NORTH-SOUTH UNIVERSITY

Home-Work

Topic: Grammars & Derivations

Course Code: CSE-425

Sec: 1

Submitted by: SYED MUSTAVI MAHEEN

ID# 1611207042

BNF: Backus-Naur Form

- A notation technique for Context-Free Grammar.
- Used to describe the syntax of languages used in computing.
- Have 2 sides, Left-Hand Side & Right-Hand Side
- In Left-Hand Side, there can only be Non-Terminals.
- In Right-Hand Side, there can be Non-Terminals/ Lexemes / Tokens.
- L.H.S + R.H.S = Grammar Rule

Example

<program> → begin <stmt_list> end

<stmt_list> → <stmt> | <stmt>; <stmt_list>

<stmt> → <var> = <expression>

<expression> → <var> + <var> | <var> - <var> | <var>

Derivations

A derivation is basically a sequence of production rules, in order to get the input string. During parsing, we take two decisions for some sentential form of input:

- Deciding the non-terminal which is to be replaced.
- Deciding the production rule, by which, the non-terminal will be replaced.

Example

begin A = B + C; B = C end

<program> → begin <stmt_list> end

begin <stmt>; <stmt_list> end

begin <var> = <expression>; <stmt_list> end

begin A = <expression>; <stmt_list> end

begin A = <var> + <var>; <stmt_list> end

begin A = B + <var>; <stmt_list> end

begin A = B + C; <stmt_list> end

begin A = B + C; <stmt> end

begin A = B + C; <var> = <expression> end

begin A = B + C; B = <expression> end

begin A = B + C; B = <var> end

begin A = B + C; B = C end

To decide which non-terminal to be replaced with production rule, we can have two options:

Left-most Derivation

If the sentential form of an input is scanned and replaced from left to right, it is called left-most derivation. The sentential form derived by the left-most derivation is called the left-sentential form.

Right-most Derivation

If we scan and replace the input with production rules, from right to left, it is known as right-most derivation. The sentential form derived from the right-most derivation is called the right-sentential form.

Sample Grammar

<exp> → <exp> + <exp> | <exp> * <exp> | <exp> | <number>

<number> → <number> <digit> | <digit>

<digit> → 0 | 1 | 2 | 3 | 4

Show 234 using Left-Most Derivation and Right-Most Derivation.

Left-Most Derivation:

<number> → <number> <digit>

→ <number> <digit> <digit>

→ <digit> <digit> <digit>

→ 2 <digit> <digit>

→ 23 <digit>

→ 234

Right-Most Derivation:

  <number> → <number> <digit>

     → <number> 4

     → <number> <digit> 4

     → <number> 34

     → <digit> 34

     → 234

Practice Grammar

1. <assgin> → <id> = <expr>
   <id>   → A | B | C
   <expr>   → <id> + <expr> | <id> * <expr> | (<expr>) | <id>

Show A = B * (A+C) using Left-Most Derivation.

Solution:

  <assgin> → <id> = <expr>

     → A = <expr>

     → A = <id> * <expr>

     → A = B * <expr>

     → A = B * (<expr>)

     → A = B * (<id> + <expr>)

     → A = B * (A + <expr>)

     → A = B * (A + <id>)

     → A = B * (A + C)

Show A = A * (B + (C * A)) using Left-Most Derivation.

Solution:

        <assgin> → <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 * <expr>))

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

              → A = A * (B + (C * A))

2. Roll → <ROLL> Class Studs </ROLL>
   Class → <CLASS> Text </CLASS>
   Text → Char Text
   Text → Char
   Char → a… (other chars)
   Studs → Stud Studs | Stud
   Stud → <STUD> TEXT </STUD>

Generate documents such as: <ROLL><CLASS>cs154</CLASSS>

                        <STUD>Sally</STUD>

                        <STUD>Fred</STUD>

                        </ROLL>

Try it yourself to practice.