

Backus- Naur form

Notación Backus- Naur

Autor 1: Kevin Alonso
Llanos Morales
*Ingeniería de sistemas y
computación,
Universidad Tecnológica
de Pereira, Pereira,
Colombia*
Correo-e:
k.llanos@utp.edu.co

Resumen— Backus-Naur Form, o formalismo Backus-Naur es uno de los lenguajes más importantes utilizados dentro de la informática, ya que representa una forma de expresar y definir los lenguajes de programación de forma concreta y precisa.

Palabras clave— Notación, Símbolos, Reglas, Metalenguaje

Abstract--- Backus-Naur Form, or formalism Backus-Naur is one of the most important languages used in computer science, since it represents a way to express and define programming languages in a concrete and precise way.

Keywords— Notation, Symbols, Rules, Metalanguage

I. INTRODUCCIÓN

La notación, o también conocida como Formalismo Backus-Naur, es un metalenguaje, es decir, que es un lenguaje que se utiliza para describir de una manera formal otro lenguaje, este formalismo se encuentra presente en la mayoría de los libros de teoría o semántica del lenguaje de programación.

II. PARA QUE SE USA

Es utilizado para expresar cómo debe escribirse una determinada proposición o sentencia en lenguaje determinado. Además, también mediante BNF se pueden expresar aspectos como qué palabras clave forman la sentencia, qué parámetros son opcionales y cuáles obligatorios, etc.

III. DONDE SE USA

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

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

Al ser un estándar de facto, no existe una versión normalizada. Por esa razón, en la documentación de distintos lenguajes pueden encontrarse distintas adaptaciones de BNF, aunque suelen ser bastante similares.

IV. CONTENIDO

1. Introducción
2. Para que se usa
3. Donde se usa
4. Contenido
5. Meta símbolos del BNF
6. Elementos terminales y no terminales
7. Ejemplos
8. Biografía
9. Cronología

V. Meta símbolos del BNF

`::=` de definición (el esquema de la derecha desarrolla el elemento de la izquierda)

`|` de alternativa (se puede elegir únicamente uno de los elementos que separa)

`{ }` de repetición (los elementos que incluyen, pueden repetirse cero o más veces)

`[]` de opción (los elementos que incluyen pueden utilizarse o no)

`()` de agrupación (sirven para agrupar los elementos que incluyen)

VI. Elementos terminales y no terminales

Elemento_no_terminal Son elementos que habrán de ser definidos por alguna regla, se encuentran entre `<>`.

Elemento_terminal Se emplea para representar los elementos que forman parte del lenguaje, lo usamos entre los siguientes símbolos ‘’.

VII. Ejemplos

Número ::= entPos
| entPos '.' entPos

entPos ::= digito
| digito entPos

digito ::= '0' | '1' | '2' | '3' | '4'
| '5' | '6' | '7' | '8' | '9'

<oración> ::= [<sujeito>] <predicado>

<sujeito> ::= <sintagma_nominal>

<predicado> ::=

(<verbo_copulativo> <atributo>)

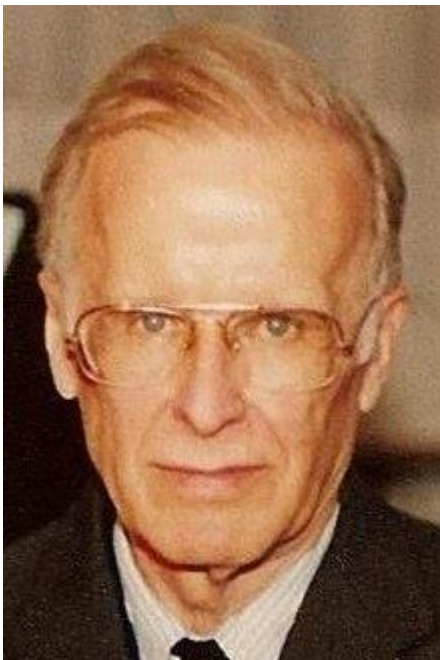
|(<verbo_predicativo>

[<complemento_directo>]

[<complemento_indirecto>]

{<complemento_circunstancial>})

VIII. Biografía



John Backus nació en Filadelfia el 3 de diciembre de

1924, y creció cerca de allí en Wilmington, Backus asistió a la prestigiosa escuela Hill Pottstown en Pensilvania. No era un buen estudiante, y sus años en la escuela se caracterizaron por una serie de fracasos.

No obstante, a las bajas calificaciones y la poca asistencia, Backus se graduó de la Hill School en 1942 y entra en la Universidad de Virginia.

Su padre, quería que siguiera sus pasos, estudiando Química, Backus comenzó el estudio de la Química, pero por un corto tiempo, en este tiempo disfrutó junto a su padre los aspectos teóricos de la ciencia, pero no le gustaba el trabajo de laboratorio. Al final de su segundo semestre, la asistencia a clases se redujo a una vez por semana, esta actitud condujo a las autoridades escolares a no permitirle a Backus continuar con los estudios.

El descubrimiento de la vocación hacía las Matemáticas, hace que Backus matricule en la Universidad de Columbia para estudiar con seriedad esta ciencia en el año 1946. En la primavera de 1949, Backus visitó el Centro de Computación de IBM en Madison Avenue, donde se encontraba la Selective Sequence Electronic Calculator, (SSEC), una de las primeras computadoras IBM. Backus le menciona a la guía que estaba buscando un trabajo. Ella le animó a hablar con el director del proyecto, y fue contratado para trabajar en la SSEC.

La SSEC no era una computadora en el sentido moderno. No tenía memoria para el almacenamiento de Software y los programas entraban en una cinta de papel perforado. Parte del trabajo de Backus fue de operar la máquina. Backus pasó tres años trabajando en la SSEC, tiempo durante el cual inventó un programa llamado Speedcoding. El programa fue el primero en incluir un factor de escala, lo que permitió manipular y almacenar grandes y pequeños números. A finales de 1953, Backus escribió un memo a su jefe en que esboza el diseño de un nuevo lenguaje de programación para la computadora IBM 704. La directiva de IBM aprobó la propuesta de Backus, y contrató a un equipo de programadores y matemáticos para trabajar con él.

En el otoño de 1954, la investigación de Backus y su equipo estaba bien definida en sus resultados y tenían listo para publicar un documento, denominado Preliminary Report, Specifications for the IBM Mathematical FORMula TRANslating System, FORTRAN. Junto con otros técnicos de IBM, visitó a los clientes que habían adquirido los IBM 704 para presentarles el nuevo lenguaje y obtener cualquier información o comentario que puedan tener. En ese momento, Backus prevé la finalización del compilador en seis meses. En su lugar, se necesitarían dos años. Cuando compilador estuvo listo, contenía 25 000 líneas

de código máquina, almacenadas en cinta magnética. Una copia del programa fue instalados en las IBM 704, junto con un manual de 51 páginas.

FORTTRAN fue diseñado para los matemáticos y científicos, y sigue siendo el lenguaje de programación por excelencia en estas áreas en la actualidad. Permite a las personas a trabajar con sus ordenadores sin tener que entender cómo funcionan las máquinas funcionan realmente, y sin tener que aprender lenguaje de máquina. Esto es lo que hace a FORTRAN mantenerse en 40 años después de su introducción gracias a la visión de Backus.

En 1959, desarrolló una notación Forma Normal de Backus, llamada posteriormente Forma Backus-Naur después de que Peter Naur hiciera modificaciones, en la misma se describe las reglas gramaticales de los lenguajes de alto nivel, y ha sido adaptada para su uso en varios lenguajes de programación. En la década de 1970, trabajó en la búsqueda de mejores métodos de programación, y desarrolló lo que él llamó un Lenguaje de nivel de función, o FP (para la programación funcional).

IX. Cronología

1942 se graduó de la escuela Hill Pottstown.

1942 ingresa en la Universidad de Virginia. Se unió al ejército.

1945 ingresa en la Facultad de Medicina de Nueva York.

1946 ingresa en la Escuela de Estudios Generales de Columbia

1949 recibe la Licenciatura en Matemáticas de la Escuela de Estudios Generales de Columbia.

1949 comienza a trabajar en el equipo de IBM SSEC.

1950-1952 Trabaja en el Laboratorio Watson de la Universidad de Columbia.

1954 Backus y su equipo publicarán FORTRAN.

1959 desarrolla una notación llamada de Backus-Naur Form, en colaboración con Naur

1991 jubilación

2007 fallece Ashland, Oregon