Backus Naur

John Backus

Autor: Jhon Mario Jiménez Ramírez

*Universidad tecnológica de Pereira, Pereira, Colombia*

Correo-e: jhon.jimenez1@utp.edu.co

***Resumen*—En ciencias de la computación, la forma Backus-Naur o Backus normal form ( BNF ) es una técnica de notación para gramáticas libres de contexto , a menudo utilizada para describir la sintaxis de los lenguajes utilizados en informática, como los lenguajes de programación de computadoras , formatos de documentos , conjuntos de instrucciones y comunicación protocolos . Se aplican donde sea que se necesiten descripciones exactas de los idiomas: por ejemplo, en las especificaciones oficiales del idioma, en los manuales y en los libros de texto sobre teoría del lenguaje de programación.**

***Palabras clave—* Términos-Acerca del índice de cuatro palabras o frases clave en orden alfabético, separadas por comas. Para obtener una lista de palabras claves sugeridas, envíe un correo electrónico en blanco a keywords@ieee.org o visite http://www.ieee.org/organizations/pubs/ani\_prod/keywrd98.txt.**

***Abstract*— It is called induction to a process of knowledge that consists in observing particular circumstances and from them generating a general conclusion. This type of procedure is inadmissible when verifying the truthfulness in the conclusion, to the extent that there may be unknown cases that deny or contradict the general conclusion. However, it can be useful as a provisional way of generating a theory that will then be empirically tested. The introduction in a period of history has been detected, and there is a high number of scientific observations. However, with the subsequent development of the scientific method, this type of procedure was completely ruled out.**

1. INTRODUCCIÓN-HISTORIA

La idea de describir la estructura del lenguaje utilizando reglas de reescritura se remonta al menos al trabajo de Pāṇini (antiguo gramático sánscrito indio y un venerado erudito en hinduismo que vivió en algún momento entre los siglos VII y IV a. C.). Su notación para describir la notación de estructura de palabras sánscritas es equivalente en potencia a la de Backus y tiene muchas propiedades similares.

En la sociedad occidental, la gramática se consideró durante mucho tiempo como un tema de enseñanza, más que un estudio científico; Las descripciones eran informales y dirigidas al uso práctico. En la primera mitad del siglo XX, lingüistas como Leonard Bloomfield y Zellig Harris comenzaron a intentar formalizar la descripción del lenguaje, incluida la estructura de las frases.

Mientras tanto, las reglas de reescritura de cadenas como sistemas lógicos formales fueron introducidas y estudiadas por matemáticos como Axel Thue (en 1914), Emil Post y Alan Turing Noam Chomsky , que enseña lingüística a estudiantes de teoría de la información en el MIT , combinó lingüística y matemática tomando lo que es esencialmente el formalismo de Thue como base para la descripción de la sintaxis del lenguaje natural . También introdujo una distinción clara entre las reglas generativas (las de las gramáticas libres de contexto) y las reglas de transformación.

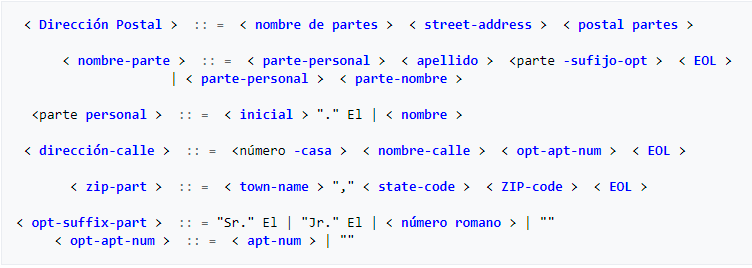
John Backus, diseñador de lenguaje de programación en IBM , propuso un metalenguaje de "fórmulas metalingüísticas" para describir la sintaxis del nuevo lenguaje de programación IAL, conocido hoy como ALGOL 58 Su notación se utilizó por primera vez en el informe ALGOL 60.

BNF es una notación para las gramáticas libres de contexto de Chomsky. Aparentemente, Backus estaba familiarizado con el trabajo de Chomsky.

Como propuso Backus, la fórmula definió "clases" cuyos nombres están encerrados entre corchetes angulares. Por ejemplo, <ab>. Cada uno de estos nombres denota una clase de símbolos básicos.

El desarrollo posterior de ALGOL condujo a ALGOL 60. En el informe del comité de 1963, Peter Naur llamó a la notación de Backus Backus forma normal. Donald Knuth argumentó que BNF debería leerse como una forma Backus-Naur, ya que "no es una forma normal en el sentido convencional", [9] a diferencia, por ejemplo, de la forma normal de Chomsky. También se sugirió una vez el nombre de forma de Backi Pāṇini en vista del hecho de que la forma normal de Backus de expansión puede no ser precisa, y que Pāṇini había desarrollado independientemente una notación similar anteriormente.

1. EJEMPLOS



Esto se traduce al español como:

Una dirección postal consta de una parte del nombre, seguida de una parte de la dirección de la calle, seguida de una parte del código postal.

Una parte de nombre consta de: una parte personal seguida de un apellido seguido de un sufijo opcional (Jr., Sr. o número dinástico) y un final de línea, o una parte personal seguida de una parte de nombre (Esta regla ilustra el uso de la recursividad en BNFs, cubriendo el caso de personas que usan múltiples nombres e iniciales.

Una parte personal consta de un nombre o una inicial seguida de un punto.

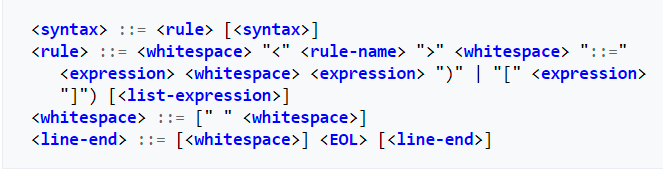
Una dirección de calle consiste en un número de casa, seguido de un nombre de calle, seguido de un especificador de apartamento opcional , seguido de un final de línea.

Una parte zip consta de un nombre de ciudad , seguido de una coma, seguido de un código de estado , seguido de un código postal seguido de un final de línea.

Una parte de sufijo-opt consiste en un sufijo, como "Sr.", "Jr." o un número romano, o una cadena vacía (es decir, nada).

Un opt-apt-num consiste en un número de apartamento o una cadena vacía (es decir, nada).

EJEMPLOS



Esto asume que no hay Whitespace necesario para la interpretación apropiada de la regla. El <QUOTE> se presume para ser el carácter ", y el <EOL> para ser el fin de línea apropiado especificado (en ASCII, retorno de carro o línea nueva, dependiendo del sistema operativo). El <rule-name> y el <text> deben ser substituidos con nombre/etiqueta o el texto literal de una regla declarada, respectivamente.

### Software que usa BNF

* ANLTR, otro generador de analizador escrito en JAVA
* Qlik Sense, una herramienta de BI, utiliza una variante de BNF para la secuencia de comandos
* Convertidor BNF (BNFC), que opera en una variante llamada "etiquetada forma Backus-Naur" (LBNF). En esta variante, cada producción para un no terminal dado recibe una etiqueta, que puede usarse como un constructor de un tipo de datos algebraicos que representa ese no terminal. El convertidor es capaz de producir tipos y analizadores para sintaxis abstracta en varios idiomas, incluidos Haskell y Java.
* Coco / R, generador de compiladores que acepta una gramática atribuida en EBNF
* DMS Software Reengineering Toolkit , sistema de análisis y transformación de programas para lenguajes arbitrarios.
* Analizador de oro BNF
* GNU bison , versión GNU de yacc
* Analizador RPA BNF.  Análisis de demostración en línea (PHP): JavaScript, XML
* Sistema XACT X4MR, un sistema experto basado en reglas para programar la traducción del lenguaje
* XPL Analyzer, una herramienta que acepta BNF simplificado para un idioma y produce un analizador para ese idioma en XPL; se puede integrar en el programa SKELETON suministrado, con el cual se puede depurar el idioma [[17]](https://en.wikipedia.org/wiki/Backus%E2%80%93Naur_form#cite_note-17) (un programa contribuido SHARE , que fue precedido por *A Compiler Generator* , ISBN  978-0-13-155077-3 )
* Yax, generador de analizador sintáctico (usado con el preprocesador Lex)
* bnfparser, una utilidad universal de verificación de sintaxis
* bnf2xml, Marcado de entrada con etiquetas XML utilizando la coincidencia avanzada de BNF.
* JavaCC, Compilador Java Compilador tm (JavaCC tm) - El generador de analizador Java.
* Herramientas de análisis de Racket, análisis tipo lex y yacc (edición Beautiful Racket)

BIOGRAFIAS

**John Backus**: nació en filadelfia, el 3 de diciembre de 1924, y murió en Oregón, Estados Unidos; el 17 de marzo de 2007, fue un científico de la computación estadunidense.

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.

**JOHN BACKUS**



REFERENCIAS

1. <https://es.wikipedia.org/wiki/John_Backus><https://es.wikipedia.org>
2. <https://es.wikipedia.org/wiki/Notaci%C3%B3n_de_Backus-Naur><https://es.wikipedia.org/wiki/M._C._Escher>
3. <http://webdiis.unizar.es/~latre/prog1f/2017-18/Problemas_1_Notacion_BNF.pdf>
4. <https://en.wikipedia.org/wiki/Backus%E2%80%93Naur_form#Introduction>