BACKUS NAUR

Backus naur

Autor 1: Aldair Estiven Lasso Acosta

*Ingeniería de Sistemas, Universidad Tecnológica de Pereira*

Correo-e: aldair.lasso@utp.edu.co

***Resumen*— la notación de Backus Naur o Backus Naur form es un metalenguaje empleado para expresar gramáticas libres de contexto es mejor dicho una manera formal de describir lenguajes formales, el BNF e extensamente utilizado para notación para las gramáticas de los lenguajes de programación, sistemas de comando y sistemas de comunicación, también para representar gramáticas del lenguaje natural,** **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.**

***Palabras clave—* metalenguaje, gramáticas, formal, lenguajes.**

***Abstract*—**

**the notation of Backus Naur or Backus Naur form is a metalanguage used to describe context-free grammars is rather a formal way to describe formal languages, the BNF and widely used for notation for the grammar of programming languages, command systems and communication systems, also to represent natural language grammar, Most textbooks for the theory or semantics of documentary programming language programming language in BNF.**

***Key Word* — metalanguage, grammar, formal, languages.**

1. INTRODUCCIÓN

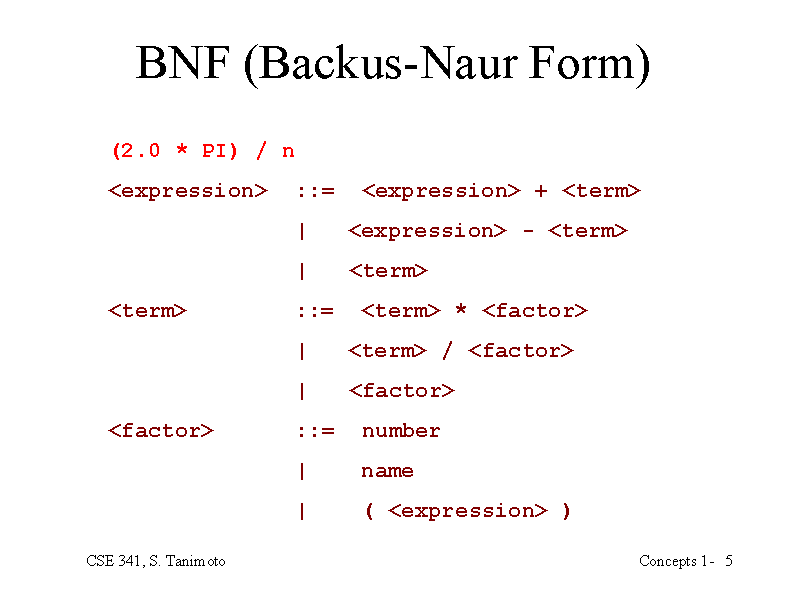
La notación de Backus-Naur, o en sus denominaciones inglesas Backus-Naur form (BNF), Backus-Naur formalismo o Backus normal form, es un metalenguaje usado para expresar gramáticas libres de contexto: mejor dicho, una manera formal de describir lenguajes formales.

El BNF se utiliza ampliamente como notación para las gramáticas de los lenguajes de programación, de sistemas de comando y de los protocolos de comunicación, así como una notación para representar partes santanicas de la lengua natural. La mayoría de los libros para la teoría o semántica del lenguaje de programación documentan el lenguaje de programación en BNF.

Otras versiones o varientes como la Augmented Backus-Naur Form (ABNF) y la Extended Backus–Naur Form (EBNF), tienen su propia documentación.

1. CONTENIDO

El software para los primeros computadores estaba primariamente escrito en lenguaje ensamblador. Normalmente para un programador es más productivo usar un lenguaje de alto nivel, y los programas escritos en lenguajes de alto nivel pueden ser reutilizados en distintos tipos de computadores. Aun teniendo en cuenta esto, pasó un tiempo hasta que los compiladores se establecieran, porque generaban código que no tenía tan buen rendimiento como los ensambladores escritos a mano, eran enormes proyectos de desarrollo por sí mismos, y la limitadísima capacidad de memoria de los primeros computadores creó muchos problemas técnicos para las implementaciones prácticas de los compiladores.



El primer compilador fue escrito por Grace Hopper, en 1952, para el lenguaje Sistema A-0. El término compilador fue acuñado por Hopper. El equipo FORTRAN dirigido por John W. Backus de IBM está generalmente acreditado por haber presentado el primer compilador completo, en 1957. El primer compilador FORTRAN necesitó de 18 años-persona para su creación.

ElBNF se utiliza como notación para las sintaxis de los lenguajes de programación de la computadora, de los sistemas de comando y de los protocolos de comunicación, así como una notación para reflejar partes de las gramáticas de la lengua natural. Utiliza reglas que se construyen con tres tipos de símbolos:

Meta símbolos:

son los propios de la BNF.

Símbolos terminales:

son los que se usan en el texto del programa tal como aparecen en la regla sin comillas.

Símbolos no terminales:

Son elementos que habrán de ser definidos por alguna regla, se encuentran entre <>. Se definen usando combinaciones de símbolos terminales, no terminales y meta símbolos.

Una especificación de BNF es un sistema de reglas de derivación, escrito como:

<símbolo>: = <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.

La idea de transcribir la escritura del lenguaje con reglas de escritura se remonta a el trabajo de gramática de hace una mucho tiempo elaborado por un indio llamado panifique la utilizó en la estructura de las palabras del idioma sanscrito.

lingüísticos de estados unidos 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.

Noam Chomsky, maestro de lingüística de 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.

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 presentando en uno de los primeros congresos a nivel mundial de computación 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 , 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 ninguna forma, a diferencia, por ejemplo de la Forma Normal de Chomsky.

Jerarquía de Chomsky:

Taxonomía de los lenguajes de programación implementada por el lingüista, filósofo y activista estadounidense Noam Chomsky.

John Backus propuso "fórmulas meta-lingüísticas" para describir la sintaxis del nuevo lenguaje de programación IAL, conocido hoy en día como ALGOL 58 (1959). El trabajo de Backus estaba basado en la Máquina de Post ideada por Emil Post.

El posterior desarrollo de ALGOL llevó al ALGOL 60; en su informe (1963), Peter Naur llamó Notación de Backus-Naur (Backus Normal Form: BNF) a la notación de Backus, y lo simplificó para minimizar el conjunto de caracteres usados.

Niklaus Wirth definió la Notación de Backus-Naur Extendida (Extended Backus-Naur Form: EBNF), una versión refinada de la BNF, a principios de los años 70 para PL/0.

La Notación de Backus-Naur Aumentada (Augmented Backus-Naur Form: ABNF) es otra variante. Tanto EBNF como ABNF son muy usadas para especificar la gramática de los lenguajes de programación, como en las entradas de los generadores de analizadores sintácticos, y en otros campos como la definición de protocolos de comunicación.

1. CONCLUSIONES

La notación de Backus Naur es una manera de escribir metalenguaje, y es empleado en la computación, tiene relación con compiladores, se utiliza para lenguajes de programación, sistema de comandos y protocolos de comunicación. Utiliza reglas que se construyen con tres tipos de símbolos: meta símbolos, símbolos terminales, símbolos no terminales. Se utiliza como notación para la sintaxis de los lenguajes de programación.

REFERENCIAS

1. J. F. Fuller, E. F. Fuchs, and K. J. Roesler, "Influence
2. <https://prezi.com/29zqo-jyqmiq/notacion-de-backus-naur-y-forma-normal-de-chomsky/>
3. <https://es.wikipedia.org/wiki/Notaci%C3%B3n_de_Backus-Naur>
4. <http://agrega.educacion.es/repositorio/28042017/b7/es_2017042812_9190105/notacion_bnf_-_v1cc.pdf>