

Backus Naur

RESUMEN

La notación de Backus-Naur, también conocida por sus denominaciones inglesas Backus-Naur form (BNF), Backus-Naur formalism o Backus normal form, es un metalenguaje usado para expresar gramáticas libres de contexto: es decir, una manera formal 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.

PALABRAS CLAVES: AFD, Notación, metalenguaje, teoría, comunicación, sintaxis, semántica, lenguaje, programación, BNF

ANDRES CASTAÑEDA

Ingeniero En sistemas
Primer semestre
Universidad Tecnológica de Pereira
Andresfelipe.castaneda@utp.edu.co

1. INTRODUCCIÓN

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

```
<simbolo> ::= <expresión con símbolos>
```

donde <simbolo> 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.

2. CONTENIDO

Como ejemplo, considere este BNF para una dirección postal de los EE.UU.

```
<dirección postal> ::= <nombre> <dirección>
<apartado postal>
<nombre> ::= <personal> <apellido> [<trato>]
<EOL>
| <personal> <nombre>
```

```
<personal> ::= <primer nombre> | <inicial> "."
<dirección> ::= [<dpto>] <numero 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).

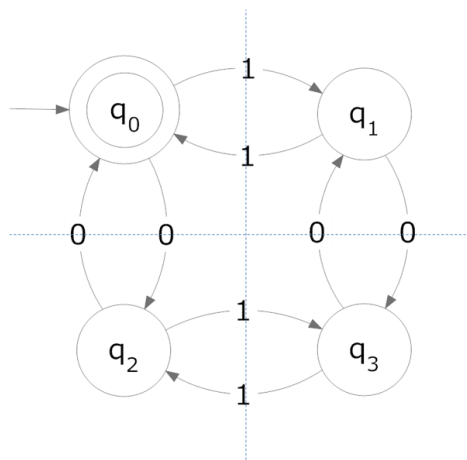
AFD

Un *autómata finito determinista* (abreviado AFD) es un *autómata finito* que además es un *sistema determinista*; es decir, para cada estado $q \in Q$ en que se encuentre el *autómata*, y con cualquier símbolo $a \in \Sigma$ del alfabeto leído, existe siempre a lo más una transición posible $\delta(q, a)$.

En un AFD no pueden darse ninguno de estos dos casos:

- Que existan dos transiciones del tipo $\delta(q, a) = q_1$ y $\delta(q, a) = q_2$, siendo $q_1 \neq q_2$;
- Que existan transiciones del tipo $\delta(q, \epsilon)$, salvo que q sea un estado final, sin transiciones hacia otros estados.

Un tipo interesante de autómatas finitos deterministas son los llamados *acíclicos* y un ejemplo de éstos son los *tries*.



AFND

Un *autómata finito no determinista* (abreviado AFND) es aquel que, a diferencia de los autómatas finitos deterministas, posee al menos un estado $q \in Q$, tal que para un símbolo $a \in \Sigma$ del alfabeto, existe más de una transición $\delta(q, a)$ posible.

Haciendo la analogía con los AFDs, en un AFND puede darse cualquiera de estos dos casos:

- Que existan transiciones del tipo $\delta(q, a) = q_1$ y $\delta(q, a) = q_2$, siendo $q_1 \neq q_2$;

- Que existan transiciones del tipo $\delta(q, \epsilon)$, siendo q un estado no-final, o bien un estado final pero con transiciones hacia otros estados.

Cuando se cumple el segundo caso, se dice que el *autómata* es un *autómata finito no determinista con transiciones vacías o transiciones ϵ* (abreviado AFND- ϵ). Estas transiciones permiten al *autómata* cambiar de estado sin procesar ningún símbolo de entrada.

Formalmente, se distingue de la 5-tupla que define a un *autómata finito determinista* en su función de transición. Mientras en un AFD esta función se define de la siguiente manera:

$$\delta : Q \times \Sigma \rightarrow$$

$$Q$$

en un AFND se define como:

$$\delta : Q \times \Sigma \rightarrow$$

$$\mathcal{P}(Q)$$

Para el caso de los AFND- ϵ , se suele expresar la función de transición de la forma:

$$\delta : Q \times (\Sigma \cup$$

$$\epsilon) \rightarrow \mathcal{P}(Q)$$

donde $P(Q)$ es el conjunto potencia de Q .

Esto significa que los autómatas finitos deterministas son un caso particular de los no deterministas, puesto que Q pertenece al conjunto $P(Q)$.

La interpretación que se suele hacer en el cómputo de un AFND es que el *autómata* puede estar en varios estados a la vez, generándose una ramificación de las configuraciones existentes en un momento dado. Otra interpretación puede ser imaginar que la máquina "adivina" a qué estado debe ir, eligiendo una transición entre varias posibles.

Note finalmente que en un *autómata finito no determinista* podemos aceptar la existencia de más de un nodo inicial, relajando aún más la definición original.

N. BIBLIOGRAFÍA

Referencias de páginas webs

1. <https://www.wikipedia.org/>