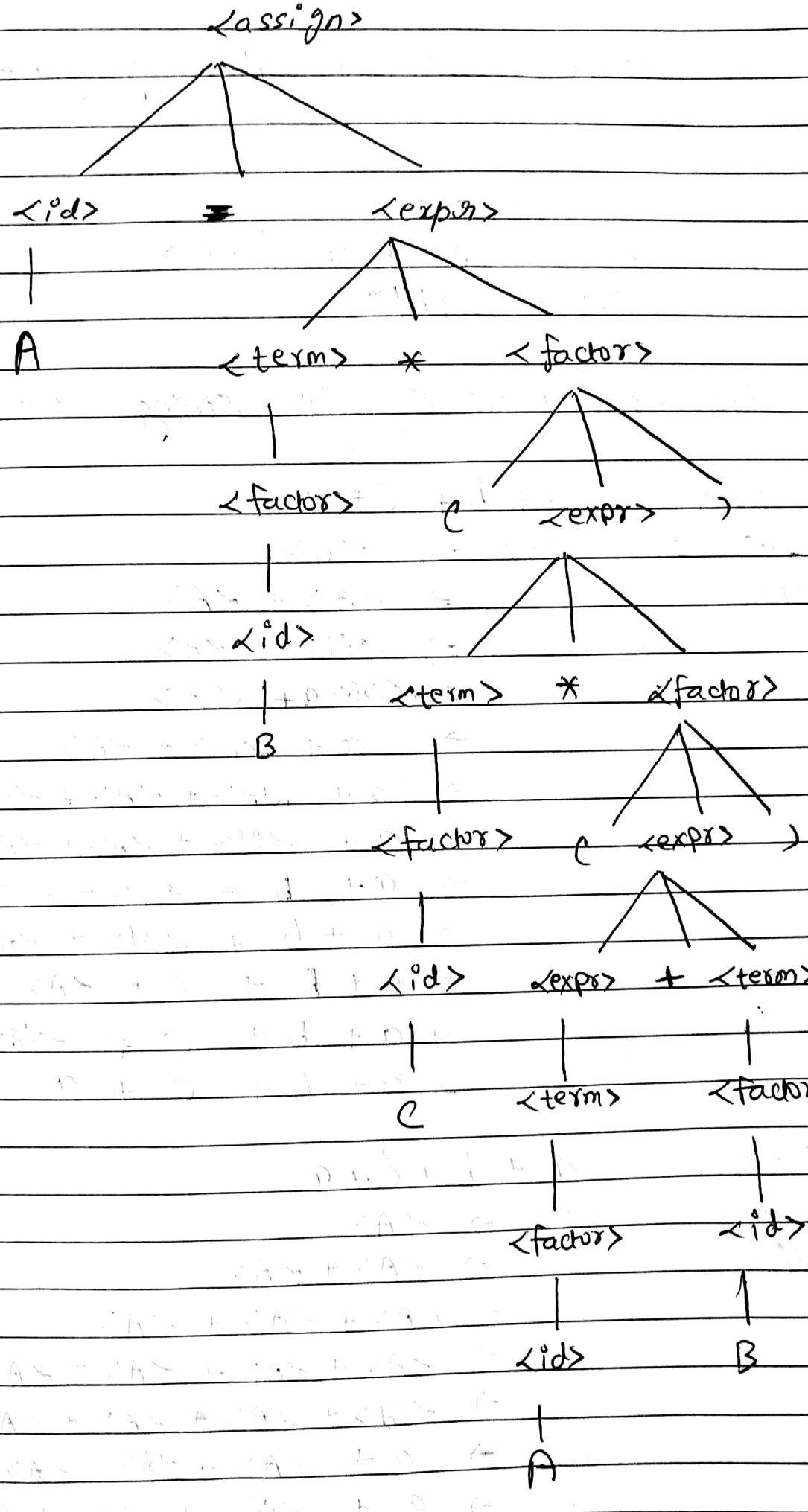
$\Rightarrow A = B * (C * (A + \langle factor \rangle))$

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

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

Date / /
Page No.
Shivalal

Parse Tree:



8. Prove that the following grammar is ambiguous:

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

$\langle A \rangle \rightarrow \langle A \rangle + \langle A \rangle \mid \langle id \rangle$

$\langle id \rangle \rightarrow a \mid b \mid c$

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

Leftmost Derivation (i):

$a + b + c + a$

$\langle S \rangle \Rightarrow \langle A \rangle$

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

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

$\Rightarrow a + \langle A \rangle$

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

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

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

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

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

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

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

$\Rightarrow a + b + c + a$

Leftmost Derivation (ii)

$a + b + c + a$

$\langle S \rangle \Rightarrow \langle A \rangle$

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

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

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

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

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

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

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

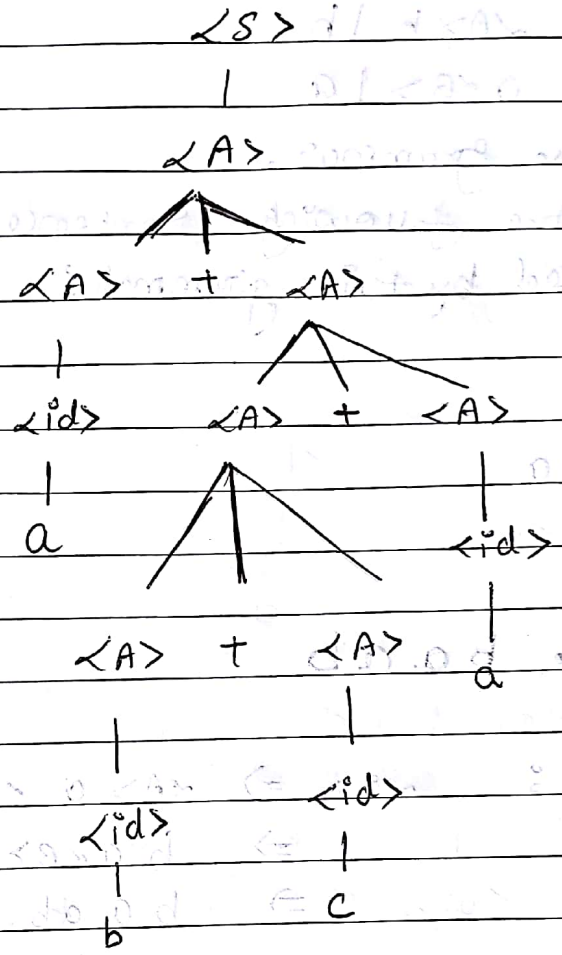
$\Rightarrow a + b + \langle id \rangle + \langle A \rangle$

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

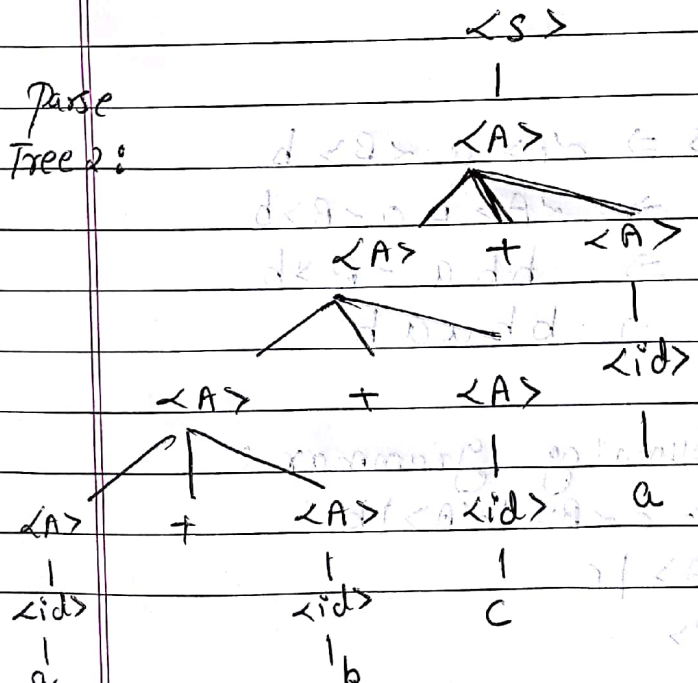
$\Rightarrow a + b + c + \langle id \rangle$

$\Rightarrow a + b + c + a$

Parse Tree 1:



Parse Tree 2:



There are two different parse tree for same String. Therefore, grammar is ambiguous.

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. bb aaab

Answer :- **a. baab**

Derivation : $\langle S \rangle \Rightarrow \langle A \rangle a \langle B \rangle b$
$\Rightarrow b a \langle B \rangle b$
$\Rightarrow b a ab.$

d. bbaab

Derivation : $\langle S \rangle \Rightarrow \langle A \rangle a \langle B \rangle b$
$\Rightarrow \langle A \rangle b a \langle B \rangle b$
$\Rightarrow bb a \langle B \rangle b$
$\Rightarrow bbaab$

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. abcd
b. acccbd
c. acccbcc
d. acd
e. accc

Answer: a. abcd

Derivation: $S \Rightarrow a\langle S\rangle c\langle B\rangle$
$\Rightarrow abC\langle B\rangle$
$\Rightarrow abcd.$

e. accc

$S \Rightarrow a\langle S\rangle c\langle B\rangle$
$\Rightarrow a\langle A\rangle c\langle B\rangle$
$\Rightarrow acc\langle B\rangle$
$\Rightarrow acc\langle A\rangle$
$\Rightarrow accc$

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, aaaabbbb, and aaaaaaaabbbbbbbb are in the language, but a, abb, ba and aabb are not.

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

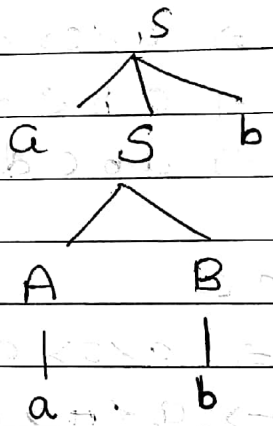
14. Draw parse trees for the sentences aabb and

aaaabbbb, as derived from the grammar of Problem 13.

Answer: ① aabb

Derivation: $S \Rightarrow aSb$
$\Rightarrow aABb$
$\Rightarrow aaBb$
$\Rightarrow aabb$

Parse Tree:



② aaaabbbb

Derivation: $S \Rightarrow aSb$
$\Rightarrow aaSbb$
$\Rightarrow aaaSbbb$
$\Rightarrow aaaABbbb$
$\Rightarrow aaaaBbbb$
$\Rightarrow aaaabbbb$

Parse Tree: