Gramáticas

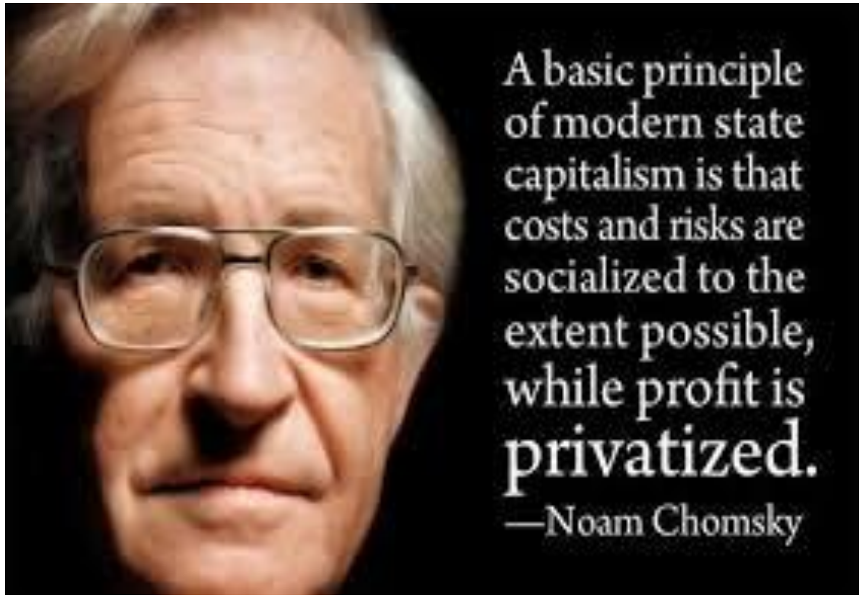
Contenido

Introducción

- Si nos referimos a los lenguajes naturales el concepto de gramática es muy antiguo. Los primeros trabajosa parecen en la India durante los comienzos del primer milenio antes de Cristo,
- Se alcanzó el máximo apogeo con Panini (siglos VII y VI a.C.). Al mismo tiempo en Grecia se desarrollaba una corriente de investigación gramatical, cuyo máximo representante sería Pitágoras.
- Sin embargo, el concepto de gramática desde un punto de vista formal tiene su origen en los trabajos de Chomsky a mediados del siglo XX.

Relación entre la jerarquía de las gramáticas de Chomsky

Gramáticas	Lenguajes	Máquinas	Problemas
			No computables
Tipo 0 (G. sin restricción)	Recursivamente enumerables	Máquinas de Turing	Computables
Tipo 1 (G. sensible al contexto)	Lenguajes sensibles al contexto	Autómatas lineales acotados	
Tipo 2 (G. Libre de contexto)	Lenguajes independientes del contexto	Autómatas de pila	
Tipo 3 (G. Regular)	Lenguajes regulares	Autómatas finitos deterministas	Expresiones regulares



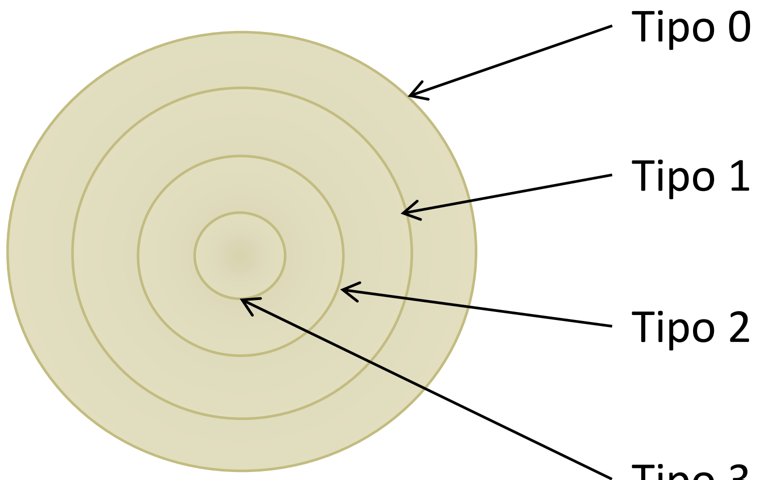
4

Jerarquía de las gramáticas de Chomsky

- Chomsky clasificó los lenguajes en una jerarquía inclusiva

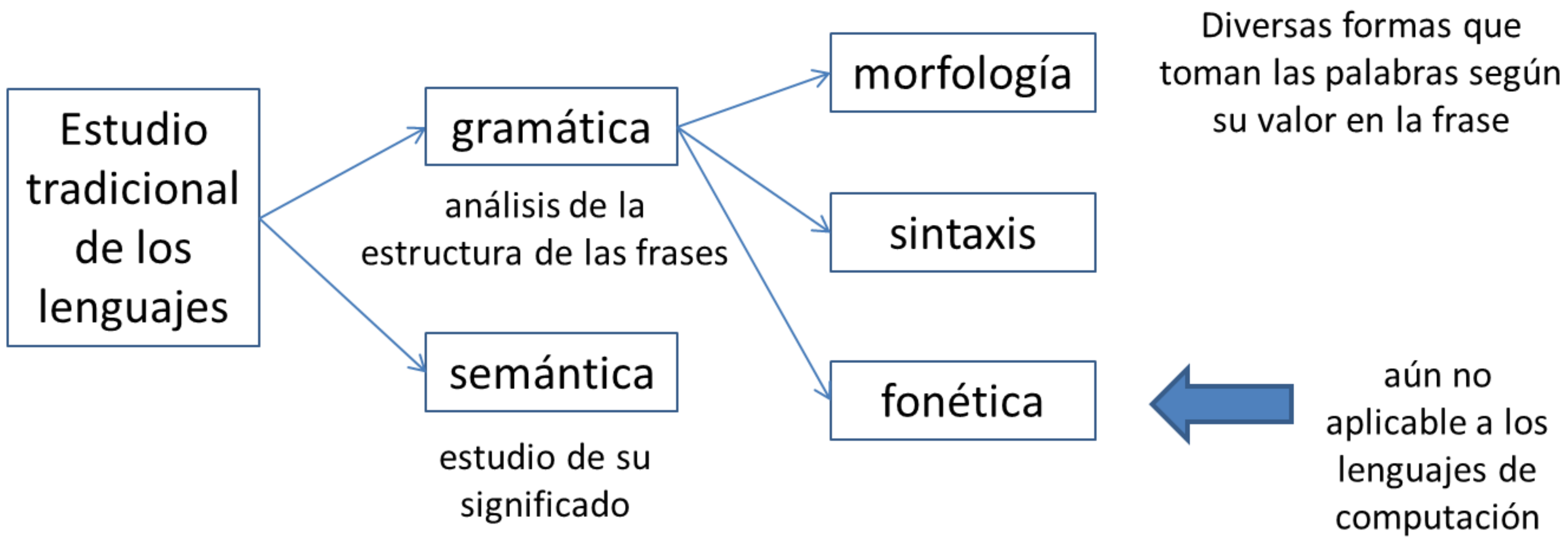
Teoría de las máquinas abstractas

Teoría de lenguajes formales



Es una jerarquía de cuatro grados (cada grado contiene a todos los siguientes)

- En 1950 el lingüista norteamericano Avram Noam Chomsky introdujo la *teoría de las gramáticas transformacionales* o *teoría de lenguajes formales,* que convirtió la lingüística en una ciencia y proporcionó una herramienta que no sólo podía aplicarse a los lenguajes naturales, sino que facilitaba el estudio y la formalización de los lenguajes para la programación de computadoras (1960).



Gramática

- Es el estudio de las reglas y principios que regulan el uso de las lenguas y la organización de las palabras dentro de una oración.

- También se denomina así al conjunto de reglas y principios que gobiernan el uso de un lenguaje muy determinado; así, cada lenguaje tiene su propia gramática.

- La gramática es un ente formal para especificar, de una manera finita, el conjunto de cadenas de símbolos que constituyen un lenguaje.
  - Permite definir un lenguaje mediante reglas que nos permiten generar o producir cadenas de un lenguaje.
  - Estas gramáticas son similares a las gramáticas de los lenguajes naturales, pero mucho más restrictivas y sencillas.
  - Un ejemplo de regla de una gramática:

    Oración → Sujeto predicado
  - Estas reglas se suelen llamar **reglas de reescritura**: el símbolo *Oración* se puede reescribir por el símbolo *Sujeto* seguido del símbolo *Predicado*.

El concepto de gramática formal. Definiciones previas.

Producción

Sea Σ un alfabeto, llamamos producción (o regla) definida sobre ese alfabeto a un par ordenado de las palabras (***x***, ***y***) donde ***x***, ***y*** $\in \Sigma^*$. Tiene el significado de que si se encuentra **x** como parte de cualquier palabra **v**, se puede sustituir **x** por **y** en **v**, lo que permite transformar palabras en otras. A las producciones también se les llama reglas de derivación.

Se representa ***x :: = y***.

Ejemplo: en $\sum = \{1, 0\}$, se podrían tener las producciones: (000 ::= 010) ó (10 ::= 01)

Producción compresora

Se dice que una producción es compresora si la longitud de su parte derecha es menor que la de la parte izquierda.

El concepto de gramática formal. Definiciones previas.

Derivación directa

Derivación directa, $v \rightarrow w$: es la aplicación de una producción (x ::= y) a una palabra **v** para convertirla en otra palabra **w** donde

$v = zxu$, $w = zyu$ (***v***, ***w***, ***z***, ***u*** $\in \Sigma^*$)

Se cumple que, para cada producción (x ::= y), existe una derivación directa de **x** a **y**: $x \rightarrow y$, lo que se deduce de lo anterior $z = u = \lambda$.

Ejemplo: Con las producciones P_1 = (000 ::= 010) y P_2 = (10 ::= 01) y la palabra 1000, se pueden hacer las siguientes derivaciones directas:

$1000 \xrightarrow{P_1} 1010$, $1010 \xrightarrow{P_2} 0110$, $0110 \xrightarrow{P_2} 0101$, y $0101 \xrightarrow{P_2} 0011$

El concepto de gramática formal. Definiciones previas.

Derivación $v \xrightarrow{*} w$

Sea **Σ** un alfabeto, **P** un conjunto de producciones definidas sobre ese alfabeto y ***v***, ***w*** **∈ Σ***.

Decimos que v produce directamente a w, o que w deriva directamente de v, si $\exists w_0, w_1, \ldots, w_m \in \Sigma^*$ tales que

$$
\begin{aligned}
v = w_0 &\rightarrow w_1 \\
w_1 &\rightarrow w_2 \\
&\cdots \\
w_{m-1} &\rightarrow w_m = w
\end{aligned}
$$

Derivación: es la aplicación de una secuencia de producciones a una palabra.

Ejemplo: partiendo del ejemplo anterior, la ejecución continuada de las derivaciones directas, proporciona una derivación de la palabra 1000 a la palabra 0011: $1000 \xrightarrow{*} 0011$.

Otra posible derivación es $1000 \xrightarrow{*} 0110$

El concepto de gramática formal. Definiciones previas.

Derivación más a la izquierda

Si utilizamos en cada derivación directa la producción aplicada a los símbolos más a la izquierda de la palabra, se dice que usamos derivación más a la izquierda.

Ejemplo: partiendo del ejemplo anterior de derivación, la derivación más a la izquierda sería:

$$1000 \xrightarrow{P_2} 0100 \xrightarrow{P_2} 0010 \xrightarrow{P_2} 0001 \xrightarrow{P_1} 0101 \xrightarrow{P_2} 0011$$

Con $P_1 = (000 ::= 010)$ y $P_2 = (10 ::= 01)$

El concepto de gramática formal. Definiciones previas.

Derivación más a la derecha

Si utilizamos en cada derivación directa la producción aplicada a los símbolos más a la derecha de la palabra, se dice que usamos derivación más a la derecha.

Ejemplo: partiendo del ejemplo anterior de derivación, la derivación más a la derecha sería:

$$1000 \xrightarrow{P_1} 1010 \xrightarrow{P_2} 1001 \xrightarrow{P_2} 0101 \xrightarrow{P_2} 0011$$

Con $P_1 = (000 ::= 010)$ y $P_2 = (10 ::= 01)$

Ejemplo 1 de uso.

Estamos familiarizados con el concepto tradicional de gramática, y esto nos permite saber (de forma intuitiva) qué es correcto y qué no lo es en un lenguaje natural.

La gramática debe describir la estructura de las frases y de las palabras de un lenguaje. Veamos una serie de reglas muy sencillas que nos permitirán comprobar que la frase “el perro corre deprisa” es correcta.

Ejemplo 1 de uso.

Reglas gramaticales:

1. < sentencia >::=< sujeto >< predicado >
2. < sujeto >::=< articulo >< nombre >
3. < predicado >::=< verbo >< complemento >
4. < predicado >::=< verbo >
5. < articulo > ::= *el*
6. < articulo > ::= *la*
7. < nombre > ::= *perro*
8. < nombre > ::= *gata*
9. < verbo > ::= *corre*
10. < verbo > ::= *come*
11. < complemento > ::= *deprisa*
12. < complemento > ::= *mucho*

Ejemplo 1 de uso.

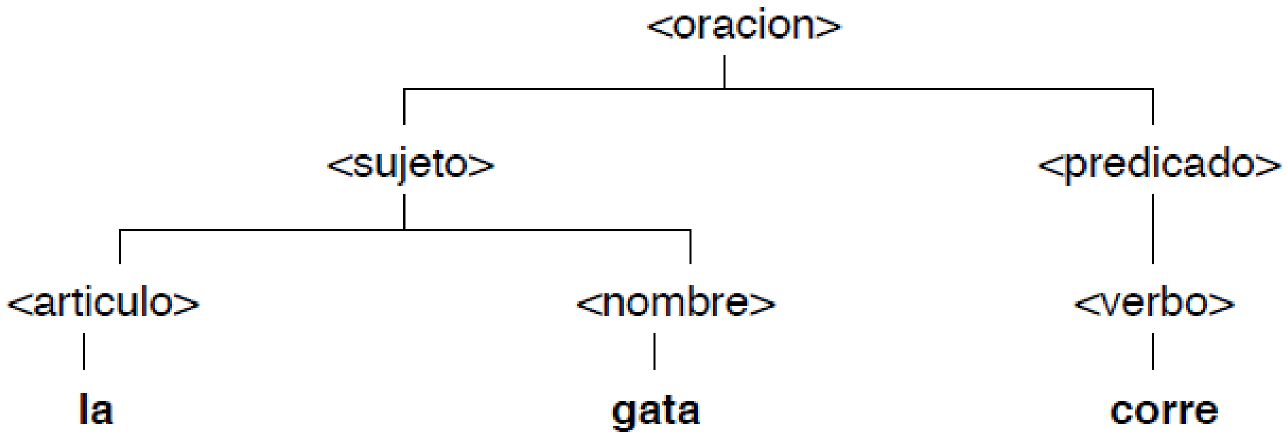
Estas reglas pueden ser consideradas como un conjunto de producciones. Si utilizamos algunas de estas producciones para llevar a cabo derivaciones a partir del item $<$ sentencia $>$ podemos llegar a obtener frases como: "*el perro corre deprisa*", "*la gata come mucho*" o "*la gata corre*". Sin embargo, nunca podríamos llegar a construir la frase "*mucho deprisa perro*".

Veamos, paso a paso, como se podría generar la frase "la gata corre" a partir del símbolo $<$ *sentencia* $>$. En cada fase del proceso hemos destacado en negrita el símbolo que se transforma.

Ejemplo 1 de uso.

$$\begin{array}{ll}
\mathit{Aplicando\ la\ pr.\ 1} & <\mathrm{sentencia}> \longrightarrow <\mathbf{sujeto}> <\mathrm{predicado}> \\
\mathit{Aplicando\ la\ pr.\ 2} & <\mathrm{sentencia}> \xrightarrow{*} <\mathrm{articulo}><\mathrm{nombre}> <\mathbf{predicado}> \\
\mathit{Aplicando\ la\ pr.\ 4} & <\mathrm{sentencia}> \xrightarrow{*} <\mathbf{articulo}> <\mathrm{nombre}><\mathrm{verbo}> \\
\mathit{Aplicando\ la\ pr.\ 6} & <\mathrm{sentencia}> \xrightarrow{*} \mathrm{la}<\mathbf{nombre}> <\mathrm{verbo}> \\
\mathit{Aplicando\ la\ pr.\ 8} & <\mathrm{sentencia}> \xrightarrow{*} \mathrm{la\ gata}<\mathbf{verbo}> \\
\mathit{Aplicando\ la\ pr.\ 9} & <\mathrm{sentencia}> \xrightarrow{*} \mathrm{la\ gata\ corre}
\end{array}$$

Sin embargo, la forma más habitual de representar este mismo proceso de generación de una cadena de símbolos es mediante un árbol de derivaciones:



Ejemplo 2 de uso

Aplicaremos el mismo método para definir un fragmento de un lenguaje de programación. Pretendemos describir cómo son las instrucciones que permiten asignar el valor de una expresión a una variable en un lenguaje como C.

Reglas gramaticales:

1. $< asignacion > ::= < variable > ' =' < expresion >$
2. $< expresion > ::= < numero >$
3. $< expresion > ::= < variable >$
4. $< expresion > ::= < expresion > '+' < expresion >$
5. $< expresion > ::= < expresion > '*' < expresion >$

Ejemplo 2 de uso

Si consideramos que A, B y C pueden ser considerados como $< variable >$ y que 2 y 4 pueden ser considerados como $< numero >$, es fácil comprobar que a partir del símbolo $< asignacion >$ y utilizando diferentes producciones podemos llegar a construir instrucciones como:

$A = B + C$

$B = B * 2$

$C = C + 4$

Sin embargo, no podríamos construir sentencias como $A = + 2 * / + 4$ o $4{=}A$

Es decir, en los ejemplos anteriores podemos ver que hay construcciones gramaticalmente correctas y otras que no lo son.

Definición formal de gramática

Existen varios tipos de Gramáticas, las que mas se usan en computación son las gramáticas generativas, definidas por Noam Chomsky.

Se llama gramática formal definida sobre un alfabeto ∑ a una tupla de la forma:

$$G = \{\Sigma_T, \Sigma_N, S, P\}$$

donde :

Σ_T es el alfabeto de símbolos terminales
Σ_N es el alfabeto de símbolos no terminales
(aparecen en los ejemplos encerrados entre **<>**)
S es el símbolo inicial de la gramática
P es un conjunto de producciones gramaticales

Hay que tener en cuenta:

$$S \in \Sigma_N$$

$$\Sigma_T \cap \Sigma_N = \emptyset$$

$$\Sigma = \Sigma_T \cup \Sigma_N$$

Ejemplo 1. Una gramática para un número natural.

$\sum_T$ = { +, -, 0, 1, . . . , 9 }

$\sum_N$ = { < Signo >, < Dígitos >, < Número >, < Carácter > }

S = < Numero >

$$P = \left\{ \begin{array}{l} <Numero>::=<Signo><Digito> \\ <Signo>::=+ \\ <Signo>::=- \\ <Digito>::=<Caracter><Digito> \\ <Digito>::=<Caracter> \\ <Caracter>::=0 \\ <Caracter>::=1 \\ \qquad\qquad \cdots \\ <Caracter>::=9 \end{array} \right\}$$

Con esta gramática, y a partir del símbolo número < Número >, podemos generar cualquier número natural, siempre que vaya precedido por un signo. Por ejemplo:-57, +5, -4999, etc.

Forma normal de Backus (BNF)

Es posible establecer una notación simplificada para las reglas de producción. Si existen dos reglas de la forma

$$u::=v$$

$$u::=w$$

se pueden representar de la forma:

$$u::=v \mid w$$

Esta forma de representar las reglas de producción recibe el nombre de "forma normal de Backus", (BNF por sus siglas en Inglés: Backus Normal Form).

Pasando el ejemplo 1 a BNF

$\sum_T$ = { +, -, 0, 1, . . . , 9 }
$\sum_N$ = { < Signo >, < Dígitos >, < Número >, < Carácter > }
S = < Numero >

$$P = \left\{ \begin{array}{l} <Numero>::=<Signo><Digito> \\ <Signo>::=+ \\ \qquad\qquad\quad |- \\ <Digito>::=<Caracter><Digito> \\ \qquad\qquad\quad |<Caracter> \\ <Caracter>::=0 \\ \qquad\qquad\quad |1 \\ \qquad\qquad\quad |\cdots \\ \qquad\qquad\quad |9 \end{array} \right\}$$

Ejemplo 2 usando BNF

G_1 = { (0, 1), {A, B}, A, P }

Donde

P = { (A ::= 1B1), (A ::= 0B0), (B ::= A), (B ::= 1), (B ::= 0), (B ::= λ) }

Usando BNF:

$$P = \left\{ \begin{array}{l} A ::= 1B1 \mid 0B0 \\ B ::= A \mid 1 \mid 0 \mid \lambda \end{array} \right\}$$

Conceptos de gramáticas

Forma sentencial

La palabra x es la forma sentencial si existe una derivación desde el axioma hasta esa palabra; es decir: $S \xrightarrow{*} x$

Ejemplo: Dada la gramática anterior G1, las siguientes son formas sentenciales:

- 010: $A \rightarrow 0B0 \rightarrow 010$
- 1A1: $A \rightarrow 1B1 \rightarrow 1A1$
- 1001: $A \rightarrow 1B1 \rightarrow 1A1 \rightarrow 10B01 \rightarrow 1001$

$$P = \left\{ \begin{array}{l} A ::= 1B1 \mid 0B0 \\ B ::= A \mid 1 \mid 0 \mid \lambda \end{array} \right\}$$

Conceptos de gramáticas

Sentencia

La palabra x es sentencia si es forma sentencial y todos sus símbolos pertenecen al alfabeto de símbolos terminales, es decir, $S \xrightarrow{*} x$, y x ε Σ_T^*.

Ejemplo: Dada la gramática anterior G1, 010 y 1001 son sentencias mientras que 1A1 no lo es.

- 010: A → 0B0 → 010
- 1A1: A → 1B1 → 1A1
- 1001: A → 1B1 → 1A1 → 10B01 → 1001

P = { A ::= 1B1 | 0B0
B ::= A | 1 | 0 | λ }

Lenguaje generado por una gramática

L(G) es un conjunto de todas las sentencias de la gramática; es decir, todas las palabras que se pueden obtener a partir del axioma (*S*) de la gramática por la aplicación de las derivaciones:

$$L(G) = \{ x \mid S \xrightarrow{*} x, x \in \Sigma_T^* \}$$

Ejemplo: ¿Cuál sería el lenguaje generado por la gramática anterior G1?

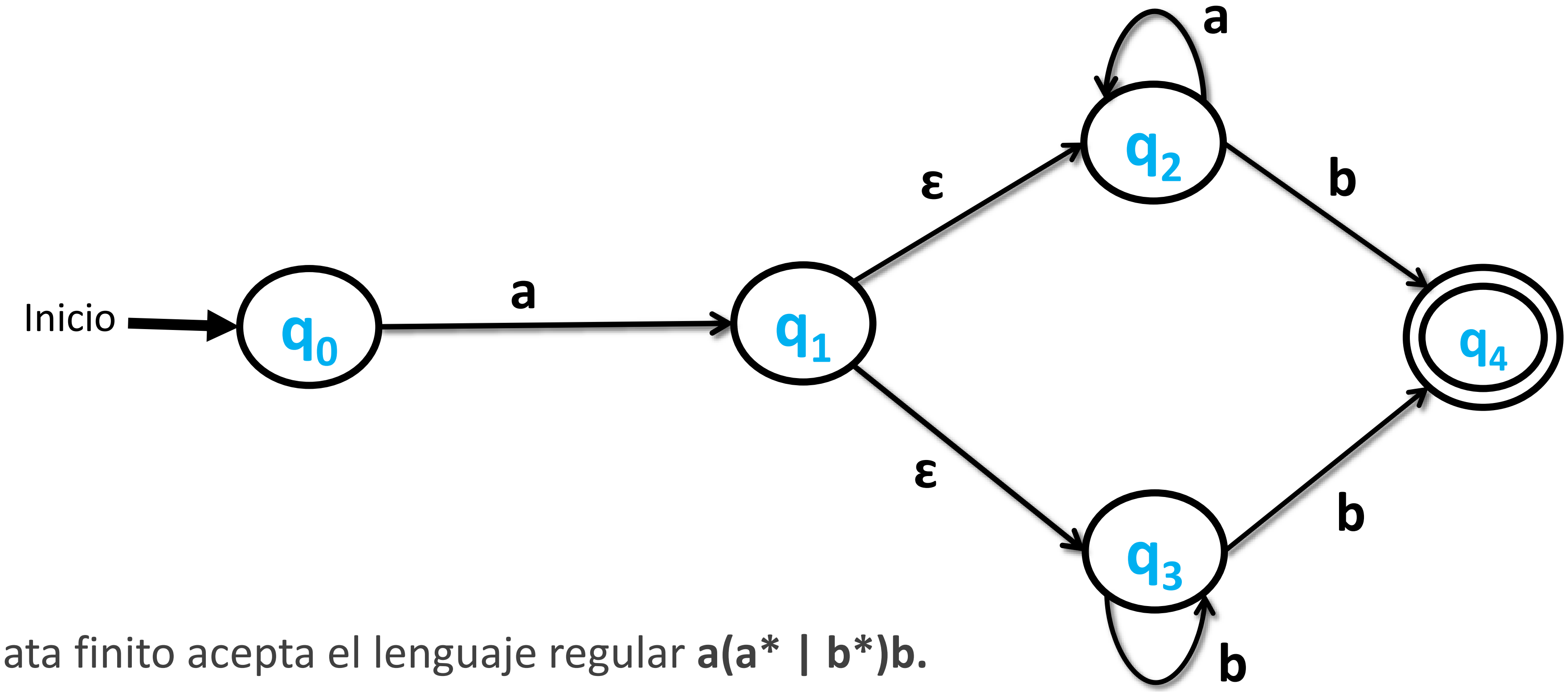
Lenguaje generado por una gramática

El lenguaje generado por la gramática G_1 sería:

$$L_{G1} = \{ 11, 101, 111, 00, 000, 101, 1001, 111, 0000, 0110, \ldots\}$$

Como se puede observar, el lenguaje obtenido es el de las palabras binarias simétricas o palíndromas ($L = \{ x \mid x = x^{-1} \}$)

Ejemplo 2: Hallar la gramática del siguiente autómata



Este autómata finito acepta el lenguaje regular **a(a* | b*)b.**

Ejemplo 2

- Todas las cadenas del lenguaje precedente estarán formadas por una “**a**” seguida de alguna "parte final".
- Si representamos esa “parte final” con "**E**" y el símbolo inicial con "**S**" podemos entonces crear la siguiente regla de producción:

$$S \rightarrow aE$$

Ejemplo 2

- La parte final "E" de una cadena estará formada por uno de los dos caminos **a**'s o **b**´s del autómata. Por lo tanto, para indicar las múltiples posibilidades que hay para E podemos crear las dos reglas siguientes:

$$E \rightarrow A$$

$$E \rightarrow B$$

Ejemplo 2

- Las 2 listas de **a**'s y **b**'s se pueden expresar de la siguiente manera:

$$\left.\begin{array}{l} A \rightarrow aA \\ A \rightarrow b \end{array}\right\} \text{ Para indicar que una cadena de } \mathbf{a}\text{'s va seguida de una “}\mathbf{b}\text{”}$$

$$\left.\begin{array}{l} B \rightarrow bB \\ B \rightarrow b \end{array}\right\} \text{ Para indicar que una cadena de } \mathbf{b}\text{'s va seguida de otra “}\mathbf{b}\text{”}$$

Ejemplo 2

- Producciones:

Producciones

$S \rightarrow aE$

$E \rightarrow A$

$E \rightarrow B$

$A \rightarrow aA$

$A \rightarrow b$

$B \rightarrow bB$

$B \rightarrow b$

Abreviando con BNF

$S \rightarrow aE$

$E \rightarrow A \mid B$

$A \rightarrow aA \mid b$

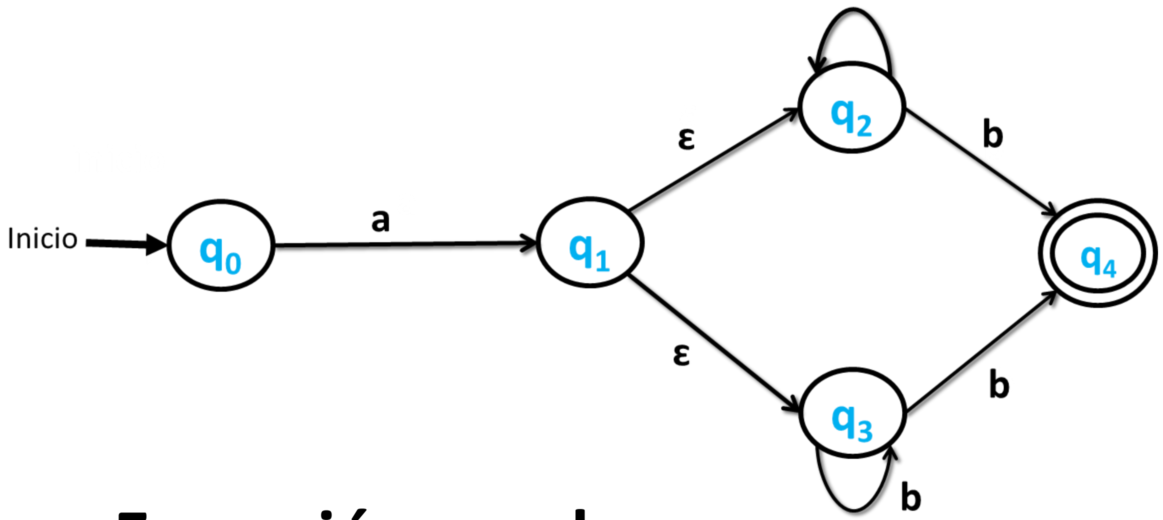
$B \rightarrow bB \mid b$

Ejemplo 2

$G = (\Sigma_N, \Sigma_T, S, P)$

- $\Sigma_N = \{S, E, A, B\}$
- $\Sigma_T = \{a, b\}$
- **P=**

$$
\left\{
\begin{array}{l}
S \rightarrow aE \\
E \rightarrow A \mid B \\
A \rightarrow aA \mid b \\
B \rightarrow bB \mid b
\end{array}
\right\}
$$



Expresión regular:

a(a* | b*)b

Ejercicios

1. Con base en la siguiente gramática:

G = ({0,1,2}, {A,B}, A, { (A::=0B), (A::=2), (B::=0A), (B::=1)})

a) Obtener derivaciones de las palabras:

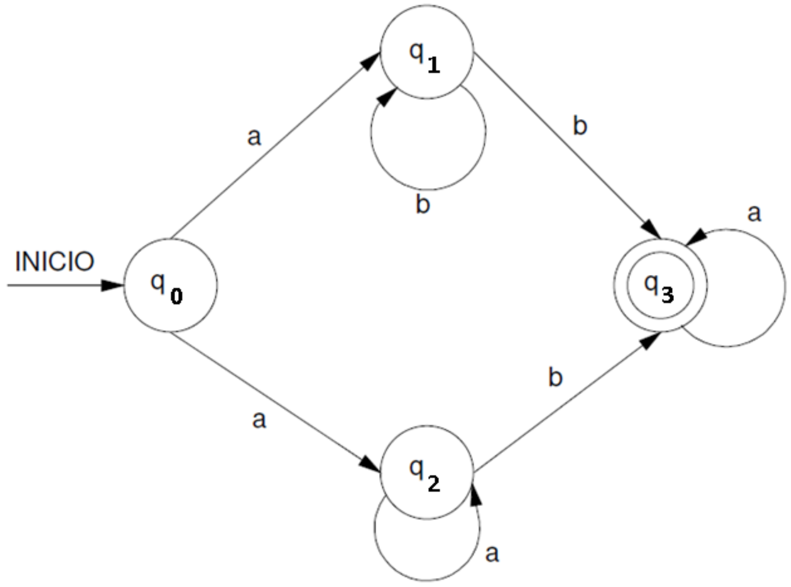
-002

-0001

b) Obtener el lenguaje que genera

Ejercicios

Hallar la gramática para el siguiente AFND:



Expresión regular:

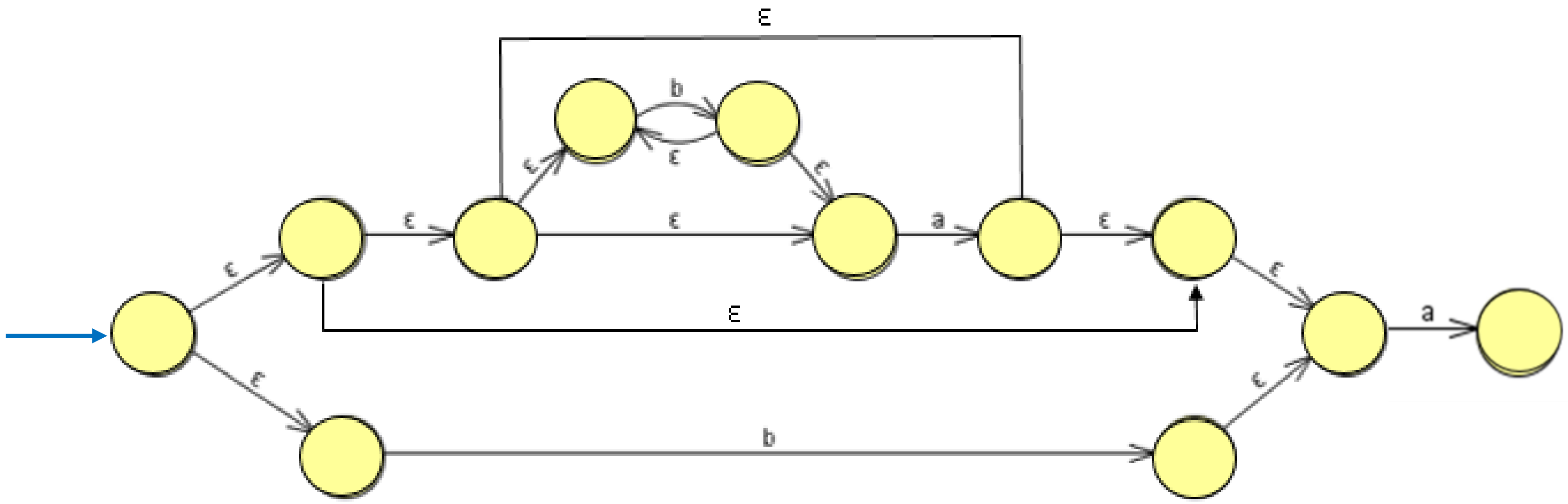
$a(b^*b|a^*b)a^*$

o

$a(b^*|a^*)ba$

Ejercicios

Hallar la gramática para el siguiente AFND:



Expresión regular:

(b|(b*a)*)a