Notación de Backus-Naur

Backus-Naur notation

Autor 1: John Stiven Acevedo Zapata

*Universidad tecnológica de pereira*

Correo-e: s.acevedo1@utp.edu.co

***Resumen*— En ciencias de la computación , la forma Backus-Naur o Backus-Naur 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—* Backus-Naur, informática, gramáticas, idiomas.**

***Abstract*— In computer science, Backus–Naur form or Backus normal form (BNF) is a notation technique for context-free grammars, often used to describe the syntax of languages used in computing, such as computer programming languages, document formats, instruction sets and communication protocols. They are applied wherever exact descriptions of languages are needed: for instance, in official language specifications, in manuals, and in textbooks on programming language theory.**

***Key Word* —** **Backus–Naur, Informatic, grammars, languages.**

1. INTRODUCCIÓN

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).

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.

1. CONTENIDO

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.

**EJEMPLOS**

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>

**BIOGRAFIAS**

**- Pāṇini**

Pāṇini vivió en Gandhara, después de la época de Buda (420–368 a. C.) y antes de la primera mención de la muerte de Pāṇini (por un león), en el Pancha-tantra (hacia el 200 a. C.).2​ Se le considera un inspirado muni (‘pensador’, siendo mauná: ‘silencio’).

La forma Pāṇini, es la transcripción en AITS,1​ de la forma पाणिनि (en escritura devanagari).1​ La pronunciación comúnmente aceptada es /pɑːˈɳini/.1​ El término Pāṇini se interpreta nomalmente como '[descendiente] de Pāṇina', de acuerdo con el Aṣṭādhyāyī 4.1.95.1​ Pāṇina significa '[descendiente] de Pāṇin', (Ashta-adhiaia, 6.4.165),1​Pāṇin es el nombre de una familia de la tribu kauśika, según el Jari-vamsa y el Visnú-purana.1​

Era llamado Daksheia (hijo de la señora Dakshí). Su abuelo se llamaba Devala. A Pāṇini también se le llamó Shalaturíia (śālāturīya): ‘natural de Shalatura’, una aldea en la margen derecha del río Indo, en el noroeste de la India, cerca de Attock y de Peshawar, en la actual provincia pakistaní de Jiber Pajtunjua.

**- Noam Chomsky**

Noam Chomsky nació el 7 de diciembre de 1928 en Filadelfia (Pensilvania), hijo del doctor William (Zev) Chomsky (estudioso de la lengua hebrea y uno de sus más distinguidos gramáticos) y de Elsie Simonofsky, maestra de hebreo. Ambos eran inmigrantes judeo-ucranianos. Desde 1945, estudió filosofía, lingüística y matemática en la Universidad de Pensilvania. Allí estuvo bajo la tutela del profesor Zellig Harris (también inmigrante judeo-ucraniano, fundador del primer departamento especializado en lingüística en Norteamérica). Harris y Elsie influyeron más que Zev en la formación de su ideología política. También por influencia de Zellig Harris, Chomsky comenzó a tomar clases de matemáticas y filosofía. Uno de sus maestros fue el filósofo Nelson Goodman, quien más tarde los presentaría en la Society of Fellows de Harvard. Recibió su doctorado en 1955, después de llevar a cabo la mayor parte de sus investigaciones en la Universidad de Harvard durante los cuatro años anteriores. En 1998 recibió el Doctorado honoris causa (lingüística) de la Universidad Rovira i Virgili.5​ Recibió esta misma distinción por parte de la Universidad Nacional de Colombia6​ en 2002, de la Universidad de Chile7​ y de la Universidad de la Frontera8​ en 2006 y de la Universidad Nacional Autónoma de México9​ en 2010.

En su tesis doctoral comenzó a desarrollar algunas de sus ideas en lingüística, elaborándolas luego en su libro Estructuras sintácticas, posiblemente su trabajo más conocido en este campo. Sus planteamientos lingüísticos han revolucionado muchos puntos clave del estudio del lenguaje humano, que se han plasmado en la teoría de la Gramática generativa transformacional.

Es profesor del Massachusetts Institute of Technology (MIT) desde 1961, donde ocupó la cátedra Ferrari P. Ward de Lenguaje Moderno y Lingüística de 1966 a 1976.

En abril de 2019 es reconocido con el Premio Fronteras del Conocimiento en la categoría de Humanidades y Ciencias Sociales. El jurado recogió en el acta que Chomsky ha situado la investigación de la mente humana y sus productos “en una nueva y fructífera vía que abarca la lingüística teórica, la psicolingüística, las ciencias cognitivas, las filosofías del lenguaje y de la mente y la psicología cognitiva”. 10​

Su cónyuge fue Carol Schatz, quien murió el 20 de diciembre de 2008.11​Tiene dos hijas y un hijo.

**- John Backus**

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.

**IMPACTOS A LA SOCIEDAD**

La facilitación del uso de lenguajes de programación.

1. CONCLUSIONES

En conclusión la Notación de Backus-Naur fue la implementación de signos, números y la matemática al lenguaje de la informática y la progtamacion.

REFERENCIAS

1. <https://en.wikipedia.org/wiki/Backus%E2%80%93Naur_form>
2. <https://es.wikipedia.org/wiki/Panini_(gram%C3%A1tico)>
3. <https://es.wikipedia.org/wiki/Noam_Chomsky>
4. <https://es.wikipedia.org/wiki/John_Backus>
5. <https://translate.google.com/?hl=es#view=home&op=translate&sl=en&tl=es&text=In%20computer%20science%2C%20Backus–Naur%20form%20or%20Backus%20normal%20form%20(BNF)%20is%20a%20notation%20technique%20for%20context-free%20grammars%2C%20often%20used%20to%20describe%20the%20syntax%20of%20languages%20used%20in%20computing%2C%20such%20as%20computer%20programming%20languages%2C%20document%20formats%2C%20instruction%20sets%20and%20communication%20protocols.%20They%20are%20applied%20wherever%20exact%20descriptions%20of%20languages%20are%20needed%3A%20for%20instance%2C%20in%20official%20language%20specifications%2C%20in%20manuals%2C%20and%20in%20textbooks%20on%20programming%20language%20theory>.