

BACKUS -NAUR (BNF)

Backus-aur (BNF)

Autor 1: Kenneth
Kaled Giraldo
Marulanda
*Ingeniería de sistemas y
computación,
Universidad Tecnológica
de Pereira, Pereira,
Colombia*
Correo-e:
k.giraldol@utp.edu.
co

Resumen— el lenguaje Backus-Naur es uno de los mas importantes ay que ha dado como base o ayuda a muchos de los otros lenguajes ya que es un metalenguaje usado para expresar gramáticas libres de contexto: es decir, una manera formal de describir lenguajes formales.

Palabras clave—programación, Backus-Naur, gramática libre, metalenguaje, lenguajes formales

Abstract— The Backus-Naur language is one of the most important a and that has given as a basis or help to many of the other languages since it is a metalanguage used to express context-free grammars: that is, a formal way of describing formal languages.

Key Word —programming, Backus-Naur, free grammar, metalanguage, formal languages

I. INTRODUCCIÓN

Es un metalenguaje que permite expresar las construcciones admitidas por un lenguaje formal, es decir, permite especificar la sintaxis de un lenguaje formal.

Generalmente, aparece en la documentación de cualquier lenguaje formal (Shell script, Java, C#, SQL, etc.), para especificar su sintaxis.

II.

CONTENIDO

1. Introducción
2. Contenido
3. Historia
4. Teoría simple

5. Ejemplos
6. Biografías

III. Historia

La idea de transcribir la estructura del lenguaje con reglas de reescritura se remonta 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).

Noam Chomsky, maestro de lingüística de alumnos de teoría de la información del MIT, combinó la lingüística y las matemáticas, tomando esencialmente el formalismo de Axel Thue como la base de su descripción de la sintaxis del lenguaje natural. También introdujo una clara distinción entre reglas generativas (de la gramática libre de contexto) y reglas transformativas (1956).

John Backus, un diseñador de lenguajes de programación de IBM, adoptó las reglas generativas de Chomsky para describir la sintaxis del nuevo lenguaje de programación IAL, conocido en la actualidad como ALGOL 58 (1959), presentando en el primer Congreso de Computación Mundial (World Computer Congress) el artículo «The syntax and semantics of the proposed international algebraic language of the Zurich ACM-GAMM Conference».

Peter Naur, en su reporte sobre ALGOL 60 de 1963, identificó la notación de Backus como la Forma Normal de Backus (Backus Normal Form), y la simplificó para usar un conjunto de símbolos menor, pero a sugerencia de Donald Knuth, su apellido fue agregado en reconocimiento a su contribución, reemplazando la palabra «Normal» por Naur, dado que no se trata de una forma normal en ningún sentido, a diferencia, por ejemplo de la Forma Normal de Chomsky.

IV. Teoría simple

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.

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

<simbolo>:: = <expresión con símbolos>

donde 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.

Mediante BNF se pueden expresar aspectos como qué palabras clave forman la sentencia, qué parámetros son opcionales y cuáles obligatorios, etc.

La notación BNF recoge cinco tipos de cláusulas básicas, que pueden combinarse para especificar las construcciones válidas de una proposición o sentencia del lenguaje

las desventajas BNF puede proporcionar información ambigua, incompleta o de difícil comprensión en sentencias complejas muy estructuradas.

V. Ejemplos

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

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

. Una dirección postal consiste de un nombre, seguido por una dirección, seguida por un apartado postal.

. Una parte "personal" consiste en un nombre o una inicial seguido(a) por un punto.

una gramática para enteros.

Un entero se define como una cadena consistente en un signo opcional(+ o -) seguido de una cadena de dígito (0 al 9). La siguiente gramática genera todos los enteros.

```
< dígito > ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
< entero > ::= < entero con signo > | < entero sin signo >
< entero con signo > ::= + < entero sin signo > | - < entero sin signo >
< entero sin signo > ::= < dígito > | < digito > < entero sin signo >
```

El símbolo de inicio es `< entero >`.

Ejemplo la derivación del entero -901 es

```
< entero > -> < dígito con signo >
-> - < entero sin signo >
-> - < dígito > < entero sin signo >
-> - < dígito > < digito > < entero sin signo >
-> - < dígito > < digito > < digito >
-> - 9 < dígito > < digito >
-> - 90 < dígito >
-> - 901
```

VI. Biografías



John Backus (Filadelfia, 3 de diciembre de 1924 - Oregón, 17 de marzo de 2007) fue un científico de la computación estadounidense.

Ganador del Premio Turing en 1977 por sus trabajos en sistemas de programación de alto nivel, en especial por su trabajo con FORTRAN.

Para evitar las dificultades de programación de las calculadoras de su época, en 1954 Backus se encargó de la dirección de un proyecto de investigación en IBM para el proyecto y realización de un lenguaje de programación más cercano a la notación matemática normal. De ese proyecto surgió el lenguaje FORTRAN, el primero de los lenguajes de programación de alto nivel que tuvo un gran impacto, incluso comercial, en la emergente comunidad informática.

Tras la realización de FORTRAN, Backus fue un miembro muy activo del comité internacional que se encargó del proyecto de lenguaje ALGOL. En ese contexto propuso una notación para la representación de las gramáticas usadas en la definición de un lenguaje de programación (las llamadas gramáticas libres de contexto). Tal notación se conoce como Notación de Backus-Naur (Backus-Naur Form o BNF) y une al nombre de Backus al de Peter Naur, un informático europeo del comité ALGOL que contribuyó a su definición.

En los años 1970, Backus se interesó sobre todo por la Programación funcional, y proyectó el lenguaje de programación FP, descrito en el texto que le sirvió para ganar el premio Turing, "Can Programming be Liberated from the Von Neumann Style?" Se trata de un lenguaje de uso fundamentalmente académico, que sin embargo animó un gran número de investigaciones. El proyecto FP, transformado en FL, se terminó cuando Backus se jubiló en IBM, en 1991.

John Backus falleció el sábado 17 de marzo de 2007, a la edad de 82 años en su casa en Ashland, Oregón por causas naturales, de acuerdo a la declaración de su familia.