Using the following grammar, show whether it is possible to generate a parse tree for the statements given. If so, show its leftmost derivation. `<assign> -> <id> = <expr>`

`<id> -> A | B | C`

`<expr> -> <expr> + <term> | <term>`

`<term> -> <term> * <factor> | <factor>`

`<factor> -> ( <expr> ) | <id>`

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

Answer:

1. <assign> -> <id> = <expr>

   -> A = <expr> + <term>

   -> A = <term> + <term> * <factor>

   -> A = <term> * <factor> + <factor> * <id>

   -> A = <factor> * <id> + <id> * A

   -> A = <id> * B + C * A

   -> A = A * B + C * A

2. <assign> -> <id> = <expr>

   -> A = <expr> + <term>

   -> A = <term> + <term>

   -> A = <factor> + <term> * <factor>

   -> A = <id> + <factor> * (<expr>)

   -> A = B + <id> * (<expr> + <term>)

   -> A = B + C * (<term> + <factor>)

   -> A = B + C * (<factor> + <id>)

   -> A = B + C * (<id> + B)

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