Forma Backus naur

Lopez Manriquez Angel 2CM5

17 de junio de 2019

1. Introducción

La notación de Backus-Naur (BNF) es un metalenguaje, es decir, es una nomenclatura que se usa para para expresar gramáticas libres de contexto, por lo que se dice que es una manera formal de describir lenguajes formales, usando ciertos símbolos y reglas. Permite una descripción compacta y precisa de los constructores sintácticos.

Es usada frecuentemente para describir la sintaxis de los lenguajes usados en computación, como lenguajes de programación, formatos para documentos, conjuntos de instrucciones y protocolos de comunicación.

2. Descripción

Esta notación contiene dos tipos de símbolos: terminales y no terminales. Los terminales denotan lexemas, mientras que los no terminales representan construcciones sintácticas, como las expresiones. La parte principal de esta notación es un conjunto finito de reglas de producción, las cuales tienen la forma:

$$A \to x_1 | \ldots | x_n$$

En donde A es un no terminal. La parte de la izquierda representa la construcción sintáctica que esta regla de producción define, y cada elemento x_i de la parte derecha es una palabra que consiste de terminales y no terminales. Las palabras $x_1, x_2, \ldots, x_n$ representan las n definiciones alternativas de A.

Los metasímbolos de la notación BNF son:

- `::=` Definición (el esquema de la derecha desarrolla al elemento de la izquierda).
- `|` Alternativa (se puede elegir únicamente uno de los elementos que separa).
- {} Repetición (los elementos que incluyen, pueden repetirse cero o más veces).
- `[]` Opción (los elementos que incluyen pueden utilizarse o no)
- `()` Agrupación (sirven para agrupar los elementos que incluyen).

Para expresar una regla de producción, usamos `<símbolo>::= expresión`, donde `símbolo` es un no terminal y `expresión` consiste de una o más secuencias de símbolos, separadas por `|` si hay más de una opción.

3. Ejemplos

- <símbolo_básico>::= <letras>| <dígitos>| <booleano>| <delimitador>
- <letras>::= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
- <dígitos>::= 0|1|2|3|4|5|6|7|8|9
- <booleano>::= #TRUE# | #FALSE#
- <delimitador>::= <operador>| <separador>| <bracket>| <declarador>| <especificador>
- <operadores>::= <aritmético>| <relacional>| <lógico>| <secuencial>
- <aritmético>::= + | - | * | / | // | ** | ^
- <relacional>::= <| = | >| ~=
- <secuencial>::= #GO TO# | #IF# | #THEN# | #ELSE# | #FOR# | #DO#
- <separador>::= # | , | . | : | ; | := | #UNTIL# | #WHILE# | #COMMENT# | #CODE#
- <bracket>::=) | (|] | [| #(# | #)# | #BEGIN# | #END#

La sintaxis de la notación BNF puede ser descrita por sí misma:

```
 <syntax>         ::= <rule> | <rule> <syntax>
 <rule>           ::= <opt-whitespace> "<" <rule-name> ">" <opt-whitespace> "::=" <opt-whitespace>
<expression> <line-end>
 <opt-whitespace> ::= " " <opt-whitespace> | ""
 <expression>     ::= <list> | <list> <opt-whitespace> "|" <opt-whitespace> <expression>
 <line-end>       ::= <opt-whitespace> <EOL> | <line-end> <line-end>
 <list>           ::= <term> | <term> <opt-whitespace> <list>
 <term>           ::= <literal> | "<" <rule-name> ">"
 <literal>        ::= '"' <text1> '"' | "'" <text2> "'"
 <text1>          ::= "" | <character1> <text1>
 <text2>          ::= "" | <character2> <text2>
 <character>      ::= <letter> | <digit> | <symbol>
 <letter>         ::= "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" | "L" | "M"
| "N" | "O" | "P" | "Q" | "R" | "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z" | "a" | "b" | "c" |
"d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" |
"t" | "u" | "v" | "w" | "x" | "y" | "z"
 <digit>          ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
 <symbol>         ::=  "|" | " " | "-" | "!" | "#" | "$" | "%" | "&" | "(" | ")" | "*" | "+" | ","
| "-" | "." | "/" | ":" | ";" | ">" | "=" | "<" | "?" | "@" | "[" | "\" | "]" | "^" | "_" | "`" |
"{" | "}" | "~"
 <character1>     ::= <character> | "'"
 <character2>     ::= <character> | '"'
 <rule-name>      ::= <letter> | <rule-name> <rule-char>
 <rule-char>      ::= <letter> | <digit> | "-"
```