UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS

Inteligencia Artificial

Lenguajes y Gramáticas

- Hugo David Calderón
- Willy Ugarte Rojas
- Jorge Valverde Rebaza

Lenguaje Formal

Es un conjunto, finito o infinito, de cadenas definidas sobre un alfabeto finito. Ejemplo.

L1 = {ab, aabb, aaabbb}

L2 = {001, 011, 111}.

Lenguaje Formal

- Siendo (Alfabeto). Un conjunto finito de símbolos. $\mathbf{\Sigma}_1$ = {a, b, c}.
- Donde cadena o palabra es una serie arbitraria de símbolos unidos; por ejemplo: aaabbbccc

Lenguaje Formal

Se sigue que existe un espacio o universo infinito de lenguajes. Sin embargo, no todos los lenguajes que pueblan este universo son del interés de la teoría de los lenguajes formales.

Lenguaje Formal

Son interesantes aquellos lenguajes en los que se observa que se sigue alguna pauta regular en la construcción de las cadenas. Desde este punto de vista, el lenguaje L_3 es digno de estudio, pero no L_4:

L_3 = {abc, aabbccc, aaabbbccc, ...}

L_4 = {a, cab, bdac, ...}

Lenguaje Formal

El lenguaje formal utiliza dispositivos mecánicos capaz de determinar si una cadena dada pertenece o no a un lenguaje determinado, este procedimiento de decisión puede ejecutarse de manera más o menos eficiente, utilizando más o menos recursos, y no todos los lenguajes son iguales a este respecto.

Lenguaje Formal

Los dos puntos de vista anterior son los que definen los objetivos principales de las dos grandes disciplinas matemáticas ocupadas de poner orden en el universo de los lenguajes formales:

- Teoría de los Lenguajes Formales (sensu strictu) y
- La Teoría de la Complejidad Computacional.

Teoría de lenguajes formales

La Teoría de los lenguajes formales estudia los lenguajes prestando atención únicamente a sus propiedades estructurales, definiendo clases de complejidad estructural y estableciendo relaciones entre las diferentes clases.

Teoría de la complejidad computacional

La Teoría de la complejidad computacional estudia los lenguajes prestando atención a los recursos que utilizaría un dispositivo mecánico para completar un procedimiento de decisión, definiendo así diferentes clases de complejidad computacional y las elaciones que existen entre ellas.

Clausura transitiva o de Kleene

La clausura transitiva o, también,clausura de Kleene de un alfabeto . **Σ**, escrito . **Σ***, es el conjunto de todas las cadenas sobre . **Σ**. Por tanto:

{a}* = {ε, a, aa, aaa, ...}

{a, b}* = {ε, a, b, aa, bb, ab, ba, aaa, ...}

Clausura transitiva o de Kleene

$L_1 = \{x \in \{a, b\}^* \mid |x| <= 2\}$

$L_2 = \{xy \mid x \in \{a, aa\} \text{ e } y \in \{b, bb\}\}$

$L_3 = \{a^n b^n \mid n >= 1\}$

$L_4 = \{(ab)n \mid n >= 1\}$

Expresiones regulares

Una expresión regular es una fórmula **r** cuya denotación es un lenguaje L(r) definido sobre un alfabeto **Σ**. Hay dos tipos de expresiones regulares: la expresiones regulares atómicas y las expresiones regulares compuestas.

Expresiones regulares atómicas

La sintaxis y la denotación de las expresiones regulares atómicas son:

1) Cualquier literal a, tal que a ϵ **Σ** es una expresión regular cuya denotación es L(a) = {a}.
2) El símbolo especial ε es una expresión regular cuya denotación es L(ε) = {ε}.
3) El símbolo especial ϕ es una expresión regular cuya denotación es L(ϕ) = { }.

Expresiones regulares compuestas

La sintaxis y la denotación de las expresiones regulares compuestas son:

1) (r1r2) es una expresión regular cuya denotación es $L(_{r1r2}) = L(_{r1})L(_{r2})$.
2) (r1 + r2) es una expresión regular cuya denotación es $L(r1 + r2) = L(r1) \cup L(r2)$.
3) (r)* es una expresión regular cuya denotación es $L((r)^*) = (L(r))^*$.

Expresiones regulares

Ejemplos:

A= aa* denotación A={a,aa,aaa,…}

B= (a+b)* denotación B={a,b}*

C= a* + b* denotación $\{a_n, b_n \mid n>=0\}$

D= a*b* denotación $\{a_n b_m \mid n,m >=0\}$

BNF (Backus - Naur Form)

Es un metalenguaje usado para expresar gramáticas libres de contexto: es decir, una manera formal de describir lenguajes formales.

Metasímbolos:

- < >: indica símbolo NO-TERMINAL o meta variable
- ::= : "Se define como"
- |: " o"
- $\{ \}_n$: Repetición. Mínimo **n** veces.
- identificador: Palabra reservada, constante o carácter TERMINAL.

BNF (Backus - Naur Form)

Ejemplo:

```
numero    ::= entPos
            | entPos '.' entPos

entPos    ::= digito
            | digito entPos

digito    ::= '0' | '1' | '2' | '3' | '4'
            | '5' | '6' | '7' | '8' | '9'
```

?	Opcional
*	0 a n veces
+	1 a n veces

BNF (Backus - Naur Form)

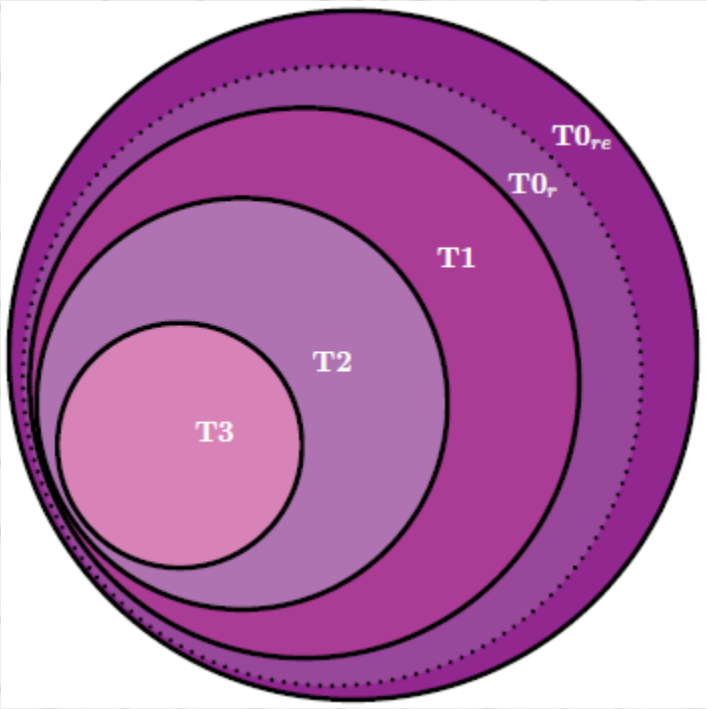
Ejemplo:

<frase>::= <sujeto><predicado>
<sujeto>::= juan | pedro | maría | salgado
<predicado>::= <verbo transitivo><objeto directo>
<predicado>::= <verbo intransitivo>
<verbo transitivo>::= ama | lava | peina | adora
<objeto directo>::= paula | antonio | sultán
<verbo intransitivo>::= corre | salta | camina

Complejidad estructural

A principios de los sesenta del siglo pasado Noam Chomsky define el espacio de lenguajes formales, la cual es conocido como Jerarquía de Chomsky

Complejidad estructural



T_3 = Lenguajes regulares.

T_2 = Lenguajes independientes del contexto.

T_1 = Lenguajes sensibles al contexto.

T_{0r} = Lenguajes recursivos.

$T0_{re}$ = Lenguajes recursivamente enumerables.

Complejidad estructural

Uno de los principales hallazgos de Chomsky fue la demostración de que podemos construir modelos matemáticos cuyas propiedades son un reflejo directo del grado de complejidad estructural de los lenguajes que se ajustan a dichos modelos.

Por tanto, en la medida que dos lenguajes pueden caracterizarse a través de un modelo matemático del mismo tipo, podremos decir que tienen la misma complejidad estructural y que pertenecen a la misma clase; y viceversa.

Complejidad estructural

Por tanto, en la medida que dos lenguajes pueden caracterizarse a través de un modelo matemático del mismo tipo, podremos decir que tienen la misma complejidad estructural y que pertenecen a la misma clase; y viceversa

Complejidad estructural

Los modelos matemáticos mediante los cuales podemos caracterizar lenguajes son dos: los autómatas y las gramáticas. Ambos, aunque parten de principios distintos, se puede demostrar que son equivalentes. Por tanto, si un lenguaje se puede caracterizar mediante un autómata propio de una determinada clase de complejidad, existe un procedimiento automático para derivar su caracterización mediante la gramática correspondiente, y viceversa.

Gramáticas

Una gramática formal es objeto o modelo matemático que permite especificar un lenguaje o lengua, es decir, es el conjunto de reglas capaces de generar todas las posibilidades combinatorias de ese lenguaje, ya sea éste un lenguaje formal o un lenguaje natural.

Gramática de libre contexto

Una gramática es un objeto $G = (\Sigma_T, \Sigma_N, S, P)$, donde

Σ_T es un alfabeto cuyos símbolos llamamos **símbolos terminales** de la gramática**.**

Σ_N es un alfabeto tal que $\Sigma_T \cap \Sigma_N = \emptyset$, cuyos símbolos llamamos **símbolos no terminales** de G.

S es un símbolo no terminal señalado al que llamamos **axioma** de la gramática.

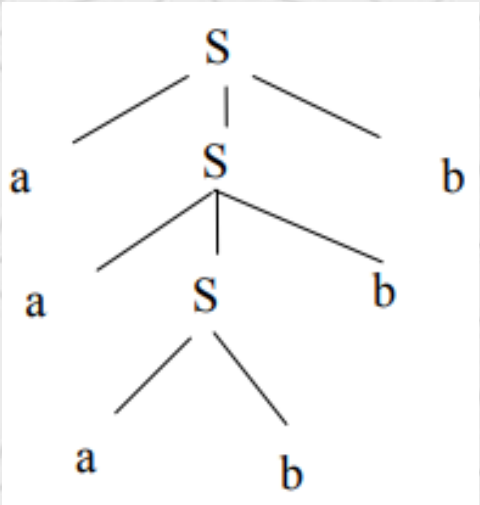
P es un conjunto de **reglas** definidas sobre el alfabeto $\Sigma = \Sigma_T \cup \Sigma_N$, tales que en la parte izquierda de todas las reglas aparece al menos un símbolo no terminal.

Gramática

Una gramática es un objeto $G = (\Sigma_T, \Sigma_N, S, P)$, donde

P: S→ ab|aSb

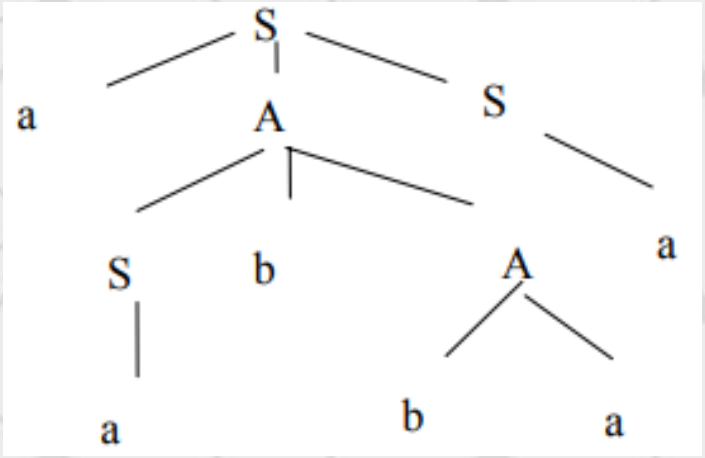
La derivación de la cadena aaabbb será: S ⇒ aSb ⇒ aaSbb ⇒ aaabbb y el árbol de derivación:



Derivación de la gramática

- Si leemos las etiquetas de las hojas de izquierda a derecha tenemos una sentencia. Llamamos a esta cadena la producción del árbol de derivación.
- Teorema. Sea $G = (\Sigma_T, \Sigma_N, S, P)$, una GLC. Entonces $S \Rightarrow^* \alpha$ (de S se deriva α) si y sólo si hay un árbol de derivación en la gramática G con la producción α.
- Si w es una cadena de L(G) para la gramática libre de contexto G, entonces w tiene al menos un árbol de derivación. Referido a un árbol de derivación particular, w tendrá una única derivación a la izquierda y otra única a la derecha.

Relación entre derivaciones y árboles



Ejemplo:

Derivación a la izquierda:

$S \Rightarrow aAS \Rightarrow aSbAS \Rightarrow aabAS \Rightarrow aabbaS \Rightarrow aabbaa$

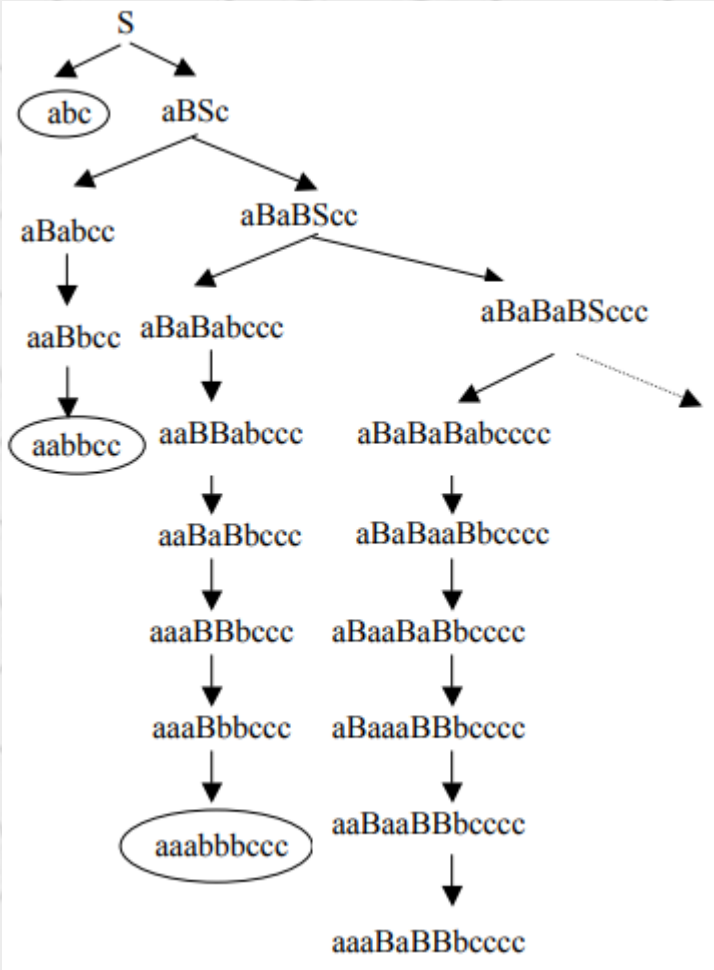
Derivación a la derecha:

$S \Rightarrow aAS \Rightarrow aAa \Rightarrow aSbAa \Rightarrow aSbbaa \Rightarrow aabbaa$

Ejemplo

Considere las reglas de la G.I.C. verifique si la palabra “aaabbbccc” está en la gramática.

S::= abc| aBSc
Ba::=aB
Bb::=bb



Ejercicio

Considere las reglas de la G.I.C. verifique si la palabra “aabbba” está en la gramática.

$$
\begin{aligned}
S &\rightarrow ASB \mid \varepsilon \\
A &\rightarrow aAb \mid \varepsilon \\
B &\rightarrow bBa \mid ba
\end{aligned}
$$

Fin