NOTACIÓN DE BACKUS NAUR

BACKUS NAUR

Autor 1: Juan Andrés Arias Tascón

*Ingeniería de Sistemas, Universidad Tecnológica de Pereira*

Correo-e: juanandres.arias@utp.edu.co

***Resumen*— Es un metalenguaje de programación que se ha constituido a lo largo del tiempo en uno de los sistemas de notación técnica más frecuentemente usados en computación. Tal como se asegura en una de sus páginas web, la fuente y creación del mismo, surgió a partir de las teorías lingüísticas de Noam Chomsky. Pero sí Naur tomó el BNF para el desarrollo de las gramáticas de los lenguajes de programación, pero también para representar en un lenguaje formalizado par-tes de las estructuras gramáticas de la lengua natural. Primeramente, John Backus y posteriormente Peter Naur establecieron una sintaxis que (a nuestro juicio) responde a la lógica matemática (básicamente como todo lenguaje) y a una estructura algorítmica. El objetivo fue que esa gramática les permitiera crear una base de instrucción para describir y modelar distintos tipos de textos, a saber: documentos que estén bajo formato, conjunto de instrucciones de programación, protocolos de comunicación, lenguajes de programación, notación para las gramáticas y sintaxis de los lenguajes de programación de la computadora, etc. Por supuesto como todo metalenguaje, el Backus-Naur es artificial, pero se estructura a semejanza de la lengua natural con el objetivo central de que sea implementado en lo que en la lingüística computacional se denomina como gramáticas de libre contexto.**

***Palabras clave— Notación, lenguaje, programación, variable, parámetro, JCUP, JLEX, java***

***Abstract*— It is a programming metalanguage that has been constituted over time in one of the most frequently used systems of technical notation in computing. As stated in one of its web pages, the source and creation of it, arose from the linguistic theories of Noam Chomsky. But Naur did take the BNF for the development of the grammar of programming languages, but also to represent part of the grammar structures of the natural language in a formalized language. First, John Backus and later Peter Naur established a syntax that (in our opinion) responds to mathematical logic (basically like all language) and to an algorithmic structure. The objective was that this grammar would allow them to create an instruction base to describe and model different types of texts, namely: documents that are under format, set of programming instructions, communication protocols, programming languages, notation for grammars and syntax of computer programming languages, etc. Of course, like all metalanguages, Backus-Naur is artificial, but it is structured in a similar way to the natural language with the central objective of being implemented in what in computational linguistics is called free-context grammar.**

***Key Word* —** **Notation, language, programming, variable, parameter, JCUP, JLEX, java**

1. INTRODUCCIÓN

Este trabajo estudia de qué manera se estructura y funciona el Backus-Naur. Por otro lado, tratar de establecer un cuerpo de ejemplos que explique en cuáles campos de investigación se implementa este metalenguaje. Esto se dice porque se le usa para describir otro lenguaje, es una nomenclatura que nos permite una descripción compacta y precisa de los constructores sintácticos usando ciertos símbolos y reglas.

También para diferenciar los elementos simbólicos incluidos en las reglas.

L notación de Backus Naur se utiliza extensamente como notación para las gramáticas de los lenguajes de programación de la computadora, 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 y/o la semántica del lenguaje de programación documentan el lenguaje de programación en BNF. Algunas variantes, tales como la augmented Backus-Naur form (ABNF), tienen su propia documentación.

1. CONTENIDO

Al describir idiomas, la forma Backus-Naur (BNF) es una notación formal para codificar gramáticas destinadas al consumo humano.

Muchos lenguajes de programación, protocolos o formatos tienen una descripción BNF en sus especificaciones.

Cada regla en forma Backus-Naur tiene la siguiente estructura:

name :: = expansion

El símbolo :: = significa "puede expandirse en" y "puede reemplazarse por".

En algunos textos, name también se llama non-terminal symbol.

Por ejemplo, en BNF, la gramática de expresión clásica es:

<expr> ::= <term> "+" <expr>

| <term>

<term> ::= <factor> "\*" <term>

| <factor>

<factor> ::= "(" <expr> ")"

| <const>

<const> ::= integer

Naturalmente, podemos definir una gramática para las reglas en BNF:

rule → name ::= expansion

name → < identifier >

expansion → expansion expansion

expansion → expansion | expansion

expansion → name

expansion → terminal

Podríamos definir identificadores utilizando la expresión regular [-A-Za-z\_0-9] +.

Un terminal puede ser un literal citado (como "+", "switch" o "<< =") o el nombre de una clase de literales (como un entero).

El nombre de una clase de literales generalmente se define por otros medios, como una expresión regular o incluso una prosa.

Opción

En EBNF, los corchetes alrededor de una expansión, [expansión], indican que esta expansión es opcional.

Por ejemplo, la regla:

  <term> :: = ["-"] <factor>

permite que se nieguen los factores.

Repetición

En EBNF, las llaves indican que la expresión puede repetirse cero o más veces.

Por ejemplo, la regla:

  <args> :: = <arg> {"," <arg>}

define una lista convencional de argumentos separados por comas.

Agrupamiento

Para indicar la precedencia, las gramáticas EBNF pueden usar paréntesis, (), para definir explícitamente el orden de expansión.

Por ejemplo, la regla:

  <expr> :: = <term> ("+" | "-") <expr>

define una forma de expresión que permite sumar y restar.

Concatenación

En algunas formas de EBNF, el operador, denota explícitamente la concatenación, en lugar de depender de la yuxtaposición.

Notación Backus-Naur

::= Definición de regla sintáctica

< > Delimitadores de nombre de regla sintáctica

“ ” Delimitadores de carácter o secuencia de

caracteres literal

| Separador de alternativas

( ) Agrupador sin repetición

{ } Agrupador con repetición

(0, 1 o más veces)

[ ] Agrupador opcional (0 o 1 vez)

Ejemplo 1:

<vocal> ::= “a” | “e” | “i” | “o”

| “u” | “á” | “é” | “í” | “ó”

| “ú” | “ü”

<consonante> ::= “b” | “c” | “d”

| “f” | “g” | “h” | “j” | “k”

| “l” | “m” | “n” | “ñ” | “p”

| “q” | “r” | “s” | “t” | “v”

| “w” | “x” | “y” | “z”

Ejemplo 2:

<letraIng> ::= “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”

Ejemplo 3

Como ejemplo, considere este BNF para una dirección postal de los EE.UU.

<dirección postal> ::= <nombre> <dirección> <apartado postal>

<nombre> ::= <personal> <apellido> [<trato>] <EOL>

| <personal> <nombre>

<personal> ::= <primer nombre> | <inicial> "."

<direccion> ::= [<dpto>] <numero de la casa> <nombre de la calle> <EOL>

<apartado postal> ::= <ciudad> "," <código estado> <código postal> <EOL>

**John Warner Backus**

Nació el 3 de diciembre de 1924, Filadelfia, P; líder del equipo de IBM que creó el lenguaje de programación Fortran; inventor del metalenguaje BNF, conocido como Backus-Normal o Backus-Naur Form; Actualmente propone los mejores métodos de programación, como el enfoque funcional.

Educación: BS, matemáticas, Universidad de Columbia, 1949; AM, matemáticas, Universidad de Columbia, 1950.

Experiencia profesional: IBM Corp .: programador, departamentos de ciencias puras y aplicadas, 1950-1953, gerente, departamento de investigación de programación, 1954-1958, personal de investigación de IBM, 1958-1963, miembro de IBM, 1963-1991; profesor adjunto de ciencias de la información, Universidad de California, Santa Cruz, 1974; Profesor visitante, Universidad de California, Berkeley, 1980, 1985.

Honores y premios: becario de IBM, 1963; W.W. Premio McDowell, IEEE, 1967; Medalla Nacional de Ciencia, 1975; Premio ACM Turing, 1977; IEEE Computer Society Pioneer Award, 1980; miembro de la Academia Nacional de Ciencias; miembro de la Academia Nacional de Ingenieros; Premio Charles Stark Draper, Academia Nacional de Ingeniería (NAE), 1993.



**Publicaciones significativas**

Backus, John W. y Harlan Herrick, "IBM 701 Speedcoding y otros sistemas de programación automática", ONR Symp. Programación automática para computadoras digitales, ONR, Washington D.C., 1954, pp. 106-113.

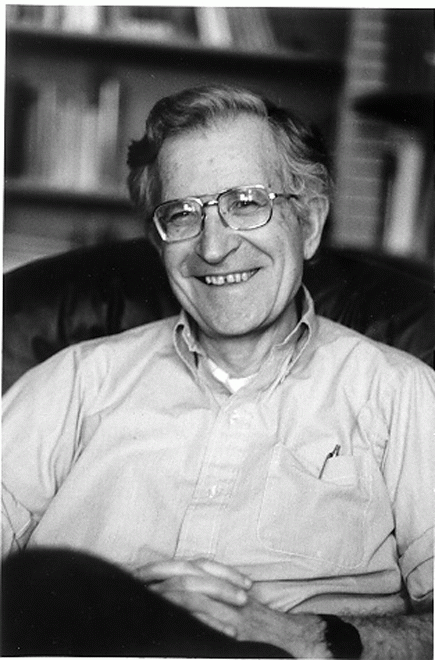
Backus, J.W., R.J. Beeber, S. Best, R. Goldberg, L. M. Haibt, H.L. Herrick, R.A. Nelson, D. Sayre, P.B. Sheridan, H. Stern, I. Ziller, R.A. Hughes y R. Nutt. Manual de referencia del programador, Sistema de codificación automática Fortran para IBM 704 EDPM, IBM Corporation, Nueva York, 1956.

Backus, John W., "La sintaxis y la semántica del lenguaje algebraico internacional propuesto de la Conferencia ACM-GAMM de Zurich", Proc. Primero Int'l. Conf. Procesamiento de la información, Butterworth, Londres, 1960, pp. 125-132.

Backus, John W., "¿Se puede liberar la programación del estilo von Neumann? Un estilo funcional y su álgebra de programas", Comm. ACM., Vol. 2 1, 1978, págs. 613-641.

**Noah Chomsky**

“El lenguaje es un proceso de creación libre; sus leyes y principios son fijos, pero la forma en que se utilizan los principios de generación es gratuita e infinitamente variada. Incluso la interpretación y el uso de las palabras implican un proceso de creación libre. ”- Noam Chomsky



Noam Chomsky es un filósofo, lógico, lingüista, científico cognitivo, activista y comentarista político estadounidense. Nacido el 7 de diciembre de 1928, es conocido como el "Padre de la lingüística moderna". Es una de las personalidades importantes en filosofía analítica. Actualmente enseña como profesor en el Instituto de Tecnología de Massachusetts (MIT). También ha escrito más de 100 libros, fue votado como "Intelectual público superior del mundo" en una encuesta realizada en 2005. Se graduó de la Universidad de Pennsylvania en el campo de la lingüística. Obtuvo su licenciatura, maestría y doctorado en lingüística. Después de eso, sirvió en la Universidad de Harvard en la sociedad de becarios.

1. CONCLUSIONES

La idea de transcribir la estructura del lenguaje con reglas de reescritura se remontan cuando menos al trabajo del gramático indio Panini (hacia el 460 a. C.), que la utilizó en su descripción de la estructura de palabras del idioma sánscrito (algunos incluso han sugerido renombrar BNF a Forma Panini-Backus). Lingüistas estadounidenses como Leonard Bloomfield y Zellig Harris llevaron esta idea un paso más adelante al tratar de formalizar el lenguaje y su estudio en términos de definiciones formales y procedimientos (1920-1960).

1. REFERENCIAS

* <https://es.wikipedia.org/wiki/Notación_de_Backus-Naur>
* <https://www.buscabiografias.com/biografia/verDetalle/1910/Noam%20Chomsky>
* <https://www.biografiasyvidas.com/biografia/b/backus.htm>
* <http://webdiis.unizar.es/~latre/prog1f/2017-18/Problemas_1_Notacion_BNF.pdf>
* <https://translate.google.com/?hl=es>