BACKUS NAUR

BACKUS NAUR

Andrés Felipe Henao Clavijo

*INGENIERIA DE SISTEMAS Y COMPUTACION UNIVERSIDAD TECNOLOGICA DE PEREIRA*

andres.henao1@utp.edu.co

***Resumen*— La notación de Backus-Naur, también conocida por sus denominaciones inglesas Backus-Naur form (BNF), Backus-Naur formalism o Backus normal form, es un metalenguaje usado para expresar gramáticas libres de contexto: es decir, una manera formal de describir lenguajes formales.**

**El BNF se utiliza extensamente como notación para las gramáticas de los lenguajes de programación, de los sistemas de comando y de los protocolos de comunicación, así como una notación para representar partes de las gramáticas de la lengua natural (por ejemplo, el metro en la poesía de Venpa). La mayoría de los libros de textos para la teoría o la semántica del lenguaje de programación documentan el lenguaje de programación en BNF.**

***Palabras clave—* matemáticas, razonamiento, variable, función, recursividad, axioma, inductivo**

***Abstract*— The Backus-Naur notation, also known by its English denominations Backus-Naur form (BNF), Backus-Naur formalism or Backus normal form, is a metalanguage used to express context-free grammars: that is, a formal way of describing languages formal.**

**The BNF is widely used as a notation for the grammars of programming languages, command systems and communication protocols, as well as a notation to represent parts of the grammars of the natural language (for example, the meter in the poetry of Venpa). Most textbooks for the theory or semantics of programming language document the programming language in BNF.**

1. INTRODUCCIÓN

Una especificación de BNF es un sistema de reglas de derivación, escrito como:

<simbolo> ::= <expresión con símbolos>

donde <símbolo> es un no terminal, y la expresión consiste en secuencias de símbolos o secuencias separadas por la barra vertical, '|', indicando una opción, el conjunto es una posible substitución para el símbolo a la izquierda. Los símbolos que nunca aparecen en un lado izquierdo son terminales.

1. CONTENIDO

¿Qué es el formalismo Backus-Naur?

La sintaxis de la línea de comandos de QlikView y la sintaxis del script se describen en una notación denominada formalismo Backus-Naur o código BNF.

La tabla siguiente ofrece una lista de símbolos que se utilizan en el código BNF, con una descripción de su interpretación:

| OR lógico: el símbolo puede emplearse a ambos lados.

( ) Paréntesis que definen la precedencia: se sirven para estructurar la sintaxis de BNF.

[ ] Los corchetes indican que los elementos que encierran son opcionales.

{ } Llaves: los elementos incluidos entre llaves pueden repetirse ninguna o más veces.

Símbolo Una categoría sintáctica no concluyente que puede dividirse en otros símbolos. Por ejemplo, combinaciones de los de arriba, otros símbolos no concluyentes, cadenas de texto, etc.

::= Marca el comienzo de un bloque que define un símbolo.

LOAD Un símbolo final que consiste en una cadena de texto. Debe escribirse tal cual en el script.

Todos los símbolos terminales se imprimen en estilo negrita. Por ejemplo, “(” debería interpretarse como un paréntesis que define la precedencia, mientras que “(” debería interpretarse como un carácter que ha de imprimirse en el script.

Ejemplo:

La descripción de la sentencia alias es:

alias fieldname as aliasname { , fieldname as aliasname}

Esto debe interpretarse como la cadena de texto "alias", seguida de un nombre de campo arbitrario, seguido por la cadena de texto "as", seguida por un nombre de alias arbitrario. Se puede dar cualquier número de combinaciones adicionales de "fieldname as alias", separadas por comas.

Por ej. las siguientes sentencias son correctas:

alias a as first;

alias a as first, b as second;

alias a as first, b as second, c as third;

Y estas sentencias no son correctas:

alias a as first b as second;

alias a as first { , b as second };

1. CONCLUSIONES

Hay muchas variantes y extensiones de BNF, generalmente por simplicidad y concisión, o para adaptarlo a una aplicación específica. Una característica común de muchas variantes es el uso de operadores de repetición de expresiones regulares como \*y +. La forma extendida de Backus-Naur (EBNF) es común.

Otra extensión común es el uso de corchetes alrededor de elementos opcionales. A pesar de que no está presente en el informe original ALGOL 60 (en vez introducido unos años más tarde en IBM 's PL I / definición), la notación es ahora universalmente reconocida.

La forma Backus-Naur aumentada (ABNF) y la forma Backus-Naur enrutamiento (RBNF) [13] son extensiones comúnmente utilizadas para describir los protocolos del Grupo de trabajo de ingeniería de Internet (IETF) .

Las gramáticas de expresión analizadas se basan en el BNF y las anotaciones de expresión regular para formar una clase alternativa de gramática formal , que es esencialmente de carácter analítico más que generativo .

Muchas de las especificaciones de BNF que se encuentran en línea hoy en día están destinadas a ser leídas por humanos y noson formales. Estos a menudo incluyen muchas de las siguientes reglas y extensiones de sintaxis:

Los elementos opcionales entre corchetes: [<item-x>].

Los elementos existentes 0 o más veces están encerrados entre llaves o con un asterisco ( \*) como <word> ::= <letter> {<letter>}o <word> ::= <letter> <letter>\*respectivamente.

Artículos 1 o más veces existentes se sufijo con un símbolo de adición (plus), +.

Los terminales pueden aparecer en negrita en lugar de cursiva, y no terminales en texto sin formato en lugar de corchetes angulares.

Cuando los elementos se agrupan, se incluyen entre paréntesis simples.