[[1]](#footnote-1)

La notación de Backus-Naur

Juan Camilo García Braham

*Universidad Tecnológica de Pereira, Pereira, Colombia*

Correo-e: juan.garcia4@utp.edu.co

Conocida también en inglés como Backus-Naur Form (BNF), Backus-Naur Formalism o Backus normal form, es un **metalenguaje** usado para expresar [gramáticas libres de contexto](https://es.wikipedia.org/wiki/Gram%C3%A1tica_libre_de_contexto): es decir, una manera formal de describir [**lenguajes formales**](https://es.wikipedia.org/wiki/Lenguaje_formal).

Para una mejor comprensión sobre la notación de Backus-Naur, hay que tener en cuentas unos cuantos términos mencionados anteriormente, como metalenguaje, lenguajes formales y la gramática libre de contexto.

Un metalenguaje “es un lenguaje que se usa para hablar acerca de otro lenguaje” **(1)**., en otras palabras, un metalenguaje es la forma correcta de hablar sobre otro lenguaje, sus características y otros aspectos del mismo. Estos metalenguajes se ven en varios campos, pero en este ensayo, se va a hacer énfasis en el metalenguaje relacionado con la informática, y específicamente la notación de Backus-Naur.

Ahora, la gramática libre de contexto es “una gramática formal en la que cada regla de producción es de la forma:

V → *w*

Donde V es un [símbolo no terminal](https://es.wikipedia.org/w/index.php?title=S%C3%ADmbolo_no_terminal&action=edit&redlink=1) y *w* es una cadena de terminales y/o no terminales”**(2)**. Para entender mejor este concepto debemos ver la definición de gramática formal, la cual dice que es una estructura matemática con reglas de formación que establecen una serie de cadenas que son admitidas por cierto lenguaje, ya sea formal o natural **(3)**.

Por último, el lenguaje formal es (como su nombre lo indica) un lenguaje cuyo contenido y reglas están formalmente especificadas. El contenido del lenguaje se puede dividir en dos ramas, los símbolos primitivos y la gramática formal. Los símbolos primitivos son el alfabeto como tal, mientras que la gramática formal son las reglas que se deben tener en cuenta a la hora de hacer una cadena de caracteres (también conocido como sintaxis) **(4)**.

Ya que se cubrieron los aspectos importantes, podemos entrar en materia. Entonces, la notación de Backus-Naur es una forma conveniente de analizar los lenguajes formales.

Su historia se remonta a 460 a. C. con el trabajo de Panini, un gramático indio; su trabajo como tal consistió en “transcribir la estructura del lenguaje con reglas de reescritura, esto lo utilizó en la descripción del idioma sánscrito (su lengua materna), y en especial su estructura de palabras **(5)**.

Con el paso del tiempo, otras personas decidieron avanzar en este tema, tratando de formalizar el lenguaje y su estudio en términos de definiciones formales y procedimientos, estas personas fueron Leonard Bloomfield y Zellig Harris, en el periodo entre 1920 y 1960.

En 1959, John Backus, un diseñador de lenguajes de programación de IBM, basándose en las reglas establecidas anteriormente por Noam Chomsky, describió la sintaxis del nuevo lenguaje de programación que lo bautizó como IAL, conocido en la actualidad como ALGOL 58 , esto lo llevó a presentar en el “primer Congreso de Computación Mundial ([World Computer Congress](https://es.wikipedia.org/w/index.php?title=World_Computer_Congress&action=edit&redlink=1)) el artículo «The syntax and semantics of the proposed international algebraic language of the Zurich ACM-GAMM Conference»”**(6)**.

Más tarde, en 1963, Peter Naur **(7)**., simplifico el trabajo hecho por John Backus, y lo bautizó como la “Forma normal de Backus”; con su reporte realizado del ALGOL 60, el experto de la ciencia de la computación Donald Knuth **(8)**., sugirió la adición del apellido de Peter al nombre que se le dio a la “Forma normal de Backus”, convirtiéndose en la “Forma normal de Backus-Naur”.

Podemos ver mejor su función con el siguiente ejemplo

“Como ejemplo, considere este BNF para una [dirección postal](https://es.wikipedia.org/wiki/Direcci%C3%B3n_postal) de los [EE.UU.](https://es.wikipedia.org/wiki/Estados_Unidos)

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

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

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

| <**personal**> <**nombre**>

<dirección> ::= [<**dpto**>] <número de la casa> <**nombre de la calle**> <**EOL**>

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

Esto se traduce a español como:

* Una dirección postal consiste en 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.
* Un nombre consiste de: una parte personal seguida por un apellido seguido opcionalmente por una jerarquía o el trato que se la da a la persona (Jr., Sr., o número dinástico) y un salto de línea (*end-of-line)*, o bien una parte personal seguida por un nombre (esta regla ilustra el uso de la repetición en BNFs, cubriendo el caso de la gente que utiliza múltiples nombres y los nombres medios o las iniciales).
* Una dirección consiste de una especificación opcional del departamento, seguido de un número de casa, seguido por el nombre de la calle, seguido por un salto de línea (*end-of-line)*.
* Un apartado postal consiste de una ciudad, seguida por una coma, seguida por un código del estado (recuerde que es un ejemplo que ocurre en EE.UU.), seguido por un código postal y este seguido por un salto de línea (*end-of-line)*.

Observe que muchas cosas (tales como el formato de una parte personal, de una especificación del apartamento, o código postal) están dejadas sin especificar aquí. Si es necesario, pueden ser descritas usando reglas adicionales de BNF, o dejadas como abstracción si es inaplicable para el propósito actual” **(9)**.

Otro ejemplo cercano a nosotros, es la forma en la cual podemos analizar los autómatas, dividiéndolos de tal forma que se muestren las reglas a seguir del lenguaje, específicamente, el lenguaje que hicimos en la última clase (liig3), de esta forma se facilita su “modus operandi”. Probablemente a lo largo de la carrera, se verán otros ejemplos de aplicación de este útil mecanismo de interpretación de lenguajes, con el fin de entenderlos y operar con ellos de manera correcta.

Referencias

1. Wikipedia, “Metalenguaje” Wikipedia. (<https://es.wikipedia.org/wiki/Metalenguaje>)

(<https://www.google.com/search?q=metalenguaje&rlz=1C1CHBF_esCO834CO834&oq=meta&aqs=chrome.0.69i59j69i60l2j69i57j69i60j0.767j0j1&sourceid=chrome&ie=UTF-8>)

1. Wikipedia, “Gramática libre de contexto”, Wikipedia. (<https://es.wikipedia.org/wiki/Gram%C3%A1tica_libre_de_contexto>)
2. Wikipedia, “Gramática formal” , Wikipedia.

(<https://es.wikipedia.org/wiki/Gram%C3%A1tica_formal>)

1. Wikipedia, “Lenguaje formal,” Wikipedia. (<https://es.wikipedia.org/wiki/Lenguaje_formal>)
2. Wikipedia, “Panini (gramático)”, Wikipedia. (<https://es.wikipedia.org/wiki/Panini_(gram%C3%A1tico))>
3. Wikipedia, “John Backus”, Wikipedia. (<https://es.wikipedia.org/wiki/John_Backus>)
4. Wikipedia “Peter Naur”, Wikipedia (<https://es.wikipedia.org/wiki/Peter_Naur>)
5. Wikipedia “Donald Knuth”, Wikipedia (<https://es.wikipedia.org/wiki/Donald_Knuth>)
6. Wikipedia “Notación de Backus-Naur”, Wikipedia (<https://es.wikipedia.org/wiki/Notaci%C3%B3n_de_Backus-Naur>)

**Juan Camilo García Braham**

Nacido en enero del 2002, en Pereira, Risaralda, Juan Camilo García Braham inició sus estudios de bachiller en el colegio Inem Felipe Pérez de Pereira, de donde salió premiado como el mejor bachiller técnico de su institución y recibió su título de Técnico en sistemas aprobado por el SENA en 2018. Actualmente está cursando su educación superior en la Universidad Tecnológica de Pereira, aspirando a ser un ingeniero de sistemas y posteriormente ingeniero robótico.

1. [↑](#footnote-ref-1)