# Form Backus Naur

Andrés Córdoba Aguirre

José Gilberto Vargas Cano

(Profesor Introducción a la Informática)

Universidad Tecnológica de Pereira

Primer Semestre

Ingeniería de Sistemas

Pereira, Abril de 2019

La notación de Backus-Naur es un metalenguaje usado para expresar gramáticas libres de contexto, es decir, una manera 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 (por ejemplo, el metro en la poesía de Venpa). La mayoría de los libros de textos para la teoría o la semántica del lenguaje de programación documentan el lenguaje de programación en BNF.

Algunas variantes, tales como la Augmented Backus-Naur Form (ABNF) y la Extended Backus–Naur Form (EBNF), tienen su propia documentación.

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.

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.

Hay muchas variantes y extensiones de BNF, posiblemente conteniendo algunos o todos los comodines de expresiones regulares como un "\*" o "+". El Extended Backus-Naur form (EBNF) es una variante común. De hecho el ejemplo anterior no es la forma pura inventada para el informe del ALGOL 60. La notación de los corchetes "[ ]" fue introducida algunos años más tarde en la definición de PL/I de la IBM pero ahora se reconoce universal. La ABNF es otra extensión usada comúnmente para describir protocolos del IETF.

Las expresiones gramaticales de analizadores sintácticos construidas en BNF y las notaciones de expresión regular para formar una clase alternativa de la gramática formal, que es esencialmente analítica más que generativa en carácter.

Muchas especificaciones de BNF disponibles en línea tienen como propósito ser legibles a simple vista y no son especificaciones formales. Estas incluyen con frecuencia algunas de estas reglas sintácticas y extensiones:

Elementos opcionales son presentados entre corchetes. Por ejemplo [<elemento-x>]

Los elementos que se repiten 0 o más veces son presentados entre paréntesis de llave o terminados con un asterisco. Por ejemplo <palabra> ::= <letra> {<letra>}

Los elementos que se repiten 1 o más veces son terminados con un '+'

Los terminales pueden aparecer en negrillas y los no-terminales en texto normal en lugar de utilizar itálicas o paréntesis de ángulo

Alternativas opcionales son separadas por el símbolo '|'

Cuando se requiere agrupar varios elementos, se hace con paréntesis simples.