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

A) Leftmost Derivation

<assign> → <id> = <expr>

→ A = <expr>

→ A = <expr> + <term>

→ A = <term> + <term>

→ A = <factor> + <term>

→ A = <id> + <term>

→ A = A + <term>

→ A = A + <factor>

→ A = A + (<expr>)

→ A = A + (<term>)

→ A = A + (<term> * <factor>)

→ A = A + (<factor> * <factor>)

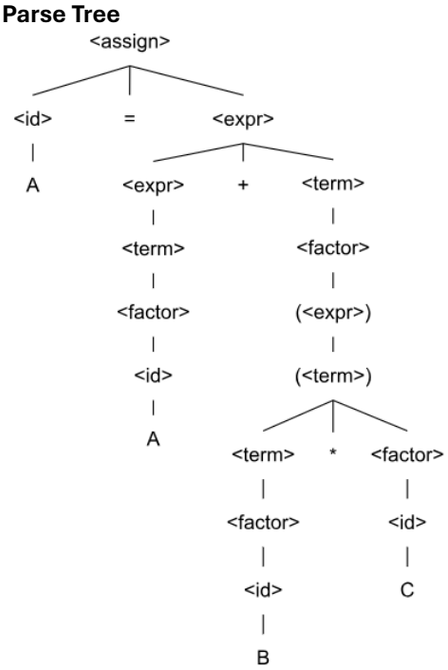
→ A = A + (<id> * <factor>)

→ A = A + (B * <factor>)

→ A = A + (B * <id>)

→ A = A + (B * C)

B) Parse Tree



2. B = (A * C) + B

A) Leftmost Derivation

<assign> → <id> = <expr>

→ B = <expr>

→ B = <expr> + <term>

→ B = <term> + <term>

→ B = <factor> + <term>

→ B = (<expr>) + <term>

→ B = (<term>) + <term>

→ B = (<term> * <factor>) + <term>

→ B = (<factor> * <factor>) + <term>

→ B = (<id> * <factor>) + <term>

→ B = (A * <factor>) + <term>

→ B = (A * <id>) + <term>

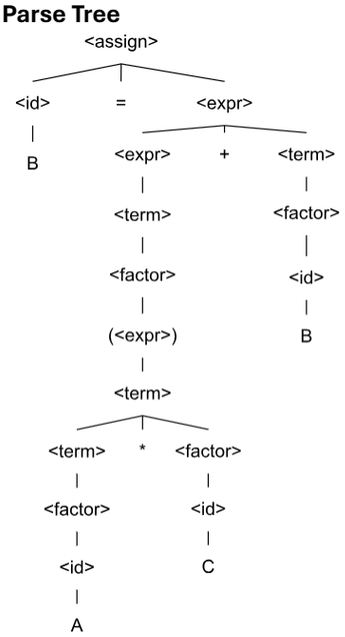
→ B = (A * C) + <term>

→ B = (A * C) + <factor>

→ B = (A * C) + <id>

→ B = (A * C) + B

B) Parse Tree



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

A) Leftmost Derivation

<assign> → <id> = <expr>

→ A = <expr>

→ A = <term> * <factor>

→ A = <factor> * <factor>

→ A = (<expr>) * <factor>

→ A = (<term>) * <factor>

→ A = (<term> * <factor>) * <factor>

→ A = (<factor> * <factor>) * <factor>

→ A = ((<expr>) * <factor>) * <factor>

→ A = ((<expr> + <term>) * <factor>) * <factor>

→ A = ((<term> + <term>) * <factor>) * <factor>

→ A = ((<factor> + <term>) * <factor>) * <factor>

→ A = ((<id> + <term>) * <factor>) * <factor>

→ A = ((C + <term>) * <factor>) * <factor>

→ A = ((C + <factor>) * <factor>) * <factor>

→ A = ((C + <id>) * <factor>) * <factor>

→ A = ((C + A) * <factor>) * <factor>

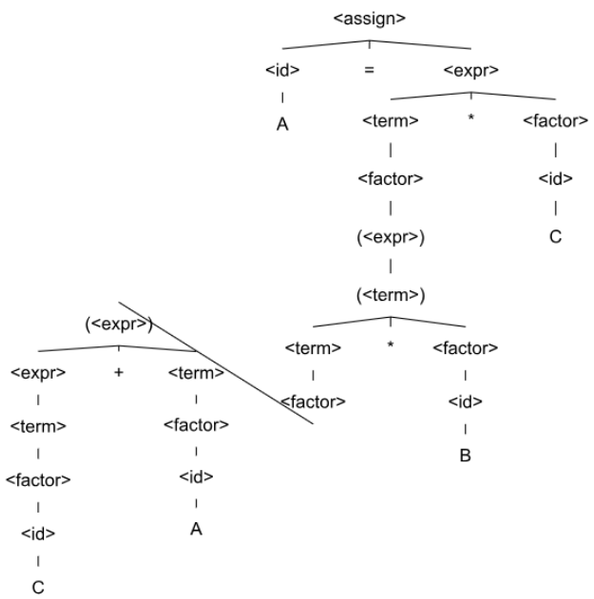
→ A = ((C + A) * <id>) * <factor>

→ A = ((C + A) * B) * <factor>

→ A = ((C + A) * B) * <id>

→ A = ((C + A) * B) * C

Parse Tree



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

A) Leftmost Derivation

<assign> → <id> = <expr>

→ C = <expr>

→ C = <term>

→ C = <term> * <factor>

→ C = <factor> * <factor>

→ C = (<expr>) * <factor>

→ C = (<expr> + <term>) * <factor>

→ C = (<term> + <term>) * <factor>

→ C = (<factor> + <term>) * <factor>

→ C = (<id> + <term>) * <factor>

→ C = (A + <term>) * <factor>

→ C = (A + <factor>) * <factor>

→ C = (A + <id>) * <factor>

→ C = (A + B) * <factor>

→ C = (A + B) * (<expr>)

→ C = (A + B) * (<expr> + <term>)

→ C = (A + B) * (<term> + <term>)

→ C = (A + B) * (<factor> + <term>)

→ C = (A + B) * (<id> + <term>)

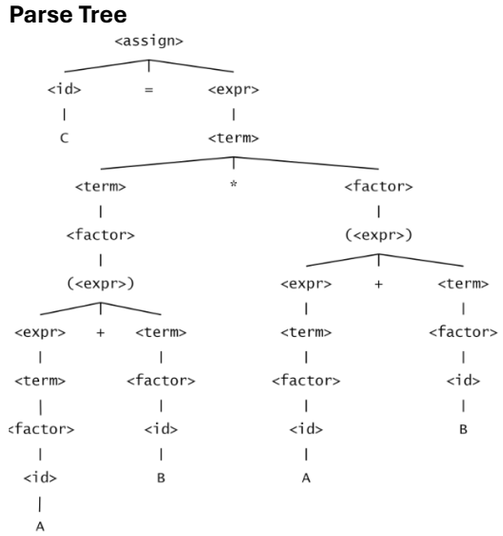
→ C = (A + B) * (A + <term>)

→ C = (A + B) * (A + <factor>)

→ C = (A + B) * (A + <id>)

→ C = (A + B) * (A + B)

B) Parse Tree



5. A = A * (B + C)

A) Leftmost Derivation

<assign> → <id> = <expr>

→ A = <expr>

→ A = <term>

→ A = <term> * <factor>

→ A = <factor> * <factor>

→ A = <id> * <factor>

→ A = A * <factor>

→ A = A * (<expr>)

→ A = A * (<expr> + <term>)

→ A = A * (<term> + <term>)

→ A = A * (<factor> + <term>)

→ A = A * (<id> + <term>)

→ A = A * (B + <term>)

→ A = A * (B + <factor>)

→ A = A * (B + <id>)

→ A = A * (B + C)

B) Parse Tree