

**ALGUNAS PROPIEDADES MATEMÁTICAS DE  
LOS SISTEMAS LINGÜÍSTICOS**

**Fernando Galindo Soria  
Septiembre de 1992  
UPIICSA-IPN**

**Lic. En C.  
de la  
Informática**

**Sección de  
Posgrado e  
Investigación.**

**Calle de Te # 950  
Col. Granjas México  
México, D.F., 08400 MÉXICO  
Tel. 6-49-03-66 Ext. 301**

**I) DESARROLLO HISTÓRICO.**

**I.1 De la Lingüística Matemática a los Sistemas Dirigidos  
por Sintaxis.**

A pesar de que el área de la Lingüística Matemática es relativamente joven, ya que, empezó a consolidarse a mediados de los 50's, su campo de aplicación en la Informática ha ido creciendo rápidamente.

Uno de sus primeros logros se presentó cuando se usó para describir la gramática del Lenguaje Algol durante los años 60's propiciando que ya para finales de esa década empezaran a surgir libros donde se mostraba como construir un compilador a partir de la gramática de un lenguaje dado, y que durante los 70's se volviera cotidiana la construcción de compiladores e intérpretes bajo este enfoque.

Lo anterior ocasionó que en prácticamente todos los cursos de compiladores y de Programación de Sistemas se enseñaran una gran cantidad de métodos para analizar tanto léxica como sintácticamente las oraciones de algún lenguaje de programación

En particular en el caso del Análisis Sintáctico se desarrolló una gran variedad de métodos que permiten detectar si una oración es sintácticamente correcta, verificando si cumple o no con las reglas gramaticales del Lenguaje de Programación.

Conforme fue madurando el área, se detectó que existían muchos problemas en los cuales el usuario se comunicaba mediante algún lenguaje con la computadora, desde los problemas más complejos de reconocimiento de lenguaje natural hasta la comunicación en algún lenguaje de control

con el Sistema Operativo o algún lenguaje de Descripción de Datos con un Manejador de Base de Datos

En particular se detectó que algunos de estos problemas se volvían un caso específico de la construcción de intérpretes, ya que es relativamente fácil, usando las técnicas de compiladores que se reconozca por ejemplo las instrucciones de un lenguaje de control, y se ejecuten. A todo este universo de problemas se les agrupó con el nombre genérico de Sistemas Dirigidos por Sintaxis, y es así que en la actualidad es común encontrar por ejemplo editores dirigidos por sintaxis.

## **I.2) Inferencia Gramatical y Programación Dirigida por Sintaxis.**

Ya para mediados de los 70's era común tratar de construir interpretes de múltiples tipos de lenguajes, sin embargo, casi desde el principio se encontró un problema que luego se volvió cotidiano, ya que, según la técnica desarrollada en la construcción de compiladores, para poder construir el compilador o intérprete de un lenguaje dado, se requiere contar con su gramática y en ningún libro de Compiladores decía como encontrar la gramática de un lenguaje.

En el caso específico de los lenguajes de programación y algunos otros, la gramática la daba el diseñador del lenguaje y reflejaba las características y restricciones que se querían imponer al sistema, sin embargo, en muchos otros casos no se contaba con la gramática, sino con múltiples ejemplos de oraciones del lenguaje y con criterios y reglas empíricas, por lo que encontrar la gramática de un lenguaje dado a partir de un conjunto de oraciones se volvió un gran reto.

En paralelo con lo anterior y también desde mediados de los 60's se empezó a aplicar la Lingüística Matemática al Reconocimiento de Patrones y en particular al Reconocimiento de Imágenes y ya desde esa época se empezó a desarrollar el Reconocimiento Sintáctico de Patrones, en el cual se aplica la Lingüística Matemática para reconocer imágenes o patrones específicos viéndolos como 'oraciones' de algún 'Lenguaje de Patrones o Imágenes'.

Ahora bien, en el caso del Reconocimiento de Patrones es común contar con un gran número de oraciones que representar imágenes o patrones particulares, si embargo, por lo común no se cuenta con la gramática del lenguaje, por lo que, desde mediados de los 60's se comenzaron a desarrollar un conjunto de métodos y técnicas orientados a la obtención de la gramática de un lenguaje a partir de ejemplos de oraciones de este lenguaje. A todo este conjunto de

herramientas se les englobó con el nombre genérico de Inferencia Gramatical.

A principios de los 80's se empezaron a combinar la Inferencia Gramatical y la construcción de Compiladores con el fin de resolver problemas de tratamiento de diferentes tipos de lenguajes y gracias a esa interrelación ya se contaba con un conjunto de herramientas con las cuales:

a) a partir de un conjunto de ejemplos de un lenguaje se puede encontrar la gramática que describe el lenguaje.

b) a partir de la gramática se puede construir un compilador o intérprete capaz de reconocer las oraciones del lenguaje.

Ya para 1983 se tenían integradas estas herramientas en un método conocido como Programación Dirigida por Sintaxis, en el cual, se muestra un proceso para desarrollar sistemas a partir de ejemplos del lenguaje con el que se quieren dar órdenes al sistema.

### I.3) **Enfoque Lingüístico.**

En un principio estas herramientas se aplicaban para encontrar la gramática y construir el compilador de lenguajes muy concretos, tipo PASCAL, FORTRAN, lenguajes de consultas y por otro lado se siguieron aplicando al Reconocimiento de Patrones.

Sin embargo, conforme avanzó el área se detectó que existían muchos otros problemas donde se podía aplicar este método, únicamente con la condición de que los problemas a atacar fueran susceptibles de representarse mediante oraciones de algún lenguaje.

Pero por otro lado, el mismo concepto de lenguaje se fue ampliando, ya que, si en un principio se utilizaron estas herramientas para manejar lenguajes como FORTRAN y PASCAL, al mismo tiempo se utilizaban para representar imágenes y patrones en general.

*Actualmente el campo de aplicación de estas herramientas ha crecido enormemente y se postula que cualquier problema susceptible de ser atacado por medios automatizados es susceptible de representarse mediante oraciones de algún Lenguaje, llegándose a plantear así el Enfoque Lingüístico, en el cual se considera que cualquier 'objeto' se puede ver como una oración de algún lenguaje X.*

Ahora bien, si se tuviera que tener la lista de todas las oraciones de un lenguaje no terminaríamos, por lo que, comúnmente en lugar de la lista se utiliza una Gramática o conjunto de reglas que representan la estructura del lenguaje.

Por lo que, el principal problema cuando se tiene que manejar objetos de los cuales no se tiene la gramática es precisamente encontrar ésta.

#### **I.4) Operaciones Lingüísticas,**

En la actualidad ya existen una gran cantidad de herramientas de Inferencia Gramatical orientadas a encontrar la gramática de múltiples tipos de lenguaje (visuales, auditivos, de trayectorias, etc.).

Sin embargo, los primeros métodos presentados, tendían a ser complejos, particulares y difíciles de programar, por lo que, se empezaron a desarrollar nuevos métodos, para diferentes tipos de problemas.

A través de esta búsqueda de métodos y herramientas, ya para principios de los 80's se empezó a detectar que muchos métodos eran parecidos y solo eran un caso particular de métodos más generales.

En la actualidad se ha llegado a que existe un grupo de operaciones lingüísticas que al combinarse entre sí cubren la gran mayoría de los problemas de Inferencia Gramatical y por otro lado se ha detectado que estas operaciones son similares a ciertos procesos algebraicos y analíticos.

En particular se cuenta con una gran cantidad de herramientas basadas en las operaciones Lingüísticas de:

- .Factorización
- .Conmutatividad
- .Distribución
- .Recursividad

La Factorización y Distribución Lingüísticas son equivalentes a las algebraicas con la diferencia de que en este caso se factorizan o distribuyen componentes de una oración.

La Recursividad es, tal vez, la herramienta lingüística más poderosa ya que permite encontrar reglas generales o patrones a partir de casos particulares.

Estas herramientas de la Inferencia Gramatical se utilizan cotidianamente desde hace varios años, tanto para desarrollo de sistemas en forma manual mediante la

Programación Dirigida por Sintaxis, como en la construcción de Sistemas Evolutivos.

Es precisamente durante el desarrollo de estas aplicaciones que se ha detectado un conjunto de propiedades de tipo matemático, presentes en la Factorización, Distribución y Recursividad Lingüística.

## 2) FACTORIZACIÓN LINGÜÍSTICA.

### 2.1) Introducción.

Las herramientas de la Inferencia Gramatical trabajan con la estructura de las oraciones buscando encontrar una estructura general (o regla sintáctica) a partir de estructuras particulares (u oración canónica).

Así, si por ejemplo, se tienen las siguientes oraciones:

Juan es hermano de Pedro y  
Juan estudia en UPIICSA

se puede detectar que en las dos oraciones se encuentra presente la palabra Juan y que un párrafo equivalente sería:

Juan es hermano de Pedro y estudia en UPIICSA.

Si se observa lo que se ha hecho es detectar que la palabra Juan era común a las dos oraciones por lo que se factorizó (o sea que se sacó como factor común) y se obtuvo un párrafo donde sólo aparece una sola vez.

Para que se pueda visualizar el proceso sustituiremos fragmentos de la oración por etiquetas de acuerdo a la siguiente tabla:

<b>Fragmento</b>	<b>Etiqueta</b>
Juan	o1
es hermano de	r1
Pedro	o2
y	+
estudia en	r2
UPIICSA	o3

Con lo que el párrafo Juan es hermano de Pedro y  
Juan estudia en UPIICSA

quedarían como:

o1 r1 o2 +  
o1 r2 o3

Donde se observa que o1 es común a las 2 oraciones.

A las oraciones

o1 r1 o1 y  
o1 r2 o3

se les conoce como oraciones canónicas y en general cuando se sustituyen los elementos de una oración por una representación que permita visualizar la estructura de la oración se obtiene una oración canónica.

Recordemos que la factorización algebraica consiste en encontrar los factores comunes en una expresión algebraica y sacarlos de la expresión, como se ve a continuación:

$$. \underline{a}b + \underline{a}c = a(b+c)$$

$$. 3x + 3y = 3(x + y)$$

$$. a(b * c) + a(e / f) = a(b * c + e / f)$$

Aplicando lo anterior a las oraciones canónicas entonces tenemos que:

$$\underline{o1r1o2} + \underline{o1r2o3} = o1(r1o2+r2o3)$$

o sea que el párrafo

$$o1r1o2 + o1r2o3$$

es equivalente al párrafo

$$o1(r1o2+r2o3)$$

si sustituimos las etiquetas por los fragmentos que representan tenemos entonces que:

$$\underline{\text{Juan}} \underline{\text{es hermano de}} \underline{\text{Pedro}} \underline{\text{y}} \underline{\text{estudia en}} \underline{\text{UPIICSA}}$$

o1                      r1                      o2 +                      r2                      o3

## 2.2) Generación de Gramáticas.

La factorización Lingüística es una herramienta muy poderosa ya que permite encontrar los factores comunes dentro de un conjunto de oraciones, por lo que, si por ejemplo tengo un conjunto de ejemplos de algún lenguaje, aplicando la

factorización se pueden encontrar algunos de los factores comunes o reglas generales del lenguaje.

Lo anterior se puede aplicar para encontrar entonces una Gramática de un lenguaje a partir de ejemplos de las oraciones del lenguaje, ya que si por ejemplo, se tienen las siguientes oraciones canónicas (que se obtuvieron sustituyendo los elementos de las oraciones del lenguaje por etiquetas)

```
a b c d e
a b m e x y z
a b m g
```

lo primero que se hace es que como esas oraciones canónicas representan la estructura de las oraciones del lenguaje, entonces las integramos para formar una primera gramática del lenguaje

```
S --> a b c d e |
      a b m e x y z |
      a b m g
```

conocida como Gramática Canónica donde 'S -->' se puede ver como el nombre de una rutina y el símbolo '|' como un separador entre los diferentes casos de un programa.

Por lo que, la Gramática Canónica se puede leer como:

La Rutina S genera tres posibles tipos de oraciones:

```
Oraciones del tipo   abcde
                    del tipo   abmexyz
                    y del tipo   abmg
```

Lo cual es congruente con el hecho de que cada uno de los tipos anteriores corresponde a cada una de las oraciones canónicas que se dieron como ejemplo, por lo que podemos afirmar que la Gramática Canónica genera al menos las oraciones que se tenían como ejemplo originalmente.

Como siguiente paso se toman las oraciones de la gramática canónica y se les aplica la factorización lingüística, para lo que se considera al símbolo '|' equivalente al '+' del Álgebra tradicional

```
S --> abcde |
      abmexyz|
      abmg
```

factorizando ab queda:

```
S --> ab( cde |
          mexyz |
          mg )
```

Factorizando m queda:

```
s --> ab( cde |
          m( exyz|
            g ) )
```

Dado que no es común el uso de paréntesis dentro de una Gramática se introducen una serie de variables auxiliares (conocidas como Variables no Terminales) en lugar de los paréntesis, quedando

Introduciendo la variable no terminal X

```
S --> abX
X --> cde|
      m( exyz|
        g )
```

Introduciendo la variable no terminal Y

```
S --> abX
X --> cde|
      mY
Y --> exyz|
      g
```

Que se lee:

El programa S genera ab y llama a la rutina X

La rutina X genera las cadenas  
cde o  
mY

La rutina Y genera las cadenas  
exyz o  
g

si analizamos este programa podemos ver que:

```
S ==> abX ==> abcde|
          abmY==> abmexyz|
                  abmg
```

Donde '==>' significa 'se sustituye por' de donde S se sustituye por abX y así sucesivamente de donde llegamos que al final



S genera        abcde  
                 abmexyz  
                 abmg

que es la lista de oraciones originales  
De donde se tiene que la gramática canónica

S --> abcde  
         abmexyz  
         abmg

y la gramática

S --> abX  
X --> cde|  
         mY  
Y --> exyz|  
         g

obtenida a partir de la primera mediante la Factorización Lingüística generan las mismas oraciones.

Sin embargo, estructuralmente estas dos gramáticas no son iguales ya que en el primer caso se tiene sólo un nivel y en el segundo la gramática incluye tres niveles (S, X, Y).

Cuando dos gramáticas general el mismo lenguaje pero con diferente estructura se dice que son débilmente equivalentes

### **2.3) De la Gramática al Programa.**

Ahora bien si analizamos un poco a fondo las dos gramáticas y las vemos como dos programas podemos observar que la gramática canónica se comporta como un programa con tres opciones en la primera opción el programa ejecuta las instrucciones:

abc ...

en la segunda opción ejecuta  
abme ...

y en la tercera  
abmg

Por lo que existe una repetición de instrucciones comunes y lo más lógico es que el programa ejecute al principio

ab  
y después las opciones  
c...

me...  
mg...

que si observamos es precisamente la idea de la segunda gramática.

Por lo que la primera gramática se comporta como el programa:

```
Program S
  1) a b c d e
  2) a b m e x y z
  3) a b m g
```

fin programa

y la segunda gramática se comporta como:

```
Program S
ab
  1)cde
  2)m
    1)exyz
    2)g
```

fin programa

O sea que las instrucciones repetitivas sólo se ejecutan una sólo vez.

#### **2.4) Aplicaciones.**

Una de las aplicaciones de la factorización lingüística se encuentra precisamente en la depuración de programas con el fin de quitar código redundante.

Sin embargo no es la única ya que una gran cantidad de métodos de Inferencia Gramatical se reducen a la aplicación de la factorización.

Una aplicación que se desarrolló en 1988 en colaboración con Javier Ortiz (en esa época Coordinador de la Maestría en Computación del CENIDET en Cuernavaca, Mor.) consistió en la aplicación de la factorización a la construcción de un Sistema Evolutivo generador de Sistemas Expertos, en el cual, la idea consistió básicamente en tomar una gran cantidad de oraciones en las cuales un experto explica como resuelve un problema y transformar cada oración en una oración canónica incluyendo por ejemplo síntomas (s), diagnósticos (d) y tratamientos (t) e ignorando todo lo demás.

A partir de ahí integrar todas las oraciones canónicas en una Gramática Canónica, mediante factorización agrupar los síntomas comunes y proponerlos como reglas generales hasta construir una cascada de reglas de las más generales a las más específicas que caracterizan un problema o diagnóstico particular.

Por ejemplo si se tiene la oración:

Paciente femenino de 15 años con 38 grados de temperatura y  
 $\begin{matrix} S1 & S2 & S3 \end{matrix}$   
dolor en el pecho, se le diagnosticó faringitis y se le  
 $\begin{matrix} S4 & d1 \end{matrix}$   
recetó antibióticos, antihistamínicos y reposo  
 $\begin{matrix} t1 & t2 & t3 \end{matrix}$

la oración canónica equivalente sería:

S1S2S3S4d1dt1t2t3

Por otro lado, si en lugar de tener una sola oración se tiene la información de todos los pacientes del hospital entonces se pueden obtener cientos o miles de reglas canónicas, las cuales mediante factorización pueden proporcionar las reglas generales de un problema y su tratamiento.

Por ejemplo si se tienen las reglas.

S --- S1S2S3S4d1t1t2t3  
S1S2S5d2t1t4  
S1S4S6S7d3t5

Factorizando S1

S --> S1X  
X --> S2S3S4D1T1T2T3  
S2S5D2T1T4  
S4S6S7D3T5

Factorizando S2

S --> S1X  
X --> S2Y  
S4S6S7D3T5  
Y --> S3S4D1T1T2T3  
S5D2T1T4

Analizando la última gramática se observa que S1 es la característica general de los pacientes y después se tiene a los pacientes con S2 o con S4.

Obtener un sistema experto a partir de las reglas es directo.

### 3) CONMUTATIVIDAD LINGÜÍSTICA.

En el anterior ejemplo los síntomas, tratamientos y diagnósticos no necesariamente están ordenados o aparecen en el mismo orden en todas las oraciones, por lo que, ya que se encontraron las oraciones canónicas el siguiente paso consiste en ordenar los elementos de la oración, Por ejemplo, si se tiene:

S5S1d2t1S2t4

Al ordenarla queda:

S1S2S5d2t1t4

Lo anterior no necesariamente es aplicable a cualquier oración ya que es mucho más común que se trabaje con oraciones en las cuales no se permite la conmutatividad con lo que tenemos dos tipos de oraciones

Las oraciones conmutables como en el caso de los sistemas expertos y de las oraciones donde los diferentes elementos son del mismo tipo (por ejemplo una lista de atributos como: alto, fuerte, estudioso)

Y las oraciones no conmutables como en el caso general de cualquier expresión en Español. Por ejemplo:

El perro mordió al gato

no es lo mismo que:

mordió el al gato perro.

Por lo que la propiedad de la Conmutatividad Lingüística se puede aprovechar por ejemplo en los sistemas expertos pero no es de aplicación generalizada, ya que sólo se puede aplicar en general a elementos del mismo tipo y que se encuentren contiguos.

Por ejemplo la expresión algebraica

$a+b+c$

es conmutable y equivalente a

$b+a+c$

$a+c+b$  etc.

Por otro lado  $a+b*c$  es conmutable bajo +  
 $b*c+a$

o bajo \*

$a+c*b$

Pero no es conmutable en forma mezclada  
 $a*c+b$

Por otro lado la expresión  $a-b/c$   
no es conmutable bajo ningún caso ya que se obtendrían  
cosas como:

$b/c-a$

$a/b-c$

En el caso de la expresión lingüística se presenta exactamente el mismo problema con la diferencia de que en este caso la cantidad de tipos de elementos y de operadores es mucho mayor y puede cambiar de un lenguaje a otro por lo que requiere de un análisis específico para cada caso.

#### 4) DISTRIBUCIÓN LINGÜÍSTICA.

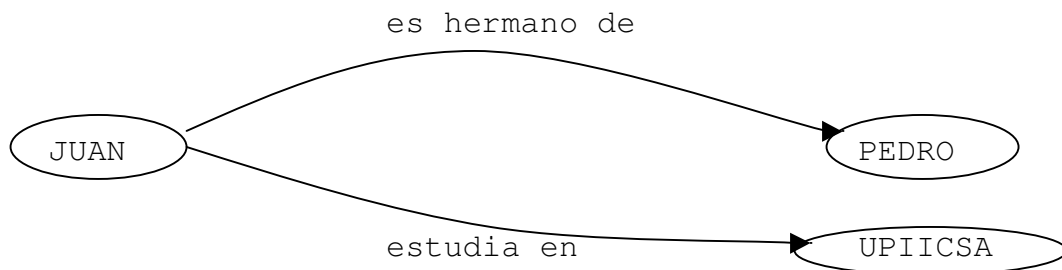
##### 4.1) Antecedentes.

En 1990 durante la construcción de un Sistema Evolutivo para representación de conocimiento desarrollado por Jesús Olivares, se presentó un problema relacionado con la generación de redes semánticas a partir de oraciones declarativas en lenguaje natural, ya que cuando las oraciones eran por ejemplo:

a) Juan es hermano de Pedro

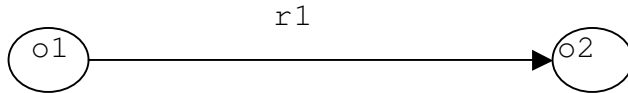
b) Juan estudia en UPIICSA

es relativamente fácil construir la red semántica



Ya que en general los elementos de una red semántica son objetos relacionados entre sí, por lo que si tengo una oración de la forma o1 r1 o2

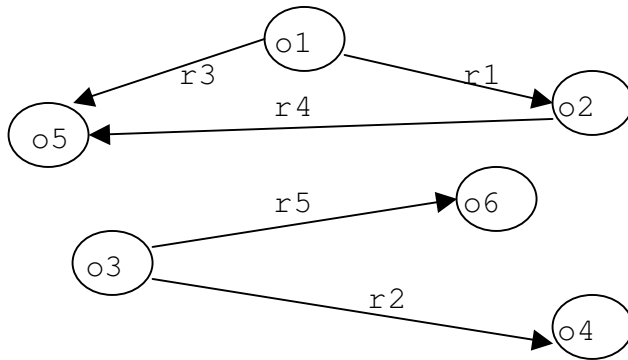
la red semántica es directa



En general si se tiene un conjunto de oraciones

o1r1o2  
o3r2o4  
o1r3o5  
o2r4o5  
o3r5o6

la red semántica es directa ya que simplemente cada objeto se asocia con un nodo (bolita) y cada relación con un flecha, quedando la red:



Sin embargo, cuando se recibe una oración declarativa en lenguaje natural, difícilmente es una lista del tipo objeto relación objeto ((oro) sino que es más común oraciones como:

a) Juan es hermano de Pedro y estudia en UPIICSA

o sea de la forma

$o1(r1o2+r2o3)$

b) Juan es inteligente, estudioso, trabajador y alegre.

o sea de la forma

$o1r1(a1+a2+a3+a4)$

donde  $a1, \dots, a4$  son los atributos de Juan.

## 4.2) Distribución Lingüística.

Ahora bien, a partir de oraciones como las anteriores no es factible general directamente la red semántica, por lo que, es necesario transformarlas en oraciones del tipo ORO.

Para lograr lo anterior se desarrolló y aplicó una técnica conocida como Distribución Lingüística.

La distribución lingüística es la operación inversa de la factorización lingüística, por lo que:

$$ol(r1o2+r2o3) = olr1o2+olr2o3$$

$$olr1(a1+a2+a3+a4) = olr1a1+olr1a2+olr1a3+olr1a4$$

De donde, si se tiene la oración:

Juan es inteligente, estudioso, trabajador y alegre

Su oración canónica es:

$$olr1(a1+a2+a3+a4)$$

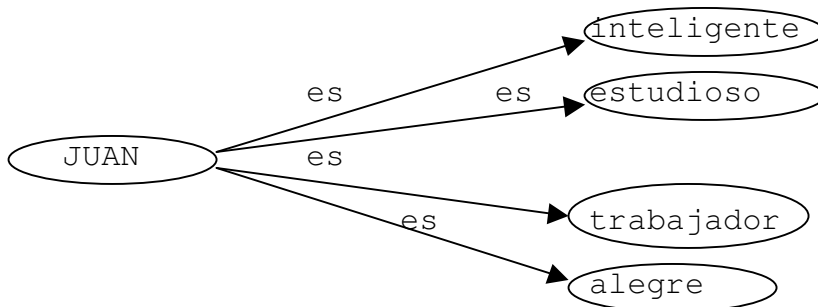
al aplicar la distribución lingüística, queda:

olr1a1+  
olr1a2+  
olr1a3+  
olr1o4

de donde se tienen las oraciones:

Juan es inteligente,  
Juan es estudioso,  
Juan es trabajador,  
Juan es alegre

y a partir de ahí la red semántica es directa



La distribución lingüística se puede aplicar a la descomposición de oraciones relativamente complejas, por ejemplo:

$$\frac{\text{Juan}}{o1} + \frac{\text{Pedro}}{o2} \frac{\text{estudian en}}{r1} \frac{\text{UPIICSA}}{o3} + \frac{\text{trabajan en el}}{r2} \frac{\text{Metro}}{o4}$$

se puede ver como

$$\begin{aligned}
 &(o1+o2) (r1o3+r2o4) = \\
 &=o1(r1o3+r2o4)+o2(r1o3+r2o4) = \\
 &= o1r1o3 + o1r2o4 + \\
 &\quad o2r1o3 + o2r2o4
 \end{aligned}$$

Que son equivalentes a las oraciones:

Juan estudia en UPIICSA y  
 Juan trabaja en el metro y  
 Pedro estudia en UPIICSA y  
 Pedro trabaja en el Metro.

De donde la red semántica sale directa.

## 5) PROPIEDADES ALGEBRAICAS DE LOS SISTEMAS LINGÜÍSTICOS.

Las operaciones de factorización, conmutatividad y distribución Lingüística son operaciones netamente algebraicas, por lo que, en este apartado se comentarán algunas características generales de este sistema algebraico.

En primer lugar, es necesario comentar que dentro de una Gramática se encuentran involucrados al menos dos operadores.

El primer operador es el operador de Concatenación que permite construir cadenas de caracteres (u oraciones) a partir de elementos simples.

Por ejemplo si se tiene el siguiente conjunto de elementos

$$V_T = ( a, b, c, d )$$

La operación de concatenación permite formar cadenas como:

abc  
 baba  
 bcdabc  
 ac



La operación de concatenación se podría ver como el '\*' de algunos otros sistemas algebraicos.

El segundo operador presente en una gramática es el operador '|' que se lee 'o' y se presenta cuando en una gramática se tiene más de una opción.

Por ejemplo la gramática

```
S --- abcd |
      abd |
      bca
```

Nos dice que S se puede sustituir por:

abcd ó abd ó bca

El operador '|' se puede ver como el '+' de algún otro sistema algebraico, de donde por ejemplo la factorización lingüística se puede visualizar como:

abcb+abd+bca=ab(cb+d) bca.

La concatenación y el operador '|' no son los únicos elementos algebraicos presentes en un sistema lingüístico ya que también se cuenta con un elemento que funciona como neutro multiplicador conocido como cadena vacía  $\epsilon$ .

La cadena vacía es una cadena de caracteres sin caracteres (se dice que la cardinalidad de  $\epsilon$  o  $|\epsilon|=0$ ) y tiene las siguientes propiedades:

Dada una cadena de caracteres  $m$ ,  $\epsilon \cdot m = m = m \cdot \epsilon$  o sea que se comporta igual que la unidad bajo la multiplicación.

Si integramos los operadores de concatenación, '|' y la cadena vacía  $\epsilon$ , entonces resulta que la Gramática se comporta como un Anillo Algebraico y como tal es susceptible de ser estudiado desde el punto de vista de la Teoría algebraica.-

## 6) RECURSIVIDAD LINGÜÍSTICA.

### 6.1) Introducción a la Recursividad.

La Recursividad es tan importante que ameritaría un trabajo por si sola, por lo que en este documento solo presentaremos una breve introducción a su interrelación con la Lingüística Matemática y el Análisis Matemático con el fin de visualizar su fuerza.

En general se considera que un sistema es recursivo cuando se llama a si mismo, por ejemplo la rutina que grafica árboles:

```
Rutina Arbol (xo, yo, t, a)
    x1 = xo + t * cos (a)
    y1 = yo + t * sen (a)
    dibuja tronco (xo, yo, x1, y1)
    Arbol (x1, y1, t/2, a - 30)
    Arbol (x1, y1, t/2, a + 30)
fin rutina
```

es una rutina recursiva porque se llama a si misma.

### 6.2) Lingüística Matemática y Recursividad.

Dentro de la Lingüística Matemática se han encontrado una gran cantidad de casos en los que la mejor representación de un lenguaje es en términos de una gramática recursiva y aun mas, en algún momento se ha postulado que el mecanismo "natural" de adquisición del lenguaje en los seres vivos es un mecanismos que genera estructuras recursivas, ya que por ejemplo, si se tiene el siguiente conjunto de oraciones:

Estudia y triunfaras  
a                    b

Estudia y trabaja y triunfaras  
a                    a                    b

Estudia y trabaja y ahorra y triunfaras  
a                    a                    a                    b

Al representarlas mediante su gramática canónica

ab  
aab  
aaab

Se observa que entramos en un ciclo de repetición en el cual se pueden poner tantos elementos como se quiera, y podemos proponer que una oración del tipo

aaa...ab

es sintácticamente válida dentro del lenguaje.

Cuando se tiene un conjunto de oraciones donde un tipo de elemento se puede repetir en forma indefinida es conveniente sustituir todo el conjunto de oraciones por una regla recursiva, ya que la representación es mucho más compacta y general.

Por ejemplo si tiene

S--> ab  
aab  
aaab  
...  
aaa...ab

Una regla recursiva asociada es:

S--> aS | b

Ya que sustituyendo S por aS o por b se pueden tener cadenas como las siguientes:

S==> aS==> aaS==> aab  
S==> aS==> aaS==> aaaS==> aaab

y en general cualquier número de a seguidos por b.

Entonces la gramática recursiva genera todas las cadenas originales y muchas mas.

### **6.3) Gramática Recursiva y Generalización.**

Por lo común la gramática recursiva es una generalización de la gramática canónica, es decir, que si tengo un conjunto de oraciones donde se detecta una estructura recursiva, ésta no solo genera todo el conjunto de oraciones, sino que además, es capaz de generar muchas otras que no estaban contempladas.

Esta propiedad de generalización de las gramáticas recursivas las hace extremadamente poderosas ya que permite encontrar a partir de unos cuantos ejemplos la estructura de un lenguaje.

Sin embargo al generalizar se puede llegar a proponer una gramática que genere estructuras oracionales que no sean

válidas en el lenguaje, o sea que la gramática puede ser tan general que produzca cosas sin sentido o contradictorias.

A pesar de lo anterior la fuerza de la Recursividad es tan enorme que se utiliza cotidianamente para atacar una gran cantidad de problemas de Lingüística y únicamente se debe ser conciente de sus peligros y no usarla a ciegas.

#### 6.4) Método de Generación de Gramáticas Recursivas.

En muchos casos la recursividad se ha introducido en forma intuitiva a los sistemas, sin embargo, ya existen métodos que permiten obtener una gramática recursiva a partir de un conjunto de oraciones del lenguaje.

Por ejemplo si se tiene la gramática:

S--> ab  
aab  
aaab

la gramática recursiva que la generaliza es:

S--> aS | b

*Ahora bien, si se observa el ejemplo se puede notar que la recursividad se introduce cuando se detecta que una cadena a se repite en forma monótona aaa alrededor o tendiendo a un punto b.*

Por ejemplo en la oración

abcabcabcabcd

la cadena abc presenta un comportamiento monótono que termina en d.

Dada una oración repetitiva para generar la gramática recursiva se siguen los siguientes pasos:

- a) Se detecta un comportamiento monótono.
- b) Se busca el elemento repetitivo.
- c) Se detecta a que punto tiende
- d) Se genera la gramática recursiva

Por ejemplo, en la cadena:

38383838386

El elemento repetitivo es 38 el sistema tiende a 6 la gramática es:

S--> 38 S | 6

o sea que toda la cadena repetitiva se sustituye por el elemento repetitivo 38 seguido de un llamado recursivo S, por otro lado el núcleo del sistema 6 se pone como otra opción de la gramática.

Por ejemplo la cadena:

mnopmnopmnopq

tiene como cadena repetitiva a mnop y tiende a q

Por lo que la gramática queda:

S--> mnop S|q

Un caso interesante se presenta cuando el comportamiento monótono es alrededor de un punto, como por ejemplo en la oración:

((((a))))

donde el número de paréntesis izquierdos es el mismo que derechos y giran alrededor de a

La letra a funciona como Núcleo del proceso recursivo.

Para generar la producción recursiva se sustituye el núcleo por la variable recursiva, de donde la gramática queda:

S--> (S)|a

## 7) ENFOQUE ANALÍTICO DE LOS SISTEMAS LINGÜÍSTICOS.

Como se podrá observar según este enfoque el concepto de recursividad es similar al concepto de límite en el Análisis Matemático ya que en los dos casos se tiene un conjunto de elementos que tienden a o giran alrededor de un núcleo o límite.

Las similitudes no se quedan en ese punto, ya que, el proceso recursivo es un proceso que se puede continuar indefinidamente y requiere de un atractor o límite o núcleo o criterio de terminación para detenerse, de donde, tal vez, el Límite del Análisis Matemático y la Recursividad son dos componentes de un fenómeno mas general.

Aparentemente lo anterior es muy factible, ya que se tienen ejemplos de procesos recursivos que "tienden a un límite" como es el caso de algunos fractales que en el límite tienden a ocupar el espacio delimitado por otros fractales.

## C O N C L U S I Ó N

En este documento se presentó una breve introducción a la Lingüística Matemática y a sus propiedades Matemáticas. En particular se presentaron las operaciones de Factorización, Conmutatividad y Distribución Lingüística y se vio su contraparte algebraica, llegándose a proponer la construcción de un Álgebra Lingüística soportada por las operaciones de concatenación y "|" y por el neutro multiplicativo o cadena vacía , por lo que, específicamente se planteó que estamos ante la presencia de un Anillo Algebraico, y es factible ver al Álgebra Lingüística, como un nuevo campo de acción y en particular empezar a aplicarle todos los resultados encontrados durante el desarrollo de la Escuela Algebraica.

Por otro lado, se dio una introducción a la operación de Recursividad Lingüística y se vio su similitud con el concepto de Límite del Análisis Matemático, por lo que, nuevamente se presentó una interrelación entre dos campos aparentemente disimbolos, y se plantea que de esta interrelación se puede enriquecer tanto el Análisis como la Lingüística.

Las operaciones algebraicas y analíticas de la Lingüística Matemática no están separadas ya que en la mayoría de los problemas de Inferencia Gramatical se aplican conjuntamente, por lo que, otro campo de estudio se encuentra en la interrelación del Álgebra y el Análisis en el estudio de problemas que involucran por ejemplo la Factorización y la Recursividad Lingüística.

Finalmente no quiero perder la oportunidad de mencionar que he oído comentarios acerca de que los pueblos prehispánicos manejaban los conceptos de Límite y Recursividad dentro de un mismo campo, por lo que, tal vez, estemos en el umbral de una puerta que ha permanecido cerrada por 500 años.