Vistazo general de la Notación de Backus-Naur

Overview of the Backus-Naur form

Jacobo Blandón Pineda

*Ingeniería de Sistemas y Computación, Universidad Tecnológica de Pereira*

Correo-e: jacobo.blandon@utp.edu.co

*Resumen*— En ciencias de la computación, la notación de Backus-Naur, también conocida por sus denominaciones inglesas Backus-Naur form (BNF), Backus-Naur formalism o Backus normal form, es una técnica de notación para gramáticas sin contexto, que se usa a menudo para describir la sintaxis de los lenguajes utilizados en computación, tales como lenguajes de programación de computadoras, formatos de documentos, conjuntos de instrucciones y protocolos de comunicación.

***Palabras clave— notación, lenguaje, gramática, sintaxis***

*Abstract*— In computer science, the Backus-Naur form (BNF) also known as Backus-Naur formalism or Backus normal form, is a notation technique for contextless grammars, which is used to often to describe the syntax of languages used in computing, such as computer programming languages, document formats, instruction sets and communication protocols.

*Key Word* — *form, language, grammar, syntax*

### INTRODUCCIÓN

La idea de transcribir la estructura del lenguaje con reglas de reescritura se remonta cuando menos al trabajo del gramático indio Panini, que la utilizó en su descripción de la estructura de palabras del idioma sánscrito. [1]

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.

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 y reglas transformativas. [2]

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 el primer Congreso de Computación Mundial el artículo «The syntax and semantics of the proposed international algebraic language of the Zurich ACM-GAMM Conference». [3]

Peter Naur, en su reporte sobre ALGOL 60 de 1963, identificó la notación de Backus como la Forma Normal de Backus, 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.[4]

Actualmente, la notación BNF es utilizada para describir la sintaxis de la mayoría de lenguajes de programación, y por ello es necesario tomar un vistazo a cómo se conforma esta notación, que es lo que este documento definirá en el curso de su desarrollo.

1. NOTACIÓN BACKUS-NAUR

En ciencias de la computación, la notación de Backus-Naur, también conocida por sus denominaciones inglesas Backus-Naur form (BNF), Backus-Naur formalism o Backus normal form, es una técnica de notación para gramáticas sin contexto, que se usa a menudo para describir la sintaxis de los lenguajes utilizados en computación. [5]

Utiliza reglas que se construyen con tres tipos de símbolos:

* Metasímbolos propios de la BNF
* Símbolos terminales, que se usan en el programa tal como aparece en la regla
* Símbolos no terminales, que deben definirse por alguna regla, usando combinaciones de símbolos terminales, no terminales y metasímbolos

Los suguientes son los metasímbolos usados en la notación:[5]

|  |  |
| --- | --- |
| METASÍMBOLO | SIGNIFICADO |
| ::= | Se define como |
| <> | Delimitadores de nombre de regla |
| | | Opción alternativa |
| (a | b) | a ó b, pero no ambas |
| [algo] | 0 ó 1 vez algo |
| {algo} | 0 o más veces algo |
| ‘xyz’ | Símbolo terminal |
| . | Final de la regla |

Tabla 1. Metasímbolos usados en un BNF.

Y sigue la siguiente estructura

<símbolo> ::= \_expresión\_

Figura 1. Estructura de una regla en la Notación BNF.

Donde *<símbolo>* es un no-terminal, y la *\_expresión\_* consiste de una o más secuencias de símbolos; las secuencias deben separarse por la barra vertical ” |”, indicando una elección, y toda ésta puede ser un posible reemplazo del símbolo a la izquierda.

Los símbolos que nunca aparecen en un lado izquierdo son terminales. Por otro lado, los símbolos que aparecen en el lado izquierdo no son terminales y siempre están encerrados entre el par “<>”. El ":: =" significa que el símbolo de la izquierda debe reemplazarse con la expresión de la derecha.

1. EJEMPLOS

<numero> ::= <entero positivo>

| <entero positivo> “.” <entero positivo>

<entero positivo> ::= <digito>

| <digito> <entero positivo>

<digito> ::= “0” | “1” | “2” | “3” | “4” | “5” | “6” | “7” | “8” | “9”

Como ejemplo, considere este posible BNF para una dirección postal de Colombia:

Figura 2. Gramática de una dirección postal en Colombia

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

<nombre> ::= <personal> <apellido> <trato> <EOL>

| <personal> <nombre>

<personal> ::= <inicial> “.” | <primer nombre> | <primer nombre> <segundo nombre>

<apellido> ::= <primer apellido> | <primer apellido> <segundo apellido>

<dirección física> ::= <nomenclatura> “#” <vía paralela> “-” <vivienda> <EOL>

<apartado postal> ::= <ciudad> “,” <departamento> <código postal> <EOL>

<trato> ::= “Sr.” | “Jr.” | <numeral romano> | “”

Esto se traduce al español como se describe a continuación:

* Una dirección postal está conformada por el nombre, seguido de dirección-física, seguido de apartado-postal.
* Una parte del nombre consta de: una parte personal seguida de un apellido seguido de un trato opcional (Jr., Sr. o número dinástico) y final de línea, o una parte personal seguida de un nombre (esta regla ilustra el uso de la recursión en los BNF, cubriendo el caso de las personas que usan nombres y apellidos múltiples e iniciales múltiples).
* Una parte personal consta de una inicial seguida de un punto, o un primer nombre, o un primer nombre seguido de un segundo nombre.
* Un apellido consta de un primer apellido, o un primer apellido seguido de un segundo apellido.
* La dirección física consta de una nomenclatura de vía seguido de una almohadilla, seguido de la vía paralela seguido por el número de vivienda y seguido por un final de línea.
* Un apartado postal que consta de una ciudad seguido de un departamento, seguido del código postal y seguido de un final de línea.
* Un trato que consta de un sufijo, como “Sr.” O “Jr.”, o de un numeral romano, o de una cadena vacía.

Otro ejemplo, un BNF para hacer representación de números positivos con decimal:

Figura 3. Gramática para representar números con decimales

Se traduce al español como se describe a continuación:

* Un número consta de un entero positivo, o de un entero positivo seguido de un punto, seguido de un entero positivo.
* Un entero positivo consta de un dígito, o de un dígito seguido de un entero positivo.
* Un dígito puede ser “0”, o “1”, o “2”, o “3”, o “4”, o “5”, o “6”, o “7”, o “8”, o “9”.

1. BNF EXTENDIDO

El BNF extendido, o Extended BNF (EBNF) añade extensiones a la notación Backus-Naur. El primer EBNF fue desarrollado por Niklaus Wirth incorporando algunos de los conceptos (con una sintaxis y notación diferentes) de la notación de sintaxis de Wirth. Sin embargo, muchas variantes de EBNF están en uso. La Organización Internacional para la Estandarización adoptó un estándar EBNF (ISO / IEC 14977) en 1996.[6]

Estos son los símbolos definidos en el estándar ISO/IEC 14977.[6]

|  |  |
| --- | --- |
| METASÍMBOLO | SIGNIFICADO |
| = | Definición |
| , | Concatenación |
| ; | Terminación |
| | | Alternación |
| [ ] | Opcional |
| { } | Repetición |
| ( ) | Agrupación |
| “ ” | Cadena terminal |
| ‘ ’ | Cadena terminal |
| (\*\*) | Comentario |
| ? ? | Secuencia especial |
| - | Excepción |

Tabla 2. Metasímbolos del EBNF según ISO/IEC 14977

1. EJEMPLOS

Volvamos al ejemplo anterior de representación de números enteros, pero esta vez tomaremos cualquier número entero con decimal con signo.

entero = “0” | [“-”], número con decimal ;

número con punto decimal = número natural | número natural, “.”, número natural ;

número natural = dígito excluyendo a 0, { dígito } ;

dígito = “0” | dígito excluyendo a 0 ;

dígito excluyendo a 0 = “1” | “2” | “3” | “4” | “5” | “6” | “7” | “8” | “9” ;

Figura 4. Gramática de un número entero con signo

Se traduce al español de este modo:

* Un entero, que puede ser un “0” o un signo negativo opcional, seguido de un número con punto decimal.
* Un número con punto decimal que consta de un número natural, o un número natural seguido de un punto, seguido de un número natural.
* Un número natural que consta de un dígito excluyendo a 0 seguido de un dígito que se puede repetir.
* Un dígito que consta de un “0” o un dígito excluyendo a 0.
* Un dígito excluyendo a 0 que puede ser “1”, “2”, “3”, “4”, “5”, “6”, “7”, “8”, o “9”.

1. CONCLUSIONES

La creación de una notación que permita describir la gramática de los lenguajes de programación sirve como una introducción a las reglas sintácticas que casi todos, por no decir todos, los lenguajes de programación siguen.

Estas notaciones muestran principios básicos de la informática y programación, al tener la posibilidad de mostrar recursión y el uso de condicionales, ya que ciertos símbolos sólo son definidos dependiendo de la condición en la que se encuentre la información ingresada, como la presencia de un signo negativo.

Estos también pueden describir la forma en la que un autómata finito funciona, o más bien, un autómata finito hace demostración del proceso que es descrito con las notaciones BNF y EBNF.

REFERENCIAS

[1] J. J. O’Connor and E. F. Robertson, “Panini biography”, *School of Mathematics and Statistics*, University of St Andrews, Scotland, nov. 2000. Available: https://www-history.mcs.st-andrews.ac.uk/Biographies/Panini.html

[2] N. Chomsky, "Three models for the description of language". *IRE Transactions on Information Theory*, IEEE, vol. 2, issue 3, pp. 113–124, sept. 1956.

[3] J. W. Backus, “The Syntax and Semantics of the Proposed International Algebraic Language of the Zurich ACM-GAMM Conference.”, *Proceedings of the International Conference on Information Processing*, UNESCO, 1959, pp.125-132.

[4] D. E. Knuth, “Backus normal form vs. Backus Naur form”, *Communications of the ACM*, vol. 7, issue 12, pp. 735–736, dec. 1964.

[5] F. Peinado, J. L. Sierra, “Repaso. Lenguajes formales”, *Procesadores de Lenguaje, Facultad de Informática,* Universidad Complutense de Madrid, 2010. Available: https://www.fdi.ucm.es/profesor/fpeinado/courses/compiling/Repaso-LenguajesFormales.pdf

[6] *Information technology – Syntactic metalanguage - Extended BNF*, ISO/IEC 14977, Dec.. 1996. Available: https://standards.iso.org/ittf/PubliclyAvailableStandards/s026153\_ISO\_IEC\_14977\_1996(E).zip