

III.5 ALGUNAS PROPIEDADES MATEMÁTICAS DE LOS SISTEMAS LINGÜÍSTICOS

*Fernando Galindo Soria**

I DESARROLLO HISTÓRICO

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.

* Fernando Galindo Soria escribió este trabajo en Septiembre de 1992 cuando estaba en la Sección de Postgrado e Investigación de la UPIICSA del IPN. Lo presentó como Conferencia Magistral y publicó en las memorias del Simposio Nacional de Computación, organizado por el CENAC del IPN en la Cd. de México en Noviembre de 1992. Igualmente se publicó en las memorias sobre "Sistemas Evolutivos" del 1er Congreso Internacional de Investigación en Ciencias Computacionales, organizado por el Instituto Tecnológico de Toluca en Metepec Edo. de México, en Septiembre de 1994.

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.

2 INFERENCIA GRAMATICAL Y PROGRAMACIÓN DIRIGIDA POR SINTAXIS

Ya para mediados de los 70's era común tratar de construir intérpretes 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 empleada 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 representan Imágenes o patrones particulares, sin 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 los problemas de tratamiento de diferentes tipos de lenguajes y gracias a esa interrelación ya se contaba con un conjunto de herramientas con las cuales:

- 1) *A partir de un conjunto de ejemplos* de un lenguaje se puede encontrar la gramática que describe el lenguaje.
- 2) *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.

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 cosa 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.

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:

- 1) *Factorización.*
- 2) *Conmutatividad.*
- 3) *Distribución.*
- 4) *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.

II FACTORIZACIÓN LINGÜÍSTICA

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 oraciones canónicas).

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:

Fragmento	Etiqueta
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 dos 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 factores comunes en una expresión algebraica y sacarlos de la expresión, como se ve a continuación:

$$a\ b + a\ c = a(b + c)$$

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

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

Aplicando lo anterior a las oraciones canónicas entonces tenemos que:

$$o1\ r1\ o2 + o1\ r2\ o3 = o1(r1\ o2 + r2\ o3)$$

o sea que el párrafo $o1\ r1\ o2 + o1\ r2\ o3$ es equivalente al párrafo $o1(r1\ o2 + r2\ o3)$ si sustituimos las etiquetas por los fragmentos que representan tenemos entonces que:

$$\begin{array}{ccccccc} \text{Juan} & \text{es} & \text{hermano de} & \text{Pedro} & \text{y} & \text{estudia en} & \text{UPIICSA} \\ o1 & & r1 & & o2 & + & r2 & & o3 \end{array}$$

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:

$$S \rightarrow a b c d e \mid$$

$$a b m e x y z \mid$$

$$a b m g$$

conocida como Gramática Canónica donde “ $S \rightarrow$ ” se puede ver como el nombre de una rutina y el símbolo “ $\mid$ ” como un separador entre los 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 a b c d e
del tipo a b m e x y z
o del tipo a b m g

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 “ $\mid$ ” equivalente al “ $+$ ” del álgebra tradicional.

$$S \rightarrow \underline{a b} c d e \mid$$

$$\underline{a b} m e x y z \mid$$

$$\underline{a b} m g$$

factorizando a b queda:

$$S \rightarrow \underline{a\ b} (c\ d\ e\ |\ m\ e\ x\ y\ z\ |\ m\ g)$$

factorizando m queda:

$$S \rightarrow a\ b\ (c\ d\ e\ |\ \underline{m}\ (e\ x\ y\ z\ |\ g))$$

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, por ejemplo:

Introduciendo la variable no terminal X

$$\begin{aligned} S &\rightarrow a\ b\ X \\ X &\rightarrow c\ d\ e\ |\ m\ (e\ x\ y\ z\ |\ g) \end{aligned}$$

Introduciendo la variable no terminal Y

$$\begin{aligned} S &\rightarrow a\ b\ X \\ X &\rightarrow c\ d\ e\ |\ m\ Y \\ Y &\rightarrow e\ x\ y\ z\ |\ g \end{aligned}$$

que se lee:

El programa S genera a b y llama a la rutina X

La rutina X genera la cadena:

c d e

o la cadena m, y llama a la rutina Y.

La rutina Y genera la cadena:

e x y z

o la cadena:

g
si analizamos este programa podemos ver que:

$$S \Rightarrow a b X \Rightarrow a b m Y \Rightarrow a b m e x y z$$

Donde “ $\Rightarrow$ ” significa “se sustituye por” de donde S se sustituye por a b X y así sucesivamente de donde llegamos que al final:

$$\begin{aligned} S \text{ genera } a b c d e, \\ a b m e x y z \\ \text{ó } a b m g \end{aligned}$$

que es la lista de oraciones originales

De donde se tiene que la gramática canónica

$$\begin{aligned} S &\rightarrow a b c d e \mid \\ &a b m e x y z \mid \\ &a b m g \end{aligned}$$

y la gramática

$$\begin{aligned} S &\rightarrow a b X \\ X &\rightarrow c d e \mid \\ &m Y \\ Y &\rightarrow e x y z \mid \\ &g \end{aligned}$$

obtenida a partir de la primera mediante la factorización lingüística generan las mismas oraciones.

Sin embargo, estructuralmente esas 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.

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

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.

a b c ...

en la segunda opción ejecuta

a b m e ...

y en la tercera

a b m g

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

a b

y después las opciones

c...

m e...

m g...

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

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

Programa S

1) a b c d e

2) a b m e x y z

3) a b m g

fin del programa

y la segunda se comporta como:

Programa S

a b

1) c d e

2) m

1) e x y z

2) g

fin del programa

O sea que las instrucciones repetitivas sólo se ejecutan una sola

vez.

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 con la colaboración de Javier Ortiz (en esa época coordinador de la Maestría en Computación del CENIDET en Cuernavaca, Morelos) 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 lo demás.

A partir de ahí integrar todas las oraciones canónicas en una Gramática Canónica, mediante factorización agrupar los síntomas comunes y ponerlos 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

s1 s2 s3

s1 s2 s3

dolor en el pecho, se le diagnóstico faringitis y se le

s4 d1

recetó <u>antibióticos</u> , <u>antihistaminicos</u> y <u>reposo</u>		
t1	t2	t3

t1		t2		t3
----	--	----	--	----

la oración canónica equivalente sería:

s1 s2 s3 s4 d1 t1 t2 t3

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:

$$\begin{array}{l} S \rightarrow s1 \ s2 \ s3 \ s4 \ d1 \ t1 \ t2 \ t3 \ | \\ \quad s1 \ s2 \ s5 \ d2 \ t1 \ t4 \ | \\ \quad s1 \ s4 \ s6 \ s7 \ d3 \ t5 \end{array}$$

factorizando s1

$$\begin{array}{l} S \rightarrow s1 \ X \\ X \rightarrow s2 \ s3 \ s4 \ d1 \ t1 \ t2 \ t3 \ | \\ \quad s2 \ s5 \ d2 \ t1 \ t4 \ | \\ \quad s4 \ s6 \ s7 \ d3 \ t5 \end{array}$$

factorizando s2

$$\begin{array}{l} S \rightarrow s1 \ X \\ X \rightarrow s2 \ Y \ | \\ \quad s4 \ s6 \ s7 \ d3 \ t5 \\ Y \rightarrow s3 \ s4 \ d1 \ t1 \ t2 \ t3 \ | \\ \quad s5 \ d2 \ t1 \ t4 \end{array}$$

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.

III CONMUTATIVIDAD LINGÜÍSTICA

En el 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:

$$s5 \ s1 \ d2 \ s2 \ t4$$

al ordenarla queda:

s1 s2 s5 d2 t1 t4

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 gato al 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:

$$b / c - a$$

$$a / b - c$$

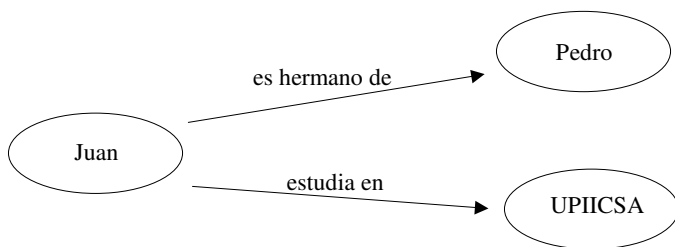
En el caso de la expresión lingüística se presenta exactamente el mismo problema con la diferencia de que en este caso la cantidad de tipo de elementos y de operadores es mucho mayor y puede cambiar de un lenguaje a otro por lo que requiere de un análisis específico para cada caso.

IV DISTRIBUCIÓN LINGÜÍSTICA

1 ANTECEDENTES

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

Juan es hermano de Pedro, Juan estudia en UPIICSA
es relativamente fácil construir la red semántica:



Pero cuando las oraciones son como en:

Juan es inteligente, estudioso, trabajador y alegre
o sea de la forma

$$o1 r1 (a1 + a2 + a3 + a4)$$

donde $a_1, \dots, a_4$ son los atributos de Juan, no es factible generar directamente la red semántica, por lo que, es necesario transformarlas en oraciones del tipo o r o

2 DISTRIBUCIÓN LINGÜÍSTICA

Para lograr lo anterior se desarrolló y aplicó una técnica conocida como Distribución Lingüística, que es la operación inversa de la factorización lingüística, por lo que:

$$o_1(r_1 o_2 + r_2 o_3) = o_1 r_1 o_2 + o_1 r_2 o_3$$

De donde, si se tiene la oración:

Juan es inteligente, estudioso, trabajador y alegre

su oración canónica es:

$$o_1 r_1(a_1 + a_2 + a_3 + a_4)$$

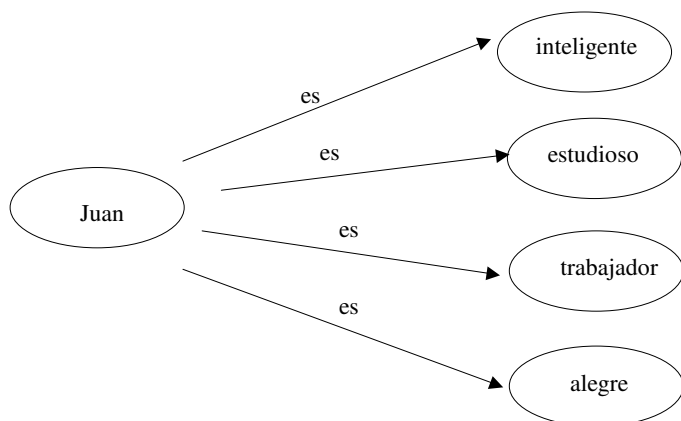
al aplicar la distribución lingüística queda:

$$o_1 r_1 a_1 + o_1 r_1 a_2 + o_1 r_1 a_3 + o_1 r_1 a_4$$

de donde se tienen las oraciones:

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

y de estas oraciones obtenidas la red semántica que se genera es:



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

$$\text{Juan y Pedro } \underline{\text{estudian en UPIICSA}} \text{ y } \underline{\text{trabajan en el Metro}}$$

$$o1 + o2 \quad r1 \quad o3 + r2 \quad o4$$

cuya oración canónica es:

$$(o1 + o2)(r1 o3 + r2 o4)$$

aplicando la distribución se obtiene:

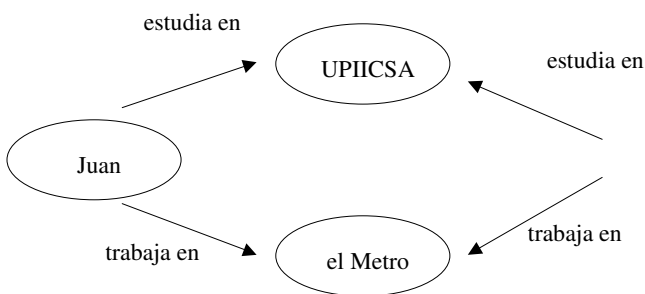
$$o1(r1 o3 + r2 o4) + o2 (r1 o3 + r2 o4)$$

$$o1 r1 o3 + o1 r2 o4 + o2 r1 o3 + o2 r2 o4$$

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 estas oraciones obtenidas se puede obtener directamente su red semántica.



V PROPIEDADES ALGEBRAICAS DE LOS SISTEMAS LINGÜÍSTICOS

Las operaciones de Factorización, Conmutatividad y Distribución Lingüísticas son operaciones netamente algebraicas, por lo que, en este apartado se comentan 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:

a b c
b a b a
b c d a b c
a c

La operación de concatenación se puede ver como la multiplicación en algunos otros sistemas algebraicos.

El segundo operador presente en una gramática es el que se lee “ó” y se presenta cuando en una gramática se tiene más de una opción.

Por ejemplo la gramática

$$\begin{aligned} S \rightarrow a b c d \mid \\ a b d \mid \\ b c a \end{aligned}$$

Nos dice que S se puede sustituir por:

$$a b c d \text{ ó } a b d \text{ ó } b c a$$

El operador “|” se puede ver como el operador de suma de algún otro sistema algebraico.

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 cadena vacía o λ .

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

Dada una cadena de caracteres m:

$$m \lambda = \lambda m = m$$

o sea que λ se comporta como la unidad bajo la multiplicación.

Si integramos los operadores de concatenación, “|” y la cadena vacía entonces la gramática se comporta como un anillo algebraico y es susceptible de ser estudiado desde el punto de vista de la teoría algebraica.

VI RECURSIVIDAD LINGÜÍSTICA

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 gráfica arboles:

```
Rutina Arbol (x0, y0, t, a)
  x1 = x0 + t * cos (a)
  y1 = y0 + t * sen (a)
  dibuja tronco (x0,y0, 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

2 LINGÜÍSTICA MATEMÁTICA Y RECURSIVIDAD

Dentro de la lingüística matemática se han encontrado una gran variedad 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 mecanismo que genera estructuras recursivas, ya que por ejemplo, si se tiene el siguiente conjunto de oraciones:

Estudia y triunfaras
 a b
 Estudia, trabaja y triunfaras
 a a b
 Estudia, trabaja, ahorra y triunfaras
 a a a b

Al representarlas mediante su gramática canónica se obtiene:

a b
 a a b
 a a a b

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

a a a ... a b

es sintácticamente válida dentro del lenguaje.

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

Por ejemplo si se tiene:

$S \rightarrow a b \mid$
 $a a b \mid$
 $a a a b$
 $\dots \mid$
 $a a a \dots a b$

Una gramática recursiva asociada es:

$$S \rightarrow a S \mid b$$

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

$$S \Rightarrow a S \Rightarrow a a S \Rightarrow a a b$$

$$S \Rightarrow a S \Rightarrow a a S \Rightarrow a a a S \Rightarrow a a a b$$

y en general cualquier número de a seguidos por una b .

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

3 GRAMÁTICA RECURSIVA Y GENERALIZACIÓN

Por lo común la gramática recursiva es una generalización de las oraciones que representa, es decir, que si se tiene un conjunto de oraciones donde se detecta una estructura recursiva, ésta no solo genera el conjunto de oraciones sino 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 también estructuras 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 consciente de sus peligros y no usarla a ciegas.

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:

$$\begin{aligned} S &\rightarrow b \mid \\ &\quad a b \mid \\ &\quad a a b \mid \\ &\quad a a a b \end{aligned}$$

la gramática recursiva que la generaliza es:

$$S \rightarrow a S \mid b$$

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 a a a alrededor o tendiendo a un punto b. El punto b, al cual tiende la cadena se le conoce también como límite o núcleo (obsérvese que este método incluye la cadena b o sea donde a se repite cero veces).*

Por ejemplo en la oración

$$a b c a b c a b c a b c d$$

la cadena a b c presenta un comportamiento monótono que termina en d.

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

- 1) *Se detecta un comportamiento monótono.*
- 2) *Se busca el elemento repetitivo.*
- 3) *Se detecta a que punto tiende.*
- 4) *Se genera la gramática recursiva.*

Por ejemplo, en la cadena:

$$3 \ 8 \ 3 \ 8 \ 3 \ 8 \ 3 \ 8 \ 3 \ 8 \ m$$

El elemento repetitivo es 3 8 el sistema tiende a m, por lo que la gramática es:

$$S \rightarrow 38S \mid m$$

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 m se pone como otra opción de la gramática.

Por ejemplo la cadena:

$$m n o p m n o p m n o p q$$

tiene como cadena repetitiva a m n o p y tiende a q

Por lo que la gramática queda como:

$$S \rightarrow m n o p S \mid q$$

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:

$$S \rightarrow (S) \mid a$$

5 ENFOQUE ANALÍTICO DE LOS SISTEMAS LINGÜÍSTICOS

Como se puede 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, límite, 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 limite” como es el caso de algunos fractales que en el limite tienden a ocupar el espacio delimitado por otros fractales.

CONCLUSIÓN

En este trabajo se presento 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 Algebra Lingüística soportada por las operaciones de concatenación, por “|” y por el neutro multiplicativo o cadena vacía λ , por lo que, específicamente se planteo que estamos ante la presencia de un Anillo Algebraico, y es factible ver al Algebra Lingüística, como un nuevo campo de acción y en particular empezar a aplicarle 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 Limite del Análisis Matemático, por lo que, nuevamente se presento 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 escuchado comentarios acerca de que los pueblos prehispánicos manejaban los conceptos de Limite 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.

