Aproximación a la Backus-Naur form

Approximation to the Backus-Naur form

Autor 1: Camilo Muñoz Albornoz
Risaralda , Universidad Tecnológica de Pereira, Pereira, Colombia
Correo-e: Camilo.Munoz2@utp.edu.co

***Resumen*— El artículo presente realizara una aproximación acerca de la temática de la notación BNF que es una forma para conocer como se escribe en los lenguajes formales, para que cualquier instrucción digitada en un lenguaje de programación, este correctamente diseñada, así pueda ser usada.**

Palabras clave--- compiladores, gramática, lenguaje, notación, sintaxis.

***Abstract*— The present article will make an approximation about the subject matter of BNF notation which is a way to know how it is written in formal languages, so that any instruction typed in a programming language, is correctly designed, so that it can be used.**

***Key Word* — compilers, grammar, language , notation , syntax**

I. INTRODUCCIÓN

Cuando las personas se comunican por medio de una lengua, esta tiene sus reglas, sus símbolos , su estructura, la cual es necesaria para que pueda ser entendida por otra persona, por ejemplo la lengua española tiene diferentes símbolos, como son las letras, con las que se forman palabras , oraciones, que tienen que estar correctas para ser entendidas, pues no es lo mismo decir "el perro es de juan" a decir "el juan es del perro", donde vemos que el orden afecta al significado de la oración. A si mismo funciona en los lenguajes de programación, donde existen sus reglas, para que este pueda ser entendido por la computadora, y para especificar sus reglas se utilizan unas técnicas de notación.

II. CONTENIDO

¿Qué es la Notación de Backus Naur?

Conceptualmente, la Notación de Backus Naur, es un metalenguaje, es decir, el lenguaje usado para describir aspectos propios de una lengua, que especifica la descripción compacta y precisa de la sintaxis de un lenguaje formal con símbolos y reglas [1]. A su vez, este permite que no exista confusión en términos de lo que está o no permitido a la hora de su uso. Específicamente, esta notación, tiene un planteamiento matemático formal, que describe el orden en el que se debe plantear un lenguaje de programación.

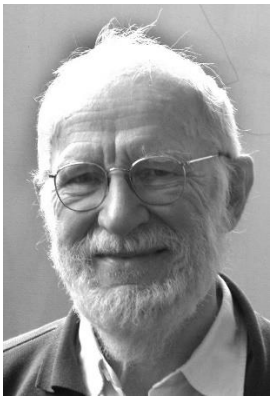
Creadores

Jhon Backus

Fue un científico de la computación estadounidense, ya ganador del premio Turing, que Nació el 3 de diciembre de 1924 en Oregón y falleció a los 82 años el 17 de marzo de 2007, fue un pionero de la informática destacado por sus 2 principales aportes al mundo de la computación, pues fue el creador de Fortran, el primer lenguaje de programación de alto nivel, y además invento la Backus normal Form, una notación que permite describir la sintaxis de un lenguaje de programación de alto nivel

Fecha de Recepción: (Letra Times New Roman de 8 puntos)
Fecha de Aceptación: Dejar en blanco

Peter Naur



Nacido el 25 de octubre de 1928 en Frederiksberg, Dinamarca y fallecido el 3 de enero de 2016 a los 87 años, fue un científico danés, pionero en la informática y ganador del premio Turing en 2005, contribuyo en la creación del lenguaje de programación algol 60, e hizo parte en la constitución del BNF (Backus-Naur form)

Historia de la BNF

Noam Chomsky maestro de lingüística en el instituto de tecnología en Massachussets, combino 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)
Jhon Backus adopto las reglas de Chomsky para describir un nuevo lenguaje de programación conocido ahora como ALGOL 58, 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, entonces surgió la llamada Backus Nour form donde se remplazó Normal por Nour, como reconocimiento a la contribución que hizo Peter Nour, gracias a sugerencia de Donald Knuth [2]

¿Características de la BNF?

Elementos [5]

- 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 metasímbolos, para describir los constructores sintácticos del LP sujeto

- símbolos terminales; son los que se usan en el texto del programa tal como aparecen en la regla (sin las comillas), para describir los símbolos (texto) del LP sujeto

- Los metasímbolos BNF son:

1. ::= de definición (el esquema de la derecha desarrolla el elemento de la izquierda)
2. | de alternativa (se puede elegir únicamente uno de los elementos que separa)
3. <> los paréntesis angulares utilizados para rodear los nombres de las categorías

Regla de producción para la BNF

BNF es una especie de juego matemático: se empieza con un símbolo (llamado símbolo de inicio y, por convención, se suele denominar S en los ejemplos) y luego se le dan las reglas por las que se puede sustituir este símbolo. El lenguaje definido por la gramática de BNF es sólo el conjunto de todas las cadenas que puedes producir siguiendo estas reglas.

Las reglas se llaman reglas de producción, y se ven así:

simbolo := alternativa1 | alternativa2 ..

la regla de fabricación simplemente establece que el símbolo a la izquierda de := debe ser reemplazado por una de las alternativas a la derecha. Las alternativas están separadas por |s. (Una variación de esto es usar ::= en lugar de :=, pero el significado es el mismo). Las alternativas suelen consistir tanto en símbolos no terminales como en símbolos terminales. Se llaman

terminales porque no hay reglas de producción para ellos: terminan el proceso de producción. [3]

También en las reglas de producción se permite la recursividad
– a izquierdas: <serie-instr> ::= <instr> | <serie-instr> ;<instr>
– a derechas: <serie-instr> ::= <instr> | <instr> ; <serie-instr>

Ejemplos de notación BNF

Ejemplo 1

En ALGOL 60, un identificador (que es el nombre de una entidad, como por ejemplo una variable) consiste en una cadena de caracteres alfanuméricos (esto es, letras y números) que deben comenzar por una letra. Podemos utilizar estas reglas en la forma de Backus-Naur para describir un conjunto de identificadores validos: [6]

- [identificador] ::= [letra] | [identificador] [letra] | [identificador] [digito]
- [letra] ::= a | b | c | … | x | y | z (la elipsis indica que se incluyen todas las letras del alfabeto).
- [digito] ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Por ejemplo, podemos producir el identificador valido x99a utilizando la primera regla para reemplazar [identificador] por [identificador] [letra], luego la segunda regla para obtener [identificador]a, después la primera regla dos veces para obtener [identificador][digito][digito]a, luego la primera regla para obtener [letra][digito][digito]a, por último, se reemplaza [letra] por x, y ambos [digito] por 9. [6]

Ejemplo 2
Sintaxis de los números enteros positivos en notación BNF

- <numero entero> ::= <signo opcional> <secuencia dígitos>
- <signo opcional> ::= + | <nada>
- <secuencia dígitos> ::= <dígito> | <dígito> <secuencia dígitos>
- <dígito> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
- <nada> ::=

Variante de la notación BNF

EBNF (Extended Backus–Naur form)

Es un metalenguaje que también sirve para especificar sintaxis de diferentes lenguajes formales, este se trata de una versión extendida de la notación de Backus-Naur, este fue inventado para superar las limitaciones del formato base. (cabe destacar que todo lo que este escrito en EBNF, Puede ser escrito en BNF)

Elementos [7]

- símbolos no-terminales; Son elementos que habrán de ser definidos por alguna regla, Se definen usando combinaciones de símbolos terminales, no terminales y metasímbolos, para describir los constructores sintácticos del LP sujeto

- símbolos terminales; Símbolo (ej. una palabra) del lenguaje a definir, se escribe con las comillas

- Metasimbolos

1. ::= Equivalencia
2. | Alternativa
3. (...) Agrupación
4. [...] Aparición opcional
5. {...} Aparición 0, 1 o más veces

La recursividad No es permitida, pero se suple con {…}
Si algún símbolo del lenguaje coincide con un metasímbolo, el símbolo del lenguaje se pone entre ‘comillas simples’

Ejemplo de la EBNF
Sintaxis de los números enteros positivos en notación EBNF

- Numero-entero ::= [Signo] Secuencia-dígitos
- Signo ::= “+”
- Secuencia-dígitos ::= Dígito {Dígito}
- Dígito ::= “0” | “1” | “2” | “3” | “4” | “5” | “6” | “7” | “8” | “9”

1. Las notas de pie de página deberán estar en la página donde se citan. Letra Times New Roman de 8 puntos

¿Para que sirven la BNF y EBNF?

Para expresar cómo debe escribirse una determinada proposición o sentencia en lenguaje determinado.

Mediante BNF se pueden expresar aspectos como qué palabras clave forman la sentencia, qué parámetros son opcionales y cuáles obligatorios, etc. [4]

¿Dónde se usa BNF Y EBNF?

La mayoría de los estándares de lenguaje de programación usan una variante de EBNF para definir la gramática del lenguaje. Esto tiene dos ventajas, pues no puede haber desacuerdo sobre cuál es la sintaxis del lenguaje y facilita mucho la creación de compiladores, ya que el analizador del compilador se puede generar automáticamente con un compilador-compilador como YACC. Además, aparecen en la documentación de cualquier lenguaje formal (Shell script, Java, C#, SQL, etc.), para especificar su sintaxis.

III. CONCLUSIONES

Los Metalenguajes BNF y EBNF son de importancia debido a que estos, son útiles para mostrar el cómo se debe escribir en un lenguaje de programación, siendo una especie de manual con sus reglas, apareciendo en la documentación de estos lenguajes. Por ello se debe tener claridad acerca de estos metalenguajes, pues un desconocimiento de la sintaxis de un lenguaje no permitiría utilizar este de manera correcta

REFERENCIAS

[1] J. Estrada. Notacion BNF. [Online]. Available: https://es.scribd.com/doc/96322761/Notacion-BNF

[2] Backus-Naur form. [Online]. Available: https://en.wikipedia.org/wiki/Backus%E2%80%93Naur_form

[3] L. Garshol. (2008, Aug 22). BNF and EBNF: What are they and how do they work?. [Online]. Available: http://www.garshol.priv.no/download/text/bnf.html

[4] GESTION DE BASE DE DATOS [Online]. Available: http://agrega.educacion.es/repositorio/28042017/b7/es_2017042812_9190105/notacion_bnf_-_v1cc.pdf

[5] M. Navarro .Sintaxis de los lenguajes de programacion. [Online]. Available: http://www.sc.ehu.es/jiwnagom/FLP/FLP-archivos/SintaxisBNF.pdf

[6] U. San Buenaaventura cali. Guía Modelos de Computación. [Online]. Available: http://lidis.usbcali.edu.co/Proyectos/matematicasDiscretas/2016-1/grupo1/guiasmodelos/Guia%20de%20la%20forma%20de%20backus%20naur.pdf

[7] J. Sierra & F. Peinado. (2009). Repaso. Lenguajes formales. [Online]. Available: https://www.fdi.ucm.es/profesor/fpeinado/courses/compiling/Repaso-LenguajesFormales.pdf

[8] F. Tomassetti. (2017, Aug 1). EBNF: How to Describe the Grammar of a Language. [Online]. Available: https://tomassetti.me/ebnf/

[9] UNCPBA. (2009). Ciencias de la Computación I. [Online]. Available: http://www.exa.unicen.edu.ar/catedras/ccomp1/ClaseGLC.pdf

.

1. Las notas de pie de página deberán estar en la página donde se citan. Letra Times New Roman de 8 puntos