Tesis #411

T 130

UNIVERSIDAD DE SONORA

BIBLIOTECA
DE CIENCIAS EXACTAS
Y NATURALES

SABER DE MIS HIJOS
MI GRANDEZA

"CALCULO DE TIPOS SINTACTICOS"

TESIS QUE PARA OBTENER EL TITULO DE:

LICENCIADO EN MATEMATICAS

P R E S E N T A

FRANCISCO CANDIDO GARCIA DURAN

HERMOSILLO, SONORA, ENERO 1985.

INTRODUCCION.

Un poco de reflexión hace claro que en la oración

yo trabajo duro en mi trabajo

aparece dos veces la palabra *trabajo* funcionando la primera vez como verbo y la segunda como sustantivo. Así pues, las palabras, o aún una misma palabra, cumplen distintas funciones dentro de una oaración. Esta "función" puede ser señalada, como hacemos en el cap. I, asignando a cada palabra un "tipo sintáctico" de tal manera que al conocer su tipo podemos saber qué papel juega dentro de una oración. Cuestiones como la anterior son temas de interés y estudio para la Linguística, entendiéndose esta como "el estudio científico del lenguaje" [20]. En particular, corresponde a la Sintaxis el estudio de dichas cuestiones de acuerdo a la clasificación bosquejada a continuación.

SEMANTICA, SINTAXIS Y FONOLOGIA.

De manera general y superficial podemos distinguir con facilidad en un hecho linguístico 3 aspectos. Por ejemplo, si una persona pronuncia la oración en español

(a) *Juan Pérez estudia Algebra Booleana*"

resulta claro que es "entendible", es decir, para el oyente de una u otra manera la oración significa algo: tiene "significado". Pero ¿qué entendemos por "significado"?, ¿cómo adquiere una palabra, o un grupo de palabras, un "significado"?, ¿porqué el grupo de palabras "*nozam eslotia achytle*" no tiene significado en español? Un segundo aspecto es el "sonoro": al pronunciar la oración emitimos una serie de sonidos. Ahora bien ¿cómo pronunciamos (emitimos) estos sonidos?, ¿porqué emitimos determinados sonidos y no otros?; además podemos escribir por medio de símbolos (letras) esta oración en español (cómo efectivamente lo hicimos) y cualquier otra, así pues, ¿qué relación hay entre un sonido y el símbolo usado para representarlo?

Un tercer aspecto es el del "orden", es decir, las palabras son emitidas y escritas con un cierto orden, y manteniendo "ciertas relaciones" entre ellas. Si cambiamos el orden alteramos la oración en mayor o menor grado, por ejem., *Juan Booleana estudia Algebra Pérez*" no es la misma oración que (a). Podemos preguntarnos entonces ¿cuáles son los "cambios de orden" permitidos?, ¿cuántos "ordenes" y "relaciones entre palabras" hay?, ¿entre los sonidos, que forman una palabra, se debe guardar un "orden"?, ¿los "ordenes" permitidos en español son los mismos que en cualquier otro idioma?. De responder a las preguntas anteriores y muchas más que se pueden plantear con respecto al lenguaje se ocupa, aunque es la única ciencia en hacerlo, la Linguística.

Los tres aspectos anteriormente mencionados dan lugar a una división de la Linguística en 3 ramas generales: Semántica, Fonología y Sintaxis. Estas ramas se ocupan de alguno de los tres aspectos generales: la Semántica se ocupa del primer aspecto, la Fonología del aspecto sonoro, y la Sintaxis del tercer aspecto. Aclararemos que no se ocupan de forma única y exclusiva de un único aspecto; aunque ha habido linguistas que han pretendido estudiar cada aspecto independientemente de los otros [30]. lo cierto es que las tres ramas generales son interdependientes. Así, por ejem., dentro de la sintaxis se recurre a nociones semánticas, al igual que en la fonología.

OTRAS DISCIPLINAS LINGUISTICAS.

Por supuesto que el lenguaje presenta otros aspectos: social, psicológico, formal, de extensión geográfica, de comunicación, fisiológico, probabilístico, etc.,..., pero el estudio de ellos generalmente es abordado con un marco de referencia dentro del cual intervienen con mayor o menor importancia, entre otros factores, alguna de las ramas generales. A manera de ejemplo, el aspecto formal es estudiado separadamente, aunque puede serlo conjuntamente dentro de la semántica, la sintaxis, y la fonología.

Los antedichos aspectos dan lugar a distintas ramas o disciplinas de la Linguística relacionadas entre sí en diferentes grados. Tal es su diversidad que "de hecho, la linguística es un conjunto considerable de disciplinas y no una SOLA" (pag.44 en [13]). Algunas de estas ramas toman conceptos y métodos de otras áreas del conocimiento, dando origen a teorías que se encuentran en la intersección de dos o más ciencias. Un ejemplo de este tipo de disciplina es la

Psicolinguística, que se ocupa de estudiar los procesos mentales que subyacen a la adquisición y uso del lenguaje. Otro ejemplo es la Linguística Matemática disciplina de reciente creación (cf. más adelante) que se encuentra en la intersección de la Matemática, la Linguísitca y las ciencias de la Computación. El "cálculo sintáctico" cuyo estudio abordamos en esta tesis, se encuadra dentro de esta disciplina.

La anterior, en apariencia, sorprendente intersección que dió lugar al nacimiento de la Linguística Matemática no fue de ningún modo fortuita, sino motivada por problemas e intereses de estudio surgidos durante el desarrollo de cada una de las tres ciencias mencionadas antes. Veamos a grandes rasgos el desarrollo histórico de la Linguística y el origen de la Linguísitca Matemática dentro de ella. Hacemos especial énfasis en la Linguística ya que para esta ciencia las teorías y resultados de la Linguística Matemática tienen gran relevancia e importancia. Más aún, algunos linguistas (veáse cap. 2 de [21]) consideran que la Linguística Matemática es la etapa más alta alcanzada hasta hoy por el desarrollo de la Linguística como ciencia.

DESARROLLO HISTORICO DE LA LINGUISTICA.

Aunque con anterioridad haya habido reflexiones y estudios sobre el lenguaje que dieron como resultado distintas obras al respecto [24], es comunmente aceptado que el establecimiento de la Linguística como ciencia fué en los inicios del siglo XIX, con el surgimiento del paradigma histórico-comparativo (paradigma en el sentido de Kuhn [33]) tenemos así la primera metodología propiamente linguística que desplaza al análisis principalmente lógico con que se abordaba en ese entonces el estudio del lenguaje; este enfoque tuvo sus raices en las ideas aristotélicas y alcanzó su máxima expresión en la obra "La Logique ou l'art de penser", mejor conocida como gramática de Port Royal, publicada en 1662 por Antoine Arnauld y Pierre Nicole, lógicos discípulos de Descartes.

La primera obra de gramática histórica-comparativa fué publicada en 1816 por Franz Bopp y la metodología utilizada dominará el panorama linguístico hasta entrado el siglo XX; así pues, durante estos años se llevó a cabo la articulación del paradigma histórico comparativo, haciéndose especial énfasis en los hechos morfológicos y fonéticos y siendo su principal característica el estudio del desarrollo, a través del tiempo, de las distintas lenguas conocidas y el es-

tablecimiento de relaciones de parentesco entre ellas.

En la segunda mitad del siglo XIX a la sombra de la linguística histórica-comparativa surge la geografía linguística y la fonética instrumental que se avocan al estudio de las variantes de una lengua y al estudio de los sonidos con ayuda de aparatos, respectivamente; estas disciplinas linguísticas presuponen alguna forma de tratamiento estadístico, por sencillo que sea, de los datos recogidos, así, se inicia el uso de métodos estadísticos en la Linguística y que desembocara en el establecimiento de la disciplina actualmente conocida como Estadística Linguística.

En el año de 1916, sale a la luz el "Cours de linguistique génerale" de Ferdinand de Saussure, que es una recopilación de sus clases de Linguística general llevada a cabo por dos de sus discípulos. Esta obra tuvo gran influencia sobre generaciones posteriores, ya que aquí se postula una teoría general del lenguaje basada en la concepción de que la linguística debía avocarse al estudio de las relaciones entre los hechos linguísticos, más que el estudio de los hechos mismos; ya que de acuerdo a Saussere estos últimos pertenecen al HABLA. En cambio las relaciones entre hechos pertenecen al deominio del habla. Entendiendo esta última como el conocimiento mental y abstracto que posee un hablante sobre su propio lenguaje y que se manifiesta concretamente en el HABLA.

Teniendo como marco general el paradigma Saussereano aparecieron varias escuelas con distintos intereses de estudio. Lo común a todas ellas es el hecho de que no buscaban leyes que reflejaran cambios observados en hechos del habla, sino buscaban estructurar en un sistema coherente las relaciones a nivel de lengua. El primer campo del lenguaje a que se aplicaron estas ideas fué el sonora, dando pie a la aparición de la Fonología. En estas fechas nace también la Semántica como disciplina independiente de la linguística, además, como una consecuencia del nuevo aire que sacudió a la Linguística con las ideas de Saussure, resurge el interés por los estudios sintácticos, en especial en la linguística norteamericana, en donde surge en 1930 una escuela netamente sintáctica basada en las teorías de L. Bloomfield. A dicha escuela es debida la técnica de análisis sintáctico conocida como "análisis en constituyentes inmediatos".

Las nuevas ideas implicaron un cambio en la metodología: como ahora había que formar sistemas estructurados se tomó como modelo a seguir los sistemas lógico-matemáticos, y hubo intentos de construir dichos modelos con métodos mate

máticos (veáse al respecto el cap. I en [29]). La escuela bloomfieldana intentó eliminar, al menos en teoría, toda referencia al significado en la descripción de los hechos linguísticos en favor de criterios materiales, observables y controlables. Con relación a la sintaxis emplearon el concepto de DISTRIBUCION como el conjunto de contextos en que el sistema característico de cada idioma permite que se presenten las unidades linguísticas. Además, basados en el Conductismo tratan de llegar al sistema de la lengua a través de los hechos observables del habla y, la realidad del habla condicionaba y determinaba a la teoría.

En 1956, surge el paradigma chomskiano con la publicación del libro ESTRUCTURAS SINTACTICAS [10] de Noam Chomsky. Dicho paradigma se encuentra actualmente en articulación y Robins, citado en [30] afirma que "*son tan rápidos, en efecto, los cambios en algunos de los conceptos y métodos teóricos... que un juicio preciso sobre su posición sería inadecuado y sin duda superado en pocos años*". En oposición al antimentalismo de la escuela bloomfieldana, N. Chomsky les reprocha sobte todo la preocupación exclusiva, exagerada y bastante estéril de segmentar los enunciados en constituyentes y de agrupar a estos últimos en clases según el criterio de distribución, y haber descuidado voluntariamente los mecanismos psicológicos subyacentes en la construcción de los enunciados por parte de los hablantes.

Para Chomsky, el punto de vista bloomfieldano convierte a la Linguísti ca en una ciencia taxonómica, es decir, meramente clasificadora, restringida a tperminos observacionales, en cambio su pretensión es que la Linguística debe ser una ciencia teórica, una ciencia en la que los conceptos teóricos juegan desde el principio un papel predominante. En la obtención de tal ciencia se debe tomar muy en cienta el hecho de la creatividad linguística, es decir, el hecho de que, en el lenguaje, medios finitos dan lugar a un número potencialmente infinito de oraciones. Así, las gramáticas generativas (veáse ejemplo de más abajo) son consideradas como teorías formales de la competencia linguística de un "hablante-oyente ideal".

Como se trata ahora de formular teorías científicas se recurre a conceptos y desarrollos de la matemática, en especial lógico-algebraicos, para realizar tal formalización. Con esta concepción, el linguista debe avocarse a la formulación de teorías que despues someterá a evaluación, para ver que tan adecua-

das son para el estudio del lenguaje. En la bibliografía al final del presente trabajo se encuentran indicados unos pocos de libros sobre el tema, en especial, los libros básicos de Chomsky.

CONCEPTOS BASICOS DE LA LINGUISTICA MATEMATICA.

La teoría de las gramáticas generativas fueron las que dieron pie para el nacimiento de la Linguística Matemática así que podemos considerar con justicia a Noam Chomsky como su creador. Algunos autores le dan un sentido más amplio al término "Linguística Matemática" [21] que el sentido restringido adoptado aquí y que es el dado por Melcuk y Gladkij en [14] . Ellos entienden por "Linguística Matemática": *un determinado círculo de trabajos matemáticos por su esencia, que han surgido de las tentativas encaminadas a describir con todo rigor los hechos de las lenguas naturales y que contienen resultados en principio útiles para la Linguística.* Más específicamente, podemos decir que el objetivo de la Linguística Matemática es la elaboración y el estudio de ciertos conceptos abstractos que se utilizan en Linguística, *en tanto que modelos de diferentes aspectos del lenguaje, y, sobre todo, en tanto que elementos básicos del metalenguaje de la ciencia linguística* (pag.19 en [29]. Con el sentido anteriormente expresado podemos decir que la Linguística Matemática es una disciplina matemática "dirigida" hacia los lenguajes naturales y hacia la Linguística; lo anterior no significa que sus resultados no tengan utilidad en otros campos, por ejemplo, en Computación, [34] y en Teoría de Autómatas [26] .

Aunque las nociones básicas que establecemos a continuación son susceptibles de recibir una interpretación fónica, aquí nos restringimos al campos de la sintaxis. Nuestras dos nociones básicas son:

1) La noción *elemento de un conjunto:* partimos siempre de la existencia de un conjunto finito no vacío

$$V = \{a_1, a_2, \ldots, a_n\}$$

que puede recibir el nombre de *alfabeto o vocabulario*, dependiendo si sus elementos son considerados letras o palabras, respectivamente. Sin pérdida de generalidad lo consideramos en el resto del trabajo como un vocabulario y a sus elementos como palabras.

2) La noción de *cadena:* entendemos por cadena, cadena sobre V, una se-

rie ordenada finita de elementos de V. La secuencia x, escrita como

$$x = a_{i_1} a_{i_2} \ldots a_{i_k} \quad ,$$

donde $i_1, i_2, \ldots, i_k$ toma los valores $1, 2, \ldots, n$, es una ordenación con repetición de n elementos tomados de k en k. Llamamos al número k la *longitud* de la cadena x y lo simbolizamos por

$$\ell(x)=k \quad \text{ó} \quad \ell(a_{i_1} a_{i_2} \ldots a_{i_k})=k$$

Podemos también definir, la *cadena vacía*, denotada Λ, como aquella cadena que tiene longitud cero.

Con las anteriores nociones y términos ya podemos establecer algunas definiciones y resultados útiles para nuestra discusión:

DEFINICION 0.1.- Dado un vocabulario V, llamaremos LENGUAJE UNIVERSAL el conjunto de todas las cadenas sobre V. El lenguaje universal es denotado por V*.

Es casi inmediato de la definición anterior y de la definición de cadena, que el lenguaje universal es un conjunto numerable; es la unión de una familia numerable de conjuntos ajenos, es decir $V^* = \bigcup_{j=0}^{\infty} A_j$, donde A_j es el conjunto de las ordenaciones con repetición de los n elementos de V tomados de j en j.

DEFINICION 0.2.- Dado un vocabulario V, llamaremos LENGUAJE SOBRE V, representado por L(V), a cualquier subconjunto de V*, es decir, $L(V) \subset V^*$

obviamente todo lenguaje sobre V es numerable.

EJEMPLO 0.1.- Dado el voacabulario $V = \{a, b\}$, un lenguaje puede ser el conjunto de cadenas

ab, aabb, aaabbb, aaaabbbb

es decir

$$L(V) = \{a^n b^n \mid \forall\, n \geq 1\}.$$

EJEMPLO 0.2.- Sea V el conjunto de las palabras de la lengua española, que puede ser enumerado mediante un diccionario. El conjunto de oraciones gramaticales sobre V, subconjunto del lenguaje universal V*, es el lenguaje natural correspondiente al español.

Notemos que al definir un lenguaje como un conjunto de cadenas podemos definir obviamente los lenguajes *unión*, *intersección* y lenguaje *complementario*, con respecto a un lenguaje universal V*. Además, al conjunto V* le podemos dar el carácter de estructura si definimos en el la operación de *concatenación* para cadenas sobre V.

DEFINICION 0.3.- Dadas dos cadenas sobre V,

$x = a_{i_1} a_{i_2} \dots a_{i_k}$

$y = a_{j_1} a_{j_2} \dots a_{j_m}$

la CONCATENACION de ellas, denotada x.y , es la cadena

$a_{i_1} a_{i_2} \dots a_{i_k} a_{j_1} a_{j_2} \dots a_{j_m}$

Observemos que la concatenación es una operación asociativa, ya que si

$x = a_{i_1} a_{i_2} \dots a_{i_k}$

$y = a_{j_1} a_{j_2} \dots a_{j_m}$

$z = a_{s_1} a_{s_2} \dots a_{s_t}$

tenemos

$$(x.y).z = (a_{i_1} a_{i_2} \dots a_{i_k} a_{j_1} a_{j_2} \dots a_{j_m}).z =$$

$$= a_{i_1} a_{i_2} \dots a_{i_k} a_{j_1} \dots a_{j_m} a_{s_1} \dots a_{s_t}$$

$$x.(y.z) = x.(a_{j_1} \dots a_{j_m} a_{s_1} \dots a_{s_t}) =$$

$$= a_{i_1} \dots a_{i_k} a_{j_1} \dots a_{j_m} a_{s_1} \dots a_{s_t}$$

así que

$$(x.y).z = x.(y.z)$$

Pero también tenemos que la cadena vacía Λ actúa como elemento neutro para la concatenación

$$\forall\, x \in V^* \ , \quad x.\Lambda = \Lambda.x = x$$

En general, la operación concatenación no es conmutativa, ya que si x= *pa*, y= *to*, xy= *pato* ≠ *topa*=yx. De todo lo anterior resulta que V* tiene estructura de monoide, es decir, de semigrupo con elemento neutro.

Dadas dos cadenas x,y es posible hablar de relaciones entre ellas, un ejemplo es la establecida en la siguiente definición.

DEFINICION 0.4.- Si x e y son cadenas sobre un vocabulario V, diremos que y es una SUBCADENA de la cadena x si existen dos cadenas z,u, sobre V tales que

$$x = z.y.u$$

La existencia de relaciones entre cadenas sobre un vocabulario V es relevante para la utilización en la Linguística de los conceptos y resultados de la Linguística Matemática. En efecto, si consideramos que la Linguística Matemática nos provee de modelos para el estudio del lenguaje, podemos decir que un modelo de un lenguaje L(V)C V* es *analítico* si considera a L(V) dado, y su propósito es obtener una descripción intrínseca de las cadenas pertenecientes a L(V), esto es, una descripción de las relaciones entre los elementos de V y entre las subcadenas con respecto a su posición en las cadenas de L(V). *"Tal punto de vista esta muy cercanamente relacionado a la teoría linguística estructural, especialmente a las así llamadas linguísticas descriptivas desarrolladas por Bloomfield, Harris, Hockett, Wells, y otros."* (pag. 2 en [23])

La gramática categorial bidireccional definida en el cap. IV cae bajo esta clasificación, en cambio, las gramáticas generativas de Chomsky no son modelos analíticos ya que su enfoque es distinto. Dichas gramáticas caen bajo la clasificación de modelos *generativos*, en tales modelos no se considera L(V) dado sino que se pretende, usando un número finito de reglas (llamadas *reglas gramaticales*), específicar o producir todas las cadenas de L(V), y solo estas cadenas, asignando a cada cadena de L(V) una descripción estructural que especifique los elementos con los que se construye la cadena, su orden, arreglo, interrelaciones y cualquiera otra información gramatical necesaria para determinar como la cadena es usada y entendida. Este punto de vista esta muy relacionado a la teoría de los sistemas formales y otros conceptos de la lógica matemática contemporánea.

De hecho, la teoría de las gramáticas generativas es la parte más acabada de la Linguística Matemática, hay estudios tanto sobre sus relaciones con la lógica matemática [15], como sobre su utilización en Linguística [14]. Aquí sólo daremos algunas definiciones y ejemplos, sin ahondar en su utilización con el fin de proveer una ilustración contrastante con la definición de gramáti-

ca categorial del cap. IV.

DEFINICION 0.5.- Una GRAMATICA DE ESTRUCTURA DE FRASE, también conocida como GRAMATICA DE CONSTITUYENTES, es una cuádrupla

$$G = \langle V, VT, \sigma, P \rangle$$

donde

V es un conjunto finito, llamado VOCABULARIO.

VT es un subconjunto de V; llamado el VOCABULARIO TERMINAL, y sus elementos, SIMBOLOS TERMINALES.

σ es un elemento distinguido de V-VT, llamado VARIABLE INICIAL.

P es un conjunto finito de pares ordenados (u,v), con u una cadena no vacía de (V-VT)* y v una cadena de V*. Las parejas son llamadas PRODUCCIONES o REGLAS DE REESCRITURA, y es usual denotarlas por $u \to v$.

Los elementos de V-VT son llamados las VARIABLES de la gramática y el conjunto llamado VOCABULARIO AUXILIAR. La manera en que puede utilizarse la gramática para producir un lenguaje L(V) es mostrada en las siguientes definiciones:

DEFINICION 0.6.- Dadas dos cadenas $w, w' \in V^*$, decimos que w' SE DERIVA DIRECTAMENTE de w, denotado $w \Rightarrow w'$, si existen palabras x, y, u, v, en V* tales que

$$w = x\ u\ y$$
$$w' = x\ v\ y$$

y

$$u \to v \in P.$$

Si hay una secuencia finita $w_0, w_1, \ldots, w_n$, $n \geq 0$, de cadenas en V* tales que $w \Rightarrow w_0$, $w_n \Rightarrow w'$, y $w_{i-1} \Rightarrow w_i$ $1 \leq i \leq n$, decimos que w' se DERIVA de w, denotando $w \overset{*}{\Rightarrow} w'$ La secuencia de cadenas w_i es llamada una DERIVACION de w' desde w.

DEFINICION 0.7.- EL LENGUAJE GENERADO POR G, denotado L(G), es

$$L(G)= \{w \in VT^* \mid \sigma \overset{*}{\Rightarrow} w \}$$

Así el lenguaje generado por G es el conjunto de cadenas de símbolos terminales que puede ser derivados desde σ.

DEFINICION 0.8.- Un lenguaje L(VT)C VT* es llamado un LENGUAJE DE CONSTITUYENTES si existe una gramática de constituyentes G tal que

$$L(VT)= L(G)$$

Imponiendo condiciones sobre las reglas de reescritura es posible definir distintos tipos de gramáticas generativas, y por tanto distintos tipos de lenguajes.

EJEMPLO 0.3.- El lenguaje de la lógica proposicional es un lenguaje de constituyentes como puede ser visto con la gramática $G=\langle V,VT,\sigma,P\rangle$ donde

$$VT=\{ p_1,p_2,\ldots,p_n ,\Lambda, V, \Rightarrow,\Leftrightarrow,(,),\sim \}$$

$$V-VT= \{E\}$$

siendo p_i, i=1,...,n símbolos de enunciados, $\sim,\Lambda,V, \Rightarrow , \Leftrightarrow$ los símbolos de las conectivas lógicas E el símbolo para "proposición".

$$\sigma=E$$

$$P= \{E \to \sim(E),\ E \to((E) \Lambda (E)),\ E \to((E)V(E)),$$

$$E \to((E) \Rightarrow(E)),\ E \to ((E)\Leftrightarrow(E)),\ E \to p_1,$$

$$E \to p_2,\ldots,E \to p_n \}$$

En aplicaciones linguísticas generalmente VT es un conjunto de palabras del lenguaje que se desea modelar, V-VT es el conjunto de variables sintácticas: *(oración)*, *(sujeto)*, *(predicado)*, *(verbo)*, *(artículo)*, *(nombre)*, *(objeto directo)* etc. y σ se entiende es *oración*. A continuación damos un ejemplo sencillo de una gramática generativa en el español.

EJEMPLO 0.4.- $G = \langle V, VT, \sigma, P \rangle$

$VT = \{$ *carro, muro, un, el, golpea* $\}$

$V\text{-}VT = \{ 0, S, P, OD, N, V, A \}$

$\sigma = 0$

donde

$0 =$ (*oración*)

$S =$ (*sujeto*)

$P =$ (*predicado*)

$OD =$ (*objeto directo*)

$N =$ (*nombre*)

$V =$ (*verbo*)

$A =$ (*Artículo*)

los paréntesis es para remarcar que son variables sintácticas y no se confundan con palabras en español.

$P = \{ 0 \rightarrow SP,\ S \rightarrow AN,\ P \rightarrow V\ OD,\ OD \rightarrow AN,$
$A \rightarrow el,\ A \rightarrow un,\ N \rightarrow carro,\ N \rightarrow muro,$
$V \rightarrow golpea \}$

Podemos generar oraciones, es decir, cadenas sobre VT, por el proceso ilustrado con la siguiente derivación

$0 \rightarrow SP$.
$SP \Rightarrow S\ V\ OD$
$SVOD \Rightarrow S\ golpea\ OD$
$S\ golpea\ OD \Rightarrow AN\ golpea\ OD$
$AN\ golpea\ OD \Rightarrow el\ N\ golpea\ OD$
$el\ N\ golpea\ OD \Rightarrow el\ carro\ golpea\ OD$
$el\ carro\ golpea\ OD \Rightarrow el\ carro\ golpea\ AN$
$el\ carro\ golpea\ AN \Rightarrow el\ carro\ golpea\ un\ N$
$el\ carro\ golpea\ un\ N \Rightarrow el\ carro\ golpea\ un\ muro$

I. PRESENTACION DEL CALCULO DE TIPOS SINTACTICOS.

Partimos de dos tipos sintácticos primitivos, el tipo n y el tipo 0, que corresponden a los sustantivos y a las oraciones respectivamente. El tipo 0 es asignado a los nombres propios, a los nombres comunes que pueden aparecer sin artículo y a expresiones compuestas que pueden aparecer en lugar de un nombre propio en una oración. Así, *Juan, María, Rafael,...* tienen cada uno tipo n; también tendrán este tipo: *leche, arroz,...*, al igual que expresiones compuestas como por ejemplo *pobre Juan*, la cual puede sustituir a *Rafael* en la oración *Rafael trabaja duro* y producir la oración *pobre Juan trabaja duro*. La oración *Rafael trabaja duro* tiene tipo 0, como lo tendrán las oraciones declarativas del español.

Usaremos en lugar de "tiene tipo sintáctico" una flecha horizontal "$\rightarrow$", y así, "*Juan* $\rightarrow n$" se leerá "*Juan* tiene tipo sintáctico n"; análogamente "*Rafael trabaja duro* $\rightarrow 0$" se leerá "*Rafael trabaja duro* tiene tipo sintáctico 0". En general "A $\rightarrow x$" se leerá como "la expresión linguística (cadena) A tiene tipo sintáctico x".

Ahora bien, de cara al problema de construir un algoritmo que nos permita distinguir oraciones, necesitamos "reflejar" de alguna manera la forma en que se obtienen dichas oraciones (cadenas) a partir de un vocabulario. Este objetivo lo logramos introduciendo una operación entre tipos sintácticos: concatenación o multiplicación.

DEFINICION 1.1.- (Operación multiplicación). Sean A,B expresiones (cadenas). Diremos que
A.B $\rightarrow x.y$ si, y sólo si, A$\rightarrow x$, B$\rightarrow y$

Se leerá "$x.y$" como "x punto y", aunque generalmente omitiremos el punto " . " escribiendo simplemente "xy". En lo que sigue lo haremos así, y sólo escribiremos el punto cuando su ausencia se preste a confusión.

La introducción de la anterior operación nos permite asignar tipos sintácticos a expresiones compuestas a partir de los tipos sintácticos de las expresiones que las componen; pero en español, además de expresiones linguísticas como las mencionadas al inicio del capítulo, expresiones a las cuales se les puede asignar alguno de los tipos primitivos, existen expresiones linguísticas como *corre, llora, golpea, atraviesa, pobre, negro, blanco, aquí, rápidamente, probablemente, por, para, entonces, y, o, con, de,...*, a las cuales no se les asigna ninguno de los dos tipos sintácticos primitivos, ni se les puede asignar un tipo usando la definición 1.1.

Así pues, ¿qué tipos sintácticos se le asignarán a dichas expresiones? ¿cómo se hará dicha asignación?. La respuesta a la primera pregunta es sencilla: les asignaremos tipos sintácticos compuestos obtenidos a partir de los tipos sintácticos primitivos por el procedimiento que se ilustra en los ejemplos que siguen más abajo. Así mismo, estos ejemplos muestran cómo efectuar la asignación de tipos a expresiones linguísticas.

EJEMPLO 1.1.- Consideremos la oración *Juan trabaja*. Sabemos que *Juan* $\rightarrow n$, y si en la oración anterior sustituimos *Juan* por cualquier otro nombre propio, o expresión de tipo n como: *Pedro, María, pobre José, perro*, etc. obtenemos una oración del español. Así, *trabaja* es una palabra que al concatenarse por la derecha con cualquier palabra o expresión de tipo n "produce" una expresión de tipo 0. Para indicar esta función de *trabaja* decimos que "*trabaja* $\rightarrow n\backslash 0$" Leemos "$n\backslash 0$" como "$n$ bajo 0".

EJEMPLO 1.2.- Por otro lado, *pobre Juan*$\rightarrow n$, ya que *pobre Juan* puede ocurrir en cualquier contexto en el cual los nombre propios ocurren. Además, expresiones como: *pobre María, pobre José, pobre Adrían, pobre perro, pobre manzana*, ..., tienen el mismo tipo, por tanto, la palabra *pobre* al concatenarse por la izquierda con la expresión de tipo n da como resultado una expresión de tipo n. Esto lo indicaremos simplemente escribiendo "*pobre* $\rightarrow n/n$". Léase "n/n" como "n sobre n".

EJEMPLO 1.3.- A partir de que *Probablemente Juan trabaja*$\rightarrow 0$, y de que si reemplazamos la oración *Juan trabaja* en la oración anterior por cualquier otra oración, por ejemplo, *Pedro llora, María estudia para el exámen*, etc. obtenemos otra oración podemos entonces establecer que *probablemente*$\rightarrow 0/0$.

EJEMPLO 1.4.- Sabemos que *Juan trabaja velozmente* $\rightarrow 0$ al igual que lo tendrán las expresiones obtenidas al reemplzar *Juan* por otra expresión de tipo n; así, *trabaja velozmente* $\rightarrow n\backslash 0$. Ahora, si sustituimos *trabaja* que tiene tipo $n\backslash 0$ por cualquier otro verbo intransitivo, como *estudiar*, *llorar*, etc., que tienen su mismo tipo, obtenemos expresiones linguísticas con tipo $n\backslash 0$ al igual que la expresión *trabaja velozmente*, por tanto, *velozmente* $\rightarrow (n\backslash 0)\backslash(n\backslash 0)$

Está claro, a partir de los ejemplos anteriores, que la formación de tipos sintácticos compuestos se logra vía dos nuevas operaciones entre tipos: una que podemos llamar operación "bajo", simbolizada por la diagonal inclinada hacia la izquierda "\"; y la otra, que llamaremos operación "sobre" simbolizada por la diagonal inclinada hacia la derecha "/". Las definiciones de dichas operaciones son las siguientes:

> *DEFINICION 1.2.-* (Operación bajo). Sea B una expresión. Diremos que $B \rightarrow x\backslash z$ (leáse "$x\backslash z$" como "x bajo z") sí, y sólo si para cualquier expresión A, tal que $A \rightarrow x$, $A.B \rightarrow z$.

> *DEFINICION 1.3.-* (Operación sobre). Sea una expresión A. Diremos que $A \rightarrow z/y$ (léase "z/y" como "z sobre y") sï, solo si para cualquier expresión B, tal que $B \rightarrow y$, $A.B \rightarrow z$.

Así mismo, es evidente en las 3 definiciones y ejemplos anteriores: que al resultado de operar con dos tipos dados x, y, le hemos concedido la categoría de tipo sintáctico; en realidad, las operaciones fueron introducidas para formar nuevos tipos sintácticos. Formalizamos lo anterior estableciendo la siguiente definisión recursiva:

> *DEFINICION 1.4.-* Si x, y son tipos sintácticos, entonces $x.y$, $y\backslash x$ y x/y tambien son tipos sintácticos.

Hasta aquí, lo que tenemos es un conjunto, cuyos miembros son tipos sintácticos, y tres operaciones para sus elementos que suscitan algunas cuestiones interesantes e importantes para nuestro propósito de construir un algoritmo. Entraremos en el estudio de ellas ilustrando algunas con ejemplos.

Una de estas cuestiones es la siguiente: sabemos que la multiplicación o concatenación entre cadenas es asociativa, por ejemplo:
(Juan . trabaja). velozmente = Juan .(trabaja . velozmente), así pues, en general, tendremos que si A→x, B→y, C→z, entonces

$$(A.B).C \rightarrow (xy)z \quad y \quad A.(B.C) \rightarrow x(yz).$$

Como (A.B).C= A.(B.C), los tipos sintácticos $(xy)z$ y $x(yz)$ deben estar relacionados, en este caso deberían ser equivalentes en algún sentido ya que la igualdad entre las cadenas que las originaron se mantiene siempre.

Para indicar esta relación entre tipos sintácticos introducimos el signo "⇒", tal que, para tipos sintácticos, x,y, arbitrarios, "$x \Rightarrow y$" se leerá como "toda expresión (cadena) de tipo x tiene tambien tipo y". Así, tendremos en general que

$$(xy)z \Rightarrow x(yz)$$

y

$$x(zy) \Rightarrow (xy)z \ ,$$

que podemos resumir como

$$(xy)z \Leftrightarrow x(yz),$$

donde "$x \Leftrightarrow y$" significa que "$x \Rightarrow y$ y $y \Rightarrow x$"

La relación anterior entre tipos sintácticos no es la única que se puede establecer, como lo ilustran los ejemplos siguientes:

EJEMPLO 1.5.- Sabemos que

$$Juan\ corre \rightarrow 0,$$

pero por la definición 1.1.

$$Juan\ corre \rightarrow n.(n\backslash 0),$$

ya que

$$Juan \rightarrow n \quad y \quad corre \rightarrow n\backslash 0$$

Así pues, *Juan corre* tiene dos tipos y es natural que entre ellos exista alguna relación, que en este caso consistirá en establecer la regla

$$n.(n\backslash 0) \Rightarrow 0$$

que nos dice simplemente que cuando una expresión de tipo $n\backslash 0$ es precedida por una expresión de tipo n produce una expresión cuyo tipo es también tipo 0.

EJEMPLO 1.6.- De la definición 1.1 y de

$$Juan \rightarrow n,\ pobre \rightarrow n/n$$

deducimos que

$$pobre\ Juan \rightarrow (n/n).n;$$

pero de la forma en que se asignan los tipos primitivos sabemos que

$$Pobre\ Juan \rightarrow n$$

Análogamente al ejemplo anterior, tenemos una expresión: *pobre Juan* con dos tipos sintácticos entre los cuales debe existir alguna relación, que podemos indicar por la regla

$$(n/n).n \Rightarrow n$$

la cual nos dice que cuando una expresión de tipo n/n es seguida por una expresión de tipo n, produce una expresión de tipo n.

De los 2 ejemplos anteriores y de los subsecuentes se verá que podemos establecer las siguientes reglas generales

$$\text{(a)}\quad x\ .\ (x\backslash y) \Rightarrow y \qquad\qquad (\text{I})$$

$$\text{(b)}\quad (x/y).\ y \Rightarrow x$$

para tipos sintácticos x, y arbitrarios.

Las reglas generales (I) no son las únicas que existen; los ejemplos de más abajo nos darán una más. La obtención de algunas otras reglas será pospuesta hasta que hayamos establecido el "cálculo de tipos sintácticos", ya que allí aparecerán como teoremas.

EJEMPLO 1.7.- *Juan corre y María estudia* $\rightarrow 0$

de:

$$Juan \rightarrow n\ ,\quad corre \rightarrow n\backslash 0,$$
$$María \rightarrow n,\quad estudia \rightarrow n\backslash 0,$$

asociado de la manera siguiente

$$(Juan\ corre)\ (y(María\ estudia)) \rightarrow 0$$

obtenemos

$$y \rightarrow (0\backslash 0)/0;$$

en cambio, asociando

$$(Juan\ corre)\,y)\,(María\ estudia) \rightarrow 0$$

se obtiene

$$y \rightarrow 0\backslash(0/0),$$

y podemos establecer la propiedad

$$(0\backslash 0)/0 \Longleftrightarrow 0\backslash(0/0).$$

EJEMPLO 1.8.- A partir de que:
Juan $\rightarrow n$, *trabaja* $\rightarrow n\backslash 0$, *Juan trabaja para* $\rightarrow 0/n$,
Juana $\rightarrow n$, *para Juana* $\rightarrow 0\backslash 0$, *Juan trabaja* $\rightarrow 0$,
en la oración *Juan trabaja para Juana*, tenemos las siguientes asociaciones:

$$(Juan\ trabaja)\,(para\ Juana)$$

obteniendo

$$para \rightarrow (0\backslash 0)/n,$$

y

$$(Juan\ trabaja\ para)\,(Juana)$$

obteniendo

$$para \rightarrow 0\backslash(0/n).$$

Ambos tipos de *para* nos permiten establecer que

$$(0\backslash 0)/n \Longleftrightarrow 0\backslash(0/n).$$

Partiendo de los resultados de los ejemplos anteriores establecemos la regla general.

$$(x\backslash y)/z \Longleftrightarrow x\backslash(y/z) \qquad \text{(II)}$$

para tipos sintácticos cualesquiera x, y, z.

Hasta aquí, lo que tenemos es un conjunto, cuyos miembros son tipos sintácticos, tres operaciones y una relación para sus elementos. Además, hemos establecido 3 reglas generales para dichos tipos, operaciones y relación, que nos permiten pensar en la existencia de un mayor número de reglas generales. Estas reglas, como ya fue mencionado anteriormente, aparecerán como teoremas en el sistema formal que a continuación establecemos dando el "salto" desde un caso concreto a la abstracción.

CALCULO DE TIPOS SINTACTICOS

Aunque en la anterior presentación hemos partido de 2 tipos sintácticos: $0, n$, podemos, como lo haremos en la parte de aplicaciones, partir de un número finito de tipos sintácticos básicos. Así establecemos el siguiente sistema formal Σ, llamado CALCULO DE TIPOS SINTACTICOS:

Términos: Todos los tipos sintácticos son términos. Además, si x y y son términos también lo serán $(x.y)$, $(x\backslash y)$, (x/y). Omitiremos los paréntisis en aquéllos casos en que su ausencia no se preste a confusión. Por otro lado, si no se causa confusión escribiremos xy por $x.y$.

Fórmulas: $x \Rightarrow y$, donde x y y son términos.

Esquemas de Axiomas:

a) $x \Rightarrow x$

b) $(x.y).z \Rightarrow x.(y.z)$

c) $x.(y.z) \Rightarrow (x.y).z$

Reglas de Inferencia:

RI 1) Si $x.y \Rightarrow z$ entonces $x \Rightarrow z/y$

RI 2) Si $x.y \Rightarrow z$ entonces $y \Rightarrow x\backslash z$

RI 3) Si $x \Rightarrow z/y$ entonces $x.y \Rightarrow z$

RI 4) Si $y \Rightarrow x\backslash z$ entonces $x.y \Rightarrow z$

RI 5) Si $x \Rightarrow y$ y $y \Rightarrow z$ entonces $x \Rightarrow z$

donde x, y, z son metavariables para los tipos sintácticos.

Otra manera de presentar las anteriores reglas de inferencia, y en general la desmostración de los teoremas de un sistema formal, es en "diagrama de árbol". Para las reglas de inferencia anteriores sería escribirlas como a continuación está hecho:

RI 1) $\dfrac{x.y \Rightarrow z}{x \Rightarrow z/y}$ RI 2) $\dfrac{x.y \Rightarrow z}{y \Rightarrow x\backslash z}$

RI 3) $\dfrac{x \Rightarrow z/y}{x.y \Rightarrow z}$ RI 4) $\dfrac{y \Rightarrow x\backslash z}{x.y \Rightarrow z}$

RI 5) $\dfrac{x \Rightarrow y \quad y \Rightarrow z}{x \Rightarrow z}$

donde la raya horizontal separa las premisas de la conclusión.

En este cálculo pueden ser demostrados varios teoremas (que serán reglas generales) como los que siguen.

TEOREMA T1.1- $x \Rightarrow (xy)/y$

Demostración:

(1) $xy \Rightarrow xy$ axioma a).

(2) $x \Rightarrow (xy)/y$ por regla de Inferencia RI 1) de (1)

TEOREMA T1.2:- $(z/y)y \Rightarrow z$

Demostración:

(1) $z/y \Rightarrow z/y$ axioma a).

(2) $(z/y)y \Rightarrow z$ por regla de inferencia RI 3) de (1).

Nótese que al probar el Teorema T 1.2 hemos probado la regla general I b) mencionada anteriormente, no probamos I a) ya que es el dual simétrico de I b), y por tanto no ofrece dificultad dar su prueba. En lugar de dar una prueba de la regla general II establecida anteriormente, estableciendo el teorema correspondiente, hemos preferido usarla como ejemplo, posteriormente, de aplicación del algoritmo de decisión del cálculo sintáctico. Allí se verá que la regla II es demostrable dentro del cálculo y, una vez hecho lo anterior, procederemos a dar la prueba respectiva. Por lo pronto sigamos con los teoremas.

TEOREMA T1.3.- $y \Rightarrow (z/y)\backslash z$

Demostración:

(1)	$(z/y)y \Rightarrow z$	T1.2.
(2)	$y \Rightarrow (z/y)\backslash z$	de (1) por RI 2).

TEOREMA T1.4.- $(z/y)(y/x) \Rightarrow z/x$

Demostración:

(1)	$(y/x)x \Rightarrow y$	T1.2.
(2)	$y \Rightarrow (z/y)\backslash z$	T1.3.
(3)	$(y/x)x \Rightarrow (z/y)\backslash z$	de (1) y (2) por RI 5).
(4)	$(z/y)((y/x)x) \Rightarrow z$	de (3) por RI 4).
(5)	$((z/y)(y/x))x \Rightarrow (z/y)((y/x)x)$	axioma b).
(6)	$((z/y)(y/x))x \Rightarrow z$	de (4) y (5) por RI 5).
(7)	$(z/y)(y/x) \Rightarrow z/x$	de (6) por RI 1).

TEOREMA T1.5.- $(z/y) \Rightarrow (z/x)/(y/x)$

Demostración:

(1)	$(z/y)(y/x) \Rightarrow z/x$	T1.4.
(2)	$z/y \Rightarrow (z/x)/(y/x)$	de (1) por RI 1).

TEOREMA T1.6.- Si $x \Rightarrow x'$ y $y \Rightarrow y'$ entonces
$xy \Rightarrow x'y'$

Demostración:

(1)	$x'y \Rightarrow x'y$	axioma a).
(2)	$x \Rightarrow x'$	hipótesis.
(3)	$x' \Rightarrow (x'y)/y$	de (1) por RI 1).
(4)	$x \Rightarrow (x'y)/y$	de (2) y (3) por RI 5).
(5)	$xy \Rightarrow x'y$	de (4) por RI 3).
(6)	$x'y' \Rightarrow x'y'$	axioma a).
(7)	$y' \Rightarrow x'\backslash(x'y')$	de (6) por RI 2).
(8)	$y \Rightarrow y'$	hipótesis.
(9)	$y \Rightarrow x'\backslash(x'y')$	de (7) y (8) por RI 5).
(10)	$x'y \Rightarrow x'y'$	de (9) por RI 4).
(11)	$xy \Rightarrow x'y'$	de (5) y (10) por RI 5).

BIBLIOTECA DE CIENCIAS EXACTAS Y NATURALES
SABER DE MIS HUMAS VI GRANDEZA

TEOREMA T1.7.- Si $x \Rightarrow x'$ y $y \Rightarrow y'$ entonces

$x/y' \Rightarrow x'/y$

Demostración:

(1)	$x/y \Rightarrow x/y$	axioma a).
(2)	$(x/y)y \Rightarrow x$	de (1) por RI 3).
(3)	$x \Rightarrow x'$	hipótesis.
(4)	$(x/y)y \Rightarrow x'$	de (2) y (3) por RI 5).
(5)	$x/y \Rightarrow x'/y$	de (4) por RI 1).
(6)	$x/y' \Rightarrow x/y'$	axioma a).
(7)	$(x/y')y' \Rightarrow x$	de (6) por RI 3).
(8)	$y' \Rightarrow (x/y')\backslash x$	de (7) por RI 2).
(9)	$y \Rightarrow y'$	hipótesis.
(10)	$y \Rightarrow (x/y')\backslash x$	de (8) y (9) por RI 5).
(11)	$(x/y')y \Rightarrow x$	de (10) por RI 4).
(12)	$x/y' \Rightarrow x/y$	de (11) por RI 1).
(13)	$x/y' \Rightarrow x'/y$	de (5) y (12) por RI 5).

Antes de establecer y probar los dos últimos teoremas de este capítulo daremos una definición que será de utilidad más adelante. Además, dichos teoremas involucran a la relación definida en la siguiente definición:

DEFINICION 1.5.- Dados x,y tipos sintácticos arbitrarios diremos que x es EQUIVALENTE a y, denotado "$x \Leftrightarrow y$", si, y sólo si, $x \Rightarrow y \curlyvee y \Rightarrow x$, es decir,

$x \Leftrightarrow y . \equiv . \; x \Rightarrow y \; y \; y \Rightarrow x$

TEOREMA T1.8.- La relación $\Leftrightarrow$ es una relación de equivalencia.

Demostración:

La reflexividad y la simetría es obvio que se cumplen: es inmediato de la definición de la relación.

Para probar la transitividad tomemos x,y,z tipos sintáticos arbitrarios. Queremos probar lo siguiente:

si $x \Leftrightarrow y$ y $y \Leftrightarrow z$ entonces $x \Leftrightarrow z$

Prueba:

(1)	$x \Rightarrow y$	de la hipótesis y la
(2)	$y \Rightarrow z$	definición 1.5.

(3)	$x \Rightarrow z$	de (1) y (2) por RI 5).
(4)	$z \Rightarrow y$	de la hipótesis
(5)	$y \Rightarrow x$	por la def. 1.5.
(6)	$z \Rightarrow x$	de (4) y (5) por RI 5).
(7)	$x \Leftrightarrow z$	de (3) y (6) aplicando def. 1.5.

TEOREMA T1.9.- Si $x \Leftrightarrow x'$ y $y \Leftrightarrow y'$ entonces

$xy \Leftrightarrow x'y'$

Demostración:

(1)	$x \Rightarrow x'$	de la hipótesis
(2)	$y \Rightarrow y'$	por def. 1.5.
(3)	$xy \Rightarrow x'y'$	de (1) y (2) por T1.6.
(4)	$x' \Rightarrow x$	de la hipótesis
(5)	$y' \Rightarrow y$	por def. 1.5.
(6)	$x'y' \Rightarrow xy$	de (1) y (2) por T1.6.
(7)	$xy \Leftrightarrow x'y'$	de (3) y (6) por def. 1.5.

Los teoremas hasta aquí establecidos son suficientes para darnos cuenta del tipo de resultados que se presentan en el CALCULO DE TIPOS SINTACTICOS. De cara al problema planteado por Lambek en de obtener un algoritmo para distinguir las oraciones de expresiones que no son oraciones, pasamos a estudiar el CALCULO desde el punto de vista de la Lógica. Principalmente nos avocamos a resolver el PROBLEMA DE DECISION.

II. PROCEDIMIENTO DE DECISION.

Dado un sistema formal se presenta el problema de hallar un algoritmo *"o método general que nos permita decidir para todo enunciado particular formulado en términos de esta teoría, si éste puede o no demostrarse en ella. Este importante problema es conocido como el PROBLEMA DE DECISION"*. [ ]. Para el caso particular del cálculo sintáctico se convierte en saber si una expresión o fórmula dada, $x \quad y$, es deducible de los axiomas y reglas de inferencia establecidos.

El cálculo sintáctico no tiene un procedimiento de decisión obvio. Si bien las fórmulas demostrables

$$(x/y)y \Rightarrow x, \quad x(x\backslash y) \Rightarrow y$$

sugieren un método para simplificar expresiones decreciendo su longitud, esta sugestión es contradicha por las fórmulas, también probables o demostrables dentro del cálculo,

$$x \Rightarrow (xy)/y, \qquad y \Rightarrow x\backslash(xy),$$
$$x \Rightarrow (y/(x\backslash y), \quad y \Rightarrow (x/y)\backslash x.$$

sin embargo, aunque el método de decisión no es obvio si existe, y es establecido adoptando un método propuesto primeramente por G. Gentzen para el cálculo proposicional intuicionista [ ].

Empecemos por establecer la definición siguiente.

DEFINICION 2.1.- Dados los tipos sintácticos $x_1, x_2, x_3, \ldots, x_n, y$, llamaremos SEQUENT[1] a la expresión o fórmula

$$x_1, x_2, x_3, \ldots, x_n \Rightarrow y$$

donde "$x_1, x_2, \ldots x_n$" significa "$(\ldots((x_1 x_2)x_3)\ldots)x_n$".

Por supuesto que hay varias maneras distintas de agrupar la cadena $x_1 x_2 \ldots x_n$, y por lo tanto varios productos posibles, pero como muestra el siguiente teorema, la manera que se escoga para el agrupamiento de una cadena dada, para este cálculo, es irrelevante ya que se obtiene, en cualquier caso, una cadena

agrupada equivalente, en el sentido de la definición 1.5, a la cadena $x_1, x_2, \ldots, x_n$ de la definición 2.1.

Será notado que el teorema T2.1 depende esencialmente de los axiomas b) y c) del sistema formal que nos ocupa. En un sistema formal, en el cual se eliminaran dichos axiomas, y se conservaran el otro axioma junto con las reglas de inferencia, se tendría también un algoritmo de decisión cuya obtención, salvo algunos detalles, es muy parecida a la que desarrollamos posteriormente.

TEOREMA T2.1.- Sea x cualquiera de los productos posibles de las $x_i, i = 1, 2, \ldots, n$ obtenido de agrupar la cadena $x_1 x_2 \ldots x_n$ de alguna manera, entonces

$$x \Longleftrightarrow x_1, x_2, \ldots, x_n$$

para tipos $x_1, x_2, \ldots, x_n$ dados.

Demostración:

Puesto que "$x_1, x_2, \ldots, x_n$" es otra manera de escribir "$(\ldots((x_1 x_2)x_3)\ldots)x_n$" probaremos que

$$x \Longleftrightarrow (\ldots((x_1 x_2)x_3)\ldots)x_n$$

La prueba será por el principio de inducción generalizado.

Para n=1,2 se cumple trivialmente pues no hay más que un sola (única) manera de agrupar la cadena.

Para n=3 hay dos maneras de agrupar la cadena: $(x_1 x_2)x_3$, $x_1(x_2 x_3)$ demostraremos que son equivalentes en el sentido de la def. 1.5.

(1)	$(x_1 x_2)x_3 \Rightarrow x_1(x_2 x_3)$	axioma b)
(2)	$x_1(x_2 x_3) \Rightarrow (x_1 x_2)x_3$	axioma c)
(3)	$(x_1 x_2)x_3 \Leftrightarrow x_1(x_2 x_3)$	de (1) y (2) por def.1.5.

Antes de pasar al caso n>3, a manera de ilustración del esquema general de la demostración, demostraremos que si se cumple para n=3 entonces también se cumple para n=4.

Para este caso hay 5 maneras distintas de agrupar la cadena $x_1 . x_2 . x_3 . x_4$:

a) $((x_1 x_2) x_3) x_4$ d) $x_1 ((x_2 x_3) x_4)$

b) $(x_1 (x_2 x_3)) x_4$ e) $x_1 (x_2 (x_3 x_4))$

c) $(x_1 x_2)(x_3 x_4)$

y veremos que b), c), d) y e) son equivalentes a a).

a) $\Longleftrightarrow$ b) $(x_1 (x_2 x_3)) x_4 \Longleftrightarrow ((x_1 x_2) x_3) x_4$

(1) $x_1 (x_2 x_3) \Longleftrightarrow (x_1 x_2) x_3$ por lo que se probó en caso n=3

(2) $x_4 \Longleftrightarrow x_4$ T1.8.

(3) $(x_1 (x_2 x_3)) x_4 \Longleftrightarrow ((x_1 x_2) x_3) x_4$ de (1) y (2) por T1.9.

c) $\Longleftrightarrow$ a) $(x_1 x_2)(x_3 x_4) \Longleftrightarrow ((x_1 x_2) x_3) x_4$

(1) $(x_1 x_2)(x_3 x_4) \Rightarrow ((x_1 x_2) x_3) x_4$ axioma c)

(2) $((x_1 x_2) x_3) x_4 \Rightarrow (x_1 x_2)(x_3 x_4)$ axioma b)

(3) $(x_1 x_2)(x_3 x_4) \Longleftrightarrow ((x_1 x_2) x_3) x_4$ de (1) y (2) por def. 1.5.

d) $\Longleftrightarrow$ a) $x_1 ((x_2 x_3) x_4) \Longleftrightarrow ((x_1 x_2) x_3) x_4$

(1) $x_1 ((x_2 x_3) x_4) \Rightarrow (x_1 (x_2 x_3)) x_4$ axioma c)

(2) $(x_1 (x_2 x_3)) x_4 \Rightarrow x_1 ((x_2 x_3) x_4)$ axioma b)

(3) $x_1 ((x_2 x_3) x_4) \Longleftrightarrow (x_1 (x_2 x_3)) x_4$ de (1) y (2) por def. 1.5.

(4) $(x_1 (x_2 x_3)) x_4 \Longleftrightarrow ((x_1 x_2) x_3) x_4$ b) $\Longleftrightarrow$ a).

(5) $x_1 ((x_2 x_3) x_4) \Longleftrightarrow ((x_1 x_2) x_3) x_4$ de (3) y (4) por T1.8

e) $\Longleftrightarrow$ a) $x_1 (x_2 (x_3 x_4)) \Longleftrightarrow ((x_1 x_2) x_3) x_4$

(1) $x_1 \Longleftrightarrow x_1$ T1.8

(2) $(x_2 x_3) x_4 \Longleftrightarrow x_2 (x_3 x_4)$ caso n=3

(3) $x_1 ((x_2 x_3) x_4) \Longleftrightarrow x_1 (x_2 (x_3 x_4))$ de (1) y (2) por T1.9

(4) $x_1 ((x_2 x_3) x_4) \Longleftrightarrow ((x_1 x_2) x_3) x_4$ d) a).

(5) $x_1 (x_2 (x_3 x_4)) \Longleftrightarrow ((x_1 x_2) x_3) x_4$ de (3) y (4) por T1.8.

Ahora establezcamos la hipótesis de inducción:

$$\forall\ k < n,\ x \Longleftrightarrow (\ldots((x_1 x_2) x_3)\ldots) x_k$$

donde x es un agrupamiento de la cadena $x_1 x_2 \ldots x_k$.

Para n tenemos en última instancia que $x = a_1 . a_2$, ya que el producto o concatenación es una operación binaria, donde

$a_1 = x_1$ ó a_1 = producto de k tipos $x_1, x_2, \ldots x_k$ $k > 1$.

ó

$a_2 = x_n$ ó a_2 = producto de n-k tipos $x_{k+1}, x_{k+2}, \ldots, x_n$.

Veamos los tres casos:

1) caso:

a_2 = producto de n-k tipos $x_{k+1}, x_{k+2}, \ldots, x_n$ y a_1 = producto de k tipos $x_1, x_2, \ldots, x_k$, $k>1$.

Por hipótesis de inducción

$$a_2 \Longleftrightarrow ((\ldots ((x_{k+1} \; x_{k+2}) x_{k+3}) \ldots) x_{n-1}) x_n$$

y

$$a_2 \Longleftrightarrow a_1' \; . \; x_n$$

donde $a_1' = (\ldots ((x_{k+1} \; x_{k+2}) x_{k+3}) \ldots) x_{n-1}$,

por lo tanto, de

$$a_1 \Longleftrightarrow a_1$$

$$a_2 \Longleftrightarrow a_1' \; . \; x_n$$

llegamos, por T1.9, a

$$x = a_1 . \; a_2 \Longleftrightarrow a_1 . \; (a_1' \; . \; x_n)$$

luego entonces

$$x \Longleftrightarrow a_1 . \; (a_1' \; . \; x_n)$$

y de

$$a_1 . (a_1' \; . \; x_n) \Longleftrightarrow (a_1 . \; a_1') \; . \; x_n$$

por T1.8 obtenemos

$$x \Longleftrightarrow (a_1 . \; a_1') \; . \; x_n \qquad (1)$$

siendo $a_1 . \; a_1'$ producto de n-1 factores por la hipótesis de inducción tenemos

$$a_1 . \; a_1' \Longleftrightarrow (\ldots ((x_1 x_2) x_3) \ldots) x_{n-1},$$

y de lo anterior junto con

$$x_n \Longleftrightarrow x_n ,$$

por T1.9, tenemos

$$(a_1 . \; a_1') \; . \; x_n \Longleftrightarrow ((\ldots ((x_1 x_2) x_3) \ldots) x_{n-1}) x_n$$

que junto con (1) aplicándoles T1.8 nos lleva al resultado deseado

$$x \Longleftrightarrow ((\ldots ((x_1 x_2) x_3) \ldots) x_{n-1}) x_n$$

2) caso:

$a_2 = x_n$, en este caso a_1 = producto de los n-1 tipos $x_1, x_2, \ldots, x_{n-1}$

y por la hipótesis de inducción

$$a_1 \Longleftrightarrow (\ldots((x_1x_2)x_3)\ldots)x_{n-1}$$

que con

$$x_n \Longleftrightarrow x_n$$

por Tl.9 nos da

$$a_1 . x_n \Longleftrightarrow ((\ldots((x_1x_2)x_3)\ldots)x_{n-1}).x_n ,$$

es decir,

$$x \Longleftrightarrow (\ldots((x_1x_2)x_3)\ldots)x_n .$$

3) caso:

$a_1 = x_1$, en este caso a_2 = producto de n-1 factores $x_2, x_3, \ldots, x_n$

y por hipótesis de inducción

$$a_2 \Longleftrightarrow (\ldots((x_2x_3)x_4)\ldots)x_n$$

$$a_2 \Longleftrightarrow a_2' . x_n$$

dónde

$$a_2' = (\ldots((x_2x_3)x_4)\ldots)x_{n-1}$$

de lo anterior y de

$$a_1 \Longleftrightarrow a_1$$

por Tl.9 tenemos

$$a_1 . a_2 \Longleftrightarrow a_1 . (a_2' . x_n)$$

que con

$$a_1 . (a_2' . x_n) \Longleftrightarrow (a_1 . a_2') . x_n$$

por transitividad nos da

$$a_1 . a_2 \Longleftrightarrow (a_1 . a_2') . x_n \qquad (2)$$

siendo $a_1 . a_2'$ un producto de n-1 tipos: $x_1, x_2, \ldots, x_{n-1}$, tenemos, por hipótesis de inducción, que

$$a_1 . a_2' \Longleftrightarrow (\ldots(x_1x_2)x_3)\ldots)x_{n-1}$$

que junto con

$$x_n \Longleftrightarrow x_n ,$$

por Tl.9 nos da

$$(a_1a_2')x_n \Longleftrightarrow ((\ldots((x_1x_2)x_3)\ldots)x_{n-1})x_n$$ que con (2) por transitividad hace que obtengamos

$$a_1a_2 \Longleftrightarrow ((\ldots((x_1x_2)x_3)\ldots)x_{n-1})x_n ,$$

es decir,

$$x \Longleftrightarrow ((\ldots((x_1x_2)x_3)\ldots)x_{n-1})x_n$$

BIBLIOTECA DE CIENCIAS EXACTAS Y NATURALES
EL SABER DE MIS HIJOS HARA MI GRANDEZA

Con este último caso terminamos la demostración ya que por el principio de inducción para toda n se tendrá

$$x \Longleftrightarrow x_1, x_2, \ldots, x_n$$

donde x =producto agrupado de n tipos: $x_1, x_2, \ldots, x_n$.

COROLARIO T2.2.- Dados los tipos $x_1, x_2, \ldots, x_n, y$ tenemos la sequent $x_1, x_2, \ldots, x_n \Rightarrow y$ si, y sólo si, $x \Rightarrow y$.
Donde x es el producto de los n tipos obtenido de alguna manera de agrupar la cadena $x_1 x_2 \ldots x_n$.

Demostración:
Probemos la parte "sólo si"

(1) $x_1, x_2, \ldots, x_n \Longleftrightarrow x$	T2.1.
(2) $x \Rightarrow x_1, x_2, \ldots, x_n$	def. 1.5 a (1).
(3) $x_1, x_2, \ldots, x_n \Rightarrow y$	hipótesis.
(4) $x \Rightarrow y$	por T1.8 de (2) y (3).

ahora la parte "si"

(1) $x_1, x_2, \ldots, x_n \Longleftrightarrow x$	T2.1.
(2) $x_1, x_2, \ldots, x_n \Rightarrow x$	def. 1.5 a (1).
(3) $x \Rightarrow y$	hipótesis.
(4) $x_1, x_2, \ldots, x_n \Rightarrow y$	de (2) y (3) por T1.8.

Así pues, la sequent es equivalente a $x \Rightarrow y$.

Una vez establecidos los resultados anteriores pasamos a establecer un sistema dormal, con procedimiento de decisión, el cual es equivalente al CALCULO DE TIPOS SINTACTICOS. En lo que sigue nos ocuparemos de probar dicha equivalencia y establecer el algoritmo de decisión. En este proceso usaremos secuencuas de tipos sintácticos que denotaremos por letras latinas mayúsculas, aceptando, además, que ellas pueden denotar secuencias vacías, es decir, aquellas que no tienen tipos sintácticos.

En el párrafo anterior hemos dado una definición implícita de SECUENCIA VACIA pero no hemos dicho qué significa una SECUENCIA. Para ésto establecemos la

DEFINICION 2.2.- Dados n tipos sintácticos $x_1, x_2, \ldots x_n$, llamaremos SECUENCIA a la cadena agrupada
$(\ldots((x_1 x_2) x_3) \ldots) x_n$

Para las secuencias podemos establecer una operación cerrada mediante la siguiente definición:

DEFINICION 2.3.- Dadas dos secuencias U, V el producto de U y V, denotado "U,V" será la secuencia obtenida yuxtaponiendo o concatenando U y V. Si U es la secuencia vacía el producto significa V y viceversa.

Con estas definiciones a la mano, ahora si establecemos el siguiente sistema formal Σ_G^2:

G-SISTEMA

G-Términos: Todos los tipos sintácticos son G-términos, al igual que todas las secuencias de tipos sintácticos.

G-Fórmula: $X \Rightarrow y$, donde X un G-término, y un tipo sintáctico. En otras palabras, las G-fórmulas son las sequents.

G-Esquema de axioma: $x \Rightarrow x$.

G-Reglas de inferencia:

GRI 1) si $T, y \Rightarrow x$ entonces $T \Rightarrow x/y$

GRI 2) si $y, T \Rightarrow x$ entonces $T \Rightarrow y\backslash x$

GRI 3) si $T \Rightarrow y$ y $U, x, V \Rightarrow z$ entonces $U,\ x/y,\ T,\ V \Rightarrow z$

GRI 4) si $T \Rightarrow y$ y $U, x, V \Rightarrow z$ entonces $U, T, y\backslash x, V \Rightarrow z$

GRI 5) si $U, x, y, V \Rightarrow z$ entonces $U, xy, V \Rightarrow z$

GRI 6) si $P \Rightarrow x$ y $Q \Rightarrow y$ entonces $P, Q \Rightarrow xy$

Donde P, Q, T, U, V son secuencias y T, P, Q no son vacías.

Nótese que las reglas GRI 1) a GRI 6) introducen una ocurrencia de una de las conectivas: $.$, $\backslash$, $/$, en la conclusión; las G-reglas 1), 2) y 6) en el lado derecho de la G-fórmula y las G-reglas 3), 4) y 5) en el lado izquierdo.

Con el fin de probar la equivalencia entre el G-sistema Σ_G y el cálculo sintáctico Σ demostraremos a continuación dos resultados importantes:

LEMA T2.3.- El axioma y las reglas de inferencia del G-sistema Σ_G son deducibles del cálculo de tipos sintácticos Σ.

Demostración:

La prueba del esquema de axioma $x \Rightarrow x$ es inmediata pues es el esquema de axioma a) de Σ.

Prueba de GRI 1) si $T, y \Rightarrow x$ entonces $T \Rightarrow x/y$.

Reemplazamos T por el producto t de sus términos y GRI 1) toma la forma

si $t, y \Rightarrow x$ entonces $t \Rightarrow x/y$,

que de acuerdo a la definición de la operación " , " es:

si $ty \Rightarrow x$ entonces $t \Rightarrow x/y$,

que no es más que la regla de inferencia RI1) de Σ.

Prueba de GRI 2) si $y, T \Rightarrow x$ entonces $T \Rightarrow y\backslash x$.

Análogamente a la prueba anterior, reemplazamos T por el producto t de sus términos y GRI 2) toma la forma

si $yt \Rightarrow x$ entonces $t \Rightarrow y\backslash x$,

que es la RI2) de Σ.

Prueba de GRI 3) si $T \Rightarrow y$ y $U, x, V \Rightarrow z$ entonces

$U, x/y, T, V \Rightarrow z$.

Tenemos cuatro casos posibles.

1) caso: U y V son secuencias vacias.

Sea t el producto de los términos en T, reemplazando en GRI 3) obtenemos el resultado

si $t \Rightarrow y$ y $x \Rightarrow z$ entonces $(x/y)t \Rightarrow z$,

cuya demostración es como sigue:

(1)	$x \Rightarrow z$	hipótesis.
(2)	$t \Rightarrow y$	hipótesis.
(3)	$x/y \Rightarrow z/t$	de (1) y (2) por T1.7
(4)	$(x/y)t \Rightarrow z$	de (3) por RI3).

2) caso: U es secuencia vacia, V no lo es.

Reemplazamos V por el producto v de sus tipos y RGI3) toma la forma

si $t \Rightarrow y$ y $xv \Rightarrow z$ entonces $\{(x/y)t\}v \Rightarrow z$,

y la prueba es:

(1)	$xv \Rightarrow z$	hipótesis.
(2)	$x \Rightarrow z/v$	de (1) por RI1).
(3)	$t \Rightarrow y$	hipótesis.
(4)	$x/y \Rightarrow (z/v)/t$	de (2) y (3) por T1.7.
(5)	$(x/y)t \Rightarrow z/v$	de (4) por RI3).
(6)	$\{(x/y)t\}v \Rightarrow z$	de (5) por RI3).

3) caso: U secuencia no vacía, V secuencia vacía.

Reemplazamos U por el producto u de sus tipos y GRI3) toma la forma

si $t \Rightarrow y$ y $ux \Rightarrow z$ entonces $\{u(x/y)\}t \Rightarrow z$,

que probamos como sigue:

(1)	$ux \Rightarrow z$	hipótesis.
(2)	$x \Rightarrow u\backslash z$	de (1) por RI2).
(3)	$t \Rightarrow y$	hipótesis.
(4)	$x/y \Rightarrow (u\backslash z)/t$	de (2),(3) por T1.7.
(5)	$(x/y)t \Rightarrow u\backslash z$	de (4) por RI3).
(6)	$u\{(x/y)t\} \Rightarrow z$	de (5) por RI4).
(7)	$\{u(x/y)\}t \Rightarrow u\{(x/y)t\}$	axioma b).
(8)	$\{u(x/y)\}t \Rightarrow z$	de (6),(7) por RI5).

4) caso: U y V, ambas no vacías.

Sean u y v el producto de los términos en U y V, respectivamente. Al reemplazar las secuencias T,U, y V por sus productos en GRI 3) obtenemos

si $t \Rightarrow y$ y $(ux)v \Rightarrow z$ entonces $\{(u(x/y)\}t\}v \Rightarrow z$,

cuya demostración es:

(1)	$(ux)v \Rightarrow z$	hipótesis
(2)	$ux \Rightarrow z/v$	de (1) por RI1).
(3)	$x \Rightarrow u\backslash(z/v)$	de (2) por RI2)
(4)	$t \Rightarrow y$	hipótesis.
(5)	$x/y \Rightarrow (u\backslash(z/v))/t$	de (3)y(4) por T1.7
(6)	$(x/y)t \Rightarrow u\backslash(z/v)$	de (5) por RI3).
(7)	$u\{(x/y)t \Rightarrow z/v$	de (6) por RI4).
(8)	$\{u(x/y)\}t \Rightarrow u\{(x/y)t\}$	axioma b).

(9) $(u(x/y))\hat{z} \Rightarrow z/v$ de (7),(8) por RI4.

(10) $((u(x/y))\hat{z})v \Rightarrow z$ de (9) por RI3).

Con este caso concluimos la prueba de GRI 3); dado que es el dual simétrico de GRI 3), no probaremos la G-regla de inferencia GRI 4) a partir de los axiomas y reglas de inferencia de Σ.

Prueba de GRI 5) si $U,x,y,V \Rightarrow z$ entonces $U, xy, V \Rightarrow z$.

Al igual que en la prueba de GRI 3) hay cuatro casos

1) caso: U,V, ambas secuencias vacías.

En este caso obtenemos:

si $x,y \Rightarrow z$ entonces $xy \Rightarrow z$,

que es inmediato de la definición 2.3.

2) caso: U es secuencia vacía, V no lo es.

Reemplazando V por el producto v de sus términos obtenemos:

si $x,y,v \Rightarrow z$ entonces $xy,v \Rightarrow z$,

con demostración:

(1) $x,y,v \Rightarrow z$ hipótesis

(2) $(xy)v \Rightarrow z$ de (1) por def. 2.1.

(3) $xy,v \Rightarrow z$ de (2) por def. 2.3.

3) caso: U es secuencia no vacía, V es secuencia vacía

Sea u el producto de los términos en U, al reemplazarla en GRI 5) nos da

si $u,x,y \Rightarrow z$ entonces $u,xy \Rightarrow z$,

cuya demostración es:

(1) $u,x,y \Rightarrow z$ hipótesis.

(2) $(ux)y \Rightarrow z$ de (1) por def 2.1.

(3) $u(xy) \Rightarrow z$ de (2) por T2.2.

(4) $u,xy \Rightarrow z$ de (3) por def 2.3.

4) caso: U,V, ambas, secuencias no vacías.

Siendo u y v los productos de los términos en las secuencias U y V, respectivamente, al sustituir en GRI 5) obtenemos

si $u,x,y,v \Rightarrow z$ entonces $u,xy,v \Rightarrow z$,

que probamos de la siguiente forma:

(1) $u,x,y,v \Rightarrow z$ hipótesis.

(2) $((ux)y)v \Rightarrow z$ de (1) por def.2.1.

(3) $(u(xy))v \Rightarrow z$ de (2) por T2.2.

(4) $u,xy,v \Rightarrow z$ de (3) por def.2.3.

Prueba de GRI 6) si $P \Rightarrow x$ y $Q \Rightarrow y$ entonces $P,Q \Rightarrow xy$

Sustituyendo P y Q en GRI 6) por el producto de sus términos p y q, respectivamente, obtenemos

si $p \Rightarrow x$ $q \Rightarrow y$ entonces $pq \Rightarrow xy$,

que es simplemente el teorema ya probado T1.6.

Con esto terminamos la demostración del lema T2.3. Ahora pasaremos a demostrar el recíproco, pero para hacerlo establecemos la siguiente regla adicional:

Regla de Reducción

si $T \Rightarrow x$ y $U,x,V \Rightarrow y$ entonces $U,T,V \Rightarrow y$,

que después probaremos (METATEOREMA MT3.3.) que no incrementa el conjunto de teoremas deducibles del G-sistema Σ_G. La introducción de dicha regla tiene la finalidad de simplificar las demostraciones del lema siguiente:

LEMA T2.4.- El cálculo de tipos sintácticos Σ es deducible del G-sistema Σ_G.

Demostración:

El esquema de axioma a) $x \Rightarrow x$ no es más que el G-esquema de axioma, así que pasamos a probar los otros dos esquemas de axiomas.

Prueba de esquema de axioma b) $(xy)z \Rightarrow x(yz)$

(1) $y \Rightarrow y$ G-axioma.

(2) $z \Rightarrow z$ G-axioma.

(3) $y,z \Rightarrow yz$ de (1)y(2) por GRI 6).

(4) $x \Rightarrow x$ G-axioma.

(5) $x,y,z \Rightarrow x(yz)$ de (3)y(4) por GRI 6).

(6) $xy,z \Rightarrow x(yz)$ de (5) por GRI 5).

(7) $(xy)z \Rightarrow x(yz)$ de (6) por GRI 5).

Omitimos la deducción del esquema de axioma c) ya que es análoga a la prueba anterior.

Prueba de RI 1) si $xy \Rightarrow z$ entonces $x \Rightarrow z/y$.

(1) $y \Rightarrow y$ G-axioma.

(2)	$xy \Rightarrow z$	hipótesis.
(3)	$(xy)/y, y \Rightarrow z$	de (1)y(2) por GRI 3).
(4)	$x \Rightarrow x$	G-axioma.
(5)	$x, y \Rightarrow xy$	de (1)y(4) por GRI 6).
(6)	$x \Rightarrow (xy)/y$	de (5) por GRI 1).
(7)	$x, y \Rightarrow z$	de (3)y(6) por Regla de Reducción.
(8)	$x \Rightarrow z/y$	de (7) por GRI 1).

Prueba de RI 2) si $xy \Rightarrow z$ entonces $y \Rightarrow x\backslash z$.

(1)	$x \Rightarrow x$	G-axioma.
(2)	$xy \Rightarrow z$	hipótesis.
(3)	$x, x\backslash(xy) \Rightarrow z$	de (1)y(2) por GRI 4).
(4)	$y \Rightarrow y$	G-axioma.
(5)	$x, y \Rightarrow xy$	de (1)y(4) por GRI 6).
(6)	$y \Rightarrow x\backslash(xy)$	de (5) por GRI 2).
(7)	$x, y \Rightarrow z$	de (3)y(6) por Regla de Reducción.
(8)	$y \Rightarrow x\backslash z$	de (8) por GRI 2).

Notemos que la anterior no es la <u>única</u> prueba de RI 2). Si reproducimos la prueba de RI 1) hasta el renglón 7) $x, y \Rightarrow z$, y a este resultado le aplicamos GRI 2) obtenemos $y \Rightarrow x\backslash z$.

En general no habrá una única manera de probar un teorema en el G-sistema Σ_G, en otras palabras, es falsa la metaproposición: Todo teorema del G-sistema tiene una única demostración. Hecha esta aclaración sigamos con nuestra demostración.

Prueba de RI 3) si $x \Rightarrow z/y$ entonces $xy \Rightarrow z$.

(1)	$x \Rightarrow z/y$	hipótesis.
(2)	$y \Rightarrow y$	G-axioma.
(3)	$x, y \Rightarrow (z/y)y$	de (1)y(2) por GRI 6).
(4)	$z \Rightarrow z$	G-axioma.
(5)	$z/y, y \Rightarrow z$	de (2)y(4) por GRI 3).
(6)	$(z/y)y \Rightarrow z$	de (5) por GRI 5).
(7)	$x, y \Rightarrow z$	de (3)y(6) por Regla de Reducción.
(8)	$xy \Rightarrow z$	de (7) por GRI 5).

La prueba de RI 4) es omitida dado que es análoga a la prueba de RI 3).

Prueba de RI 5) si $x \Rightarrow y$ y $y \Rightarrow z$ entonces $x \Rightarrow z$.
Esta regla de inferencia es un caso especial de la Regla de Reducción: $T=x$ y U,V, ambas, secuencias vacías.

Habiendo probado los lemas T2.3 y T2.4, su conjunción nos establece el siguiente teorema:

METATEOREMA MT2.1.- El cálculo de tipos sintácticos Σ es equivalente al G-sistema Σ_G.

A partir del cuál se prueban fácilmente, usando además el corolario T2.2, los metaresultados siguientes:

METATEOREMA MT2.2.- La sequent $X \Rightarrow y$ es demostrable en Σ_G si, y sólo si, $x \Rightarrow y$ es demostrable en Σ.

Donde x es el producto obtenido de alguna de las maneras de agrupar o asociar la cadena $x_1 x_2 \ldots x_n$, formada con los tipos sintácticos que componen la secuencia X.

METACOROLARIO MC2.1.- $x \Rightarrow y$ es demostrable en Σ si, y sólo si, es demostrable en Σ_G.

EJEMPLO 2.1.- Para ilustrar el metacorolario anterior veamos las pruebas de la expresión $x \Rightarrow (y/x) \backslash y$ en Σ y en Σ_G.

En Σ :

$$\dfrac{\dfrac{y/x \Rightarrow y/x}{(y/x)x \Rightarrow y}}{x \Rightarrow (y/x)\backslash y}$$

En Σ_G:

$$\dfrac{\dfrac{x \Rightarrow x \qquad y \Rightarrow y}{y/x, x \Rightarrow y}}{x \Rightarrow (y/x)\backslash y}$$

Por simplificación no hemos dado los axiomas y reglas aplicados, y ambas pruebas las hemos escrito en "diagrama de árbol".

Recordamos que hemos introducido el sistema Σ_G para demostrar que el sistema original Σ tiene un procedimiento de decisión, para este fin nos será útil la siguiente definición:

DEFINICION 2.4.- Dados un G-término o una G-fórmula, su GRADO es el número total de ocurrencias de las concectivas .,/,\, en él. Las ocurrencias o apariciones de las comas "," no son contadas.

Para un G-término, o una G-fórmula, α denotamos el grado de α por "$d(\alpha)$". De la definición es claro que el grado es siempre un número natural, incluyendo al cero: $d(\alpha) \in N$. Por ejemplo, $((\nu_1 / \nu_2) \cdot \nu_2) \backslash \nu_3$ tiene grado 3, pero $((\nu_1 / \nu_2), \nu_3) \backslash \nu_3$ tiene grado 2.

Con esta definición vemos que en las G-reglas de inferencia el grado de la conclusión es más grande que la suma de los grados de las premisas, y este hecho nos permite dar un procedimiento de decisión para Σ_G.

Dicho procedimiento consiste en el método de buscar pruebas trabajando desde abajo hacia arriba a partir de la conclusión, es decir, dado un sequent $\Gamma \Rightarrow x$, intentamos construir una prueba en forma de "árbol" trabajando desde abajo hacia arriba usando el G-esquema de axioma y las G-reglas de inferencia, pero no usando la Regla de Reducción. Todo paso hacia arriba disminuye en uno el grado de la G-fórmula correspondiente, es decir, elimina una ocurrencia de una de las conectivas .,/,\, y hay solamente un número finito de maneras de hacer este paso. Por lo tanto el número total de pruebas que pueden ser intentadas es finito: para una G-fórmula α dada, si $d(\alpha)=n$ entonces el número de pruebas que se pueden intentar es $n!2^n$. La sequent $\Gamma \Rightarrow x$ es deducible si, y sólo si, una de las pruebas intentadas es válida.

Hemos establecido el siguiente resultado:

METACOROLARIO MC2.2.- Σ_G tiene un procedimiento de decisión.

Ilustramos el uso del procedimiento de decisión con un ejemplo:

EJEMPLO 2.2.- Pruebe que $(x \backslash y)/z \Leftrightarrow x \backslash (y/z)$ es deducible en Σ_G.

Para aplicar el procedimiento de decisión expresamos la fórmula como

$$(x\backslash y)/z \Rightarrow x\backslash(y/z) \qquad x\backslash(y/z) \Rightarrow (x\backslash y)/z.$$

Primero intentamos construir una prueba para

$$(x\backslash y)/z \Rightarrow x\backslash(y/z),$$

que tiene grado 4. Para eliminar alguna de las conectivas del lado izquierdo de $\Rightarrow$ necesitaríamos utilizar las G-reglas de inferencia GRI 3) y GRI 4) que exigen la aparición de una secuencia T no vacía, requisito que en este caso no se cumple, por lo que intentamos eliminar una concectiva del lado derecho. Vemos que la GRI 2) es la que introduciría la conectiva $\backslash$ al lado derecho de $\Rightarrow$, por lo que si suponemos que dicha G-regla de inferencia fue la que se aplicó obtenemos:

$$\frac{x,(x\backslash y)/z \Rightarrow y/z}{(x\backslash y)/z \Rightarrow x\backslash(y/z)} \quad \text{GRI 2)}$$

donde la expresión arriba de la raya horizontal es de grado 3. Tratando de disminuir el grado, y construir una prueba, suponemos que en el paso anterior, que nos lleva a

$$x,(x\backslash y)/z \Rightarrow y/z$$

durante alguna demostración, se aplicó la GRI 1) a unas premisas, aún por determinar, obteniendo así:

$$\frac{\dfrac{x,(x\backslash y)/z,z \Rightarrow y}{x,(x\backslash y)/z \Rightarrow y/z}\ \text{GRI 1)}}{(x\backslash y)/z \Rightarrow x\backslash(y/z)}\ \text{GRI 2)}$$

La expresión superior en el árbol anterior es de grado 2, para reducir el grado podemos suponer que se aplicó la GRI 3) con las premisas T=z, U=x y V la secuencia vacía, llegando a:

$$\frac{\dfrac{\dfrac{z \Rightarrow z \qquad x,x\backslash y \Rightarrow y}{x,(x\backslash y)/z,z \Rightarrow y}\ \text{GRI 3)}}{x,(x\backslash y)/z \Rightarrow y/z}\ \text{GRI 1)}}{(x\backslash y)/z \Rightarrow x\backslash(y/z)}\ \text{GRI 2)}$$

donde el grado de $x,x\ y \Rightarrow z$ es igual a 1. Podemos reducir el grado si suponemos que para obtener la anterior expresión se utilizó la GRI 4) con T=x,U y V, ambas, secuencias vacías, obteniendo:

$$\dfrac{\dfrac{\dfrac{\dfrac{x \Rightarrow x \qquad y \Rightarrow y}{\text{GRI 4)}}}{z \Rightarrow z \qquad x, x\backslash y \Rightarrow y}\ \text{GRI 3)}}{x, (x\backslash y)/z, z \Rightarrow y}\ \text{GRI 1)}}{x, (x\backslash y)/z \Rightarrow y/z}\ \text{GRI 2)}$$

$$(x\backslash y)/z \Rightarrow x\backslash(y/z)$$

que es una prueba válida y, por tanto, la expresión

$$(x\backslash y)/z \Rightarrow x\backslash(y/z)$$

es deducible en Σ_G. Nótese que el procedimiento de decisión no tan sólo dice que una expresión o fórmula es deducible, sino que también da una prueba en Σ_G de la expresión a la que se aplica. Además, por el metacorolario MC2.1. tenemos que la expresión es deducible en Σ.

En segundo lugar, la expresión

$$x\backslash(y/z) \Rightarrow (x\backslash y)/z$$

es también deducible en Σ_G, y por metacorolario MC2.1. es deducible así mismo en Σ, como vemos de la prueba:

$$\dfrac{\dfrac{\dfrac{\dfrac{z \Rightarrow z \qquad y \Rightarrow y}{\text{GRI 3)}}}{x \Rightarrow x \qquad (y/z), z \Rightarrow y}\ \text{GRI 4)}}{x, x\backslash(y/z), z \Rightarrow y}\ \text{GRI 2)}}{x\backslash(y/z), z \Rightarrow x\backslash y}\ \text{GRI 1)}$$

$$x\backslash(y/z) \Rightarrow (x\backslash y)/z$$

obtenida aplicando el procedimiento de decisión.

Como las dos expresiones originales son deducibles en Σ_G, y en Σ, tenemos que su conjunción

$$(x\backslash y)/z \Leftrightarrow x\backslash(y/z)$$

es deducible tanto en Σ_G como en Σ.

Hasta aquí tenemos el sistema Σ_G con procedimiento de decisión, pero es obvio, usando el metacorolario MC2.1, el resultado que nos interesaba al inicio del capítulo:

METACOROLARIO MC2.3.- El sistema Σ tiene un procedimiento de decisión.

Dada una expresión o fórmula $x \Rightarrow y$ de Σ, el procedimiento para decidir si es deducible en Σ, o no lo es, consta de los siguientes pasos:

1) Aplique el procedimiento de decisión en Σ_G para decidir si $x \Rightarrow y$ es o no es deducible en Σ_G.

2) Si la expresión es deducible en Σ_G entonces también es deducible en Σ. En caso de que no sea deducible en Σ_G tampoco lo es en Σ.

Si estamos interesados en dar una prueba de $x \Rightarrow y$ en Σ no es difícil hacerlo a partir de la prueba en Σ_G: viendo la demostración del lema T2.3 traducimos los pasos de la demostración Σ_G a pasos en Σ, con aplicación de las reglas de inferencia de Σ, obteniendo así la demostración buscada.

Este es el momento adecuado para recordar que antes de demostrar el lema T2.4 introducimos la Regla de Reducción, e hicimos la afirmación de que su introducción no incrementaba el conjunto de teoremas deducibles en Σ_G. Observando que en la prueba del lema T2.4 la Regla de Reducción se aplicó a premisas que eran G-fórmulas demostrables dentro de Σ_G, podemos ver que nuestra afirmación es cierta si probamos el siguiente teorema, llamado en honor a G. Gentzen TEOREMA DE GENTZEN:

METATEOREMA MT2.3.- si $T \Rightarrow x$ y $U, x, V \Rightarrow y$ son demostrables en Σ_G, entonces $U, T, V \Rightarrow y$ es demostrable en Σ_G.

Para probar este teorema estableceremos 2 definiciones y demostraremos un lema del cual es (el teorema) una consecuencia inmediata.

DEFINICION 2.5.- Dadas las G-fórmulas $T \Rightarrow x$ y $U, x, V \Rightarrow y$ llamaremos REDUCCION A

$$\frac{T \Rightarrow x \qquad U, x, V \Rightarrow y}{U, T, V \Rightarrow y}$$

En otras palabras, llamamos reducción a una instancia de la Regla de Reducción. Para las reducciones podemos extender el concepto de grado de una G-fórmula obteniendo un concepto que será útil en la prueba de MT2.3. Dicho concepto lo establecemos en la siguiente y última definición del capítulo:

DEFINICION 2.6.- Dada una reducción

$$\frac{T \Rightarrow x \qquad U, x, V \Rightarrow y}{U, T, V \Rightarrow y}$$

el GRADO DE LA REDUCCION es el número natural $d(T) + d(U) + d(V) + d(x) + d(y)$.

De la definición es claro que el grado de una reducción siempre es un entero no negativo. Disponiendo de estas dos definiciones ya estamos en condiciones de establecer el resultado que nos conducirá a demostrar el Teorema de Gentzen:

METALEMA MT2.4.- En cualquier reducción cuyas premisas han sido probadas sin la Regla de Reducción, la conclusión es o idéntica con una de las premisas o, de otra forma, puede ser reemplazada por una o dos reducciones con grado más pequeño.

Demostración:

Sea $\dfrac{T \Rightarrow x \qquad U, x, V \Rightarrow y}{U, T, V \Rightarrow y}$

una reducción arbitraria cuyas premisas $T \Rightarrow x$, y $U, x, V \Rightarrow y$ han sido probadas sin la Regla de Reducción.

Consideramos siete casos, los cuales no necesitan ser mutuamente excluyentes:

1) caso: La premisa $T \Rightarrow x$ es una instancia del G-esquema de axioma.

Para este caso $T=x$ y la conclusión de la reducción coincide con la otra premisa.

2) caso: La premisa $U, x, V \Rightarrow y$ es una instancia del G-esquema de axioma.

En este caso las secuencias U y V son vacías, y $x=y$, por tanto la conclusión $U, T, V \Rightarrow y$ coincide con la premisa $T \Rightarrow x$.

3) caso: El último paso en la prueba de la premisa $T \Rightarrow x$ usa una de las G-reglas de inferencia, pero no introduce la conectiva principal de x.

Como x se encuentra a la derecha del signo $\Rightarrow$, la premisa $T \Rightarrow x$ es inferida por la G-regla de inferencia GRI 3), o por la GRI 4) o por la GRI 5) dado

quents tiene la forma $T' \Rightarrow x$ con $d(T') < d(T)$. La reducción

$$\frac{T' \Rightarrow x \qquad U, x, V \Rightarrow y}{U, T', V \Rightarrow y}$$

tiene grado más pequeño que la reducción dada. Además, la G-regla de inferencia que nos lleva desde $T' \Rightarrow x$ a $T \Rightarrow x$ también nos llevara desde $U, T', V \Rightarrow y$ $U, T, V \Rightarrow y$, como verificamos para los 3 subcasos siguientes:

1) subcaso: Usamos la GRI 3).
Si se usó dicha regla, entonces
$T' = N, z, S$
$T = N, z/w, M, S$

para tipos z, w y secuencias N, M, S, dónde M no es vacía. El último paso, que nos conduce desde $T' \Rightarrow x$ a $T \Rightarrow x$, tiene la forma:

$$\frac{M \Rightarrow w \qquad N, z, S \Rightarrow x}{N, z/w, M, S \Rightarrow x}$$

y, así mismo, GRI 3) nos permite deducir $U, T, V \Rightarrow y$ desde $U, T' V \Rightarrow y$ como muestra la siguiente prueba:

(1) $M \Rightarrow w$
(2) $U, N, z, S, V \Rightarrow y$
(3) $U, N, z/w, M, S, V \Rightarrow y$ de (1)y(2) por GRI 3).

2) subcaso: Utilizamos la GRI 4).
Para este caso
$T' = N, z, S.$
$T = N, M, w \backslash z, S$

con w, z tipos sintácticos y N, S, M, secuencias, la última no vacía. El paso desde $T' \Rightarrow x$ a $T \Rightarrow x$ tiene la forma:

$$\frac{M \Rightarrow w \qquad T' \Rightarrow x}{T \Rightarrow x} \quad \text{GRI 4)}$$

y también GRI 4) nos permite inferir $U, T, V \Rightarrow y$ desde $U, T', V \Rightarrow y$:

(1) $M \Rightarrow w$
(2) $U, N, z, S, V \Rightarrow y$
(3) $U, N, M, w \backslash z, S, V \Rightarrow y$ de (1)y(2) por GRI 4)

3) subcaso: GRI 5) fue la utilizada.

Aquí

$T' = N,z,w,S$

$T = N,zw,S$

con N,S secuencias y z,w tipos. El último paso en la inferencia de $T \Rightarrow x$ desde $T' \Rightarrow x$ es:

$$\frac{N,z,w,S \Rightarrow x}{N,xw,S \Rightarrow x} \quad \text{GRI 5).}$$

y la deducción de $U,T,V \Rightarrow y$ desde $U,T',V \Rightarrow y$

(1) $U,N,z,w,S,V \Rightarrow y$

(2) $U,N,zw,S,V \Rightarrow y$

4) caso: El último paso en la prueba de $U,x,V \Rightarrow y$ usa una de las G-reglas, pero no introduce la conectiva principal de x.

Así, $U,x,V \Rightarrow y$ es inferida desde una o dos sequents, una de las cuales tiene la forma $U',x,V' \Rightarrow y'$. Puesto que la G-regla de inferencia introduce una ocurrencia de la conectiva, es decir aumenta en uno el grado de la G-fórmula,

$$d(U')+d(V')+d(y')<d(U)+d(V)+d(y).$$

De aquí tenemos que la reducción

$$\frac{T \Rightarrow x \qquad U',x,V' \Rightarrow y'}{U',T,V' \Rightarrow y'}$$

tiene grado más pequeño que la reducción dada inicialmente. Además, la misma regla que nos conduce desde $U',x,V' \Rightarrow y'$ a $U,x,V \Rightarrow y$ nos llevará desde $U',T,V' \Rightarrow y'$ a $U,T,V \Rightarrow y$, como es mostrado para los diferentes subcasos.

1) subcaso: Usamos la GRI 1)

$U' = U$

$V' = V,z$

$y = y'/z$

para secuencias U',V' y los tipos sintácticos y',z. El último paso en la deducción de $U,x,V \Rightarrow y$ desde $U',x,V' \Rightarrow y'$ sería

$$\frac{U,x,V,z \Rightarrow y'}{U,x,V \Rightarrow y'/z} \quad \text{GRI 1)}$$

y la deducción de $U,T,V \Rightarrow y$ desde $U',T,V' \Rightarrow y'$:

(1) $U',T,V' \Rightarrow y'$

(2) $U,T,V,z \Rightarrow y'$

(3) $U,T,V \Rightarrow y'/z$ de (2) por RGI 1).

2) subcaso: Utilizamos la GRI 2).

Análogamente al 1) subcaso,

$U' = z, U$

$V' = V$

$y = z \backslash y'$

con tipos sintácticos z, y' y las secuencias V', U'. En la deducción de $U, x, V \Rightarrow y$ desde $U', x, V' \Rightarrow y'$ el último paso sería

$$\frac{z, U, x, V \Rightarrow y'}{U, x, V \Rightarrow z \backslash y'} \quad \text{GRI 2).}$$

con la deducción de $U, T, V \Rightarrow y$ desde $U', T, V' \Rightarrow y'$ de la siguiente manera:

(1) $U', T, V' \Rightarrow y'$

(2) $z, U, T, V \Rightarrow y'$

(3) $U, T, V \Rightarrow z \backslash y'$ de (2) por GRI 2)

3) subcaso: Hemos usado la GRI 3).

Para este subcaso,

$V = V'$

$U = y / y', U'$

con las secuencias V', U', y el tipo sintáctico y'. El último paso en la inferencia desde $U', x, V' \Rightarrow y'$ a $U, x, V \Rightarrow y$ sería

$$\frac{U', x, V' \Rightarrow y' \qquad y \Rightarrow y}{y / y', U', x, V' \Rightarrow y} \quad \text{GRI 3)}$$

Además, la deducción de $U, T, V \Rightarrow y$ desde $U', T, V' \Rightarrow y'$ usando GRI 3) sería de la forma:

(1) $U', T, V' \Rightarrow y'$

(2) $y \Rightarrow y$

(3) $y / y', U', T, V' \Rightarrow y$ de (1)y(2) por GRI 3).

(4) $U, T, V \Rightarrow y$

4) subcaso: Hemos utilizado GRI 4).

Por analogía con el subcaso 3) tenemos,

$U = U'$

$V = V', y' \backslash y$

donde y' es tipo sintáctico y U', V', son secuencias. El último paso desde $U', x, V' \Rightarrow y'$ a $U, x, V \Rightarrow y$ es:

$$\frac{U', x, V' \Rightarrow y' \qquad y \Rightarrow y}{U', x, V', y' \backslash y \Rightarrow y} \quad \text{GRI 4)}$$

y la deducción de $U,T,V \Rightarrow y$ desde $U',T,V', \Rightarrow y'$ es:

(1) $U',T,V', \Rightarrow y'$

(2) $y \Rightarrow y$

(3) $U',T,V',y' \backslash y \Rightarrow y$ de (1)y (2) por GRI 4).

5) subcaso: La GRI 5) ha sido usada.

Tenemos:

$V = V'$

$U' = N,z,w$

$U = N,zw$

$y = y'$

para los tipos sintácticos y',z,w y las secuencias V',N. La deducción de $U,x,V \Rightarrow y$ a partir de $U',x,V' \Rightarrow y'$ con la GRI 5) es:

$$\frac{N,z,w,x,V \Rightarrow y}{N,zw,x,V \Rightarrow y} \text{ GRI 5)}$$

y la deducción desde $U',T,V' \Rightarrow y'$ de $U,T,V \Rightarrow y$ sería:

(1) $U',T,V' \Rightarrow y'$

(2) $N,z,w,T,V \Rightarrow y$

(3) $N,zw,T,V \Rightarrow y$ de (2) por GRI 5)

6) subcaso: En este último subcaso suponemos que ha sido utilizada la GRI 6).

Aquí

$V = V',z$

$U = U'$

$y = y'z$

para las secuencias U',V' y los tipos y',z. El último paso en la inferencia de $U,x,V \Rightarrow y$ desde $U',x,V' \Rightarrow y'$ sería:

$$\frac{U',x,V' \Rightarrow y' \qquad z \Rightarrow z}{U',x,V',z \Rightarrow y'z} \text{ GRI 6)}$$

la inferencia de $U,T,V \Rightarrow y$ a partir de $U',T,V' \Rightarrow y'$ es:

(1) $U',T,V' \Rightarrow y'$

(2) $z \Rightarrow z$

(3) $U',T,V',z \Rightarrow y'z$

5) caso: Los últimos pasos en las pruebas de ambas premisas de la reducción introducen la concectiva principal de $x=x'.x''$.

Para este caso podemos reemplazar la reducción

$$\dfrac{\dfrac{T' \Rightarrow x' \quad T'' \Rightarrow x''}{T', T'' \Rightarrow x'x''}\ \text{GRI 6)} \quad \dfrac{U, x', x'', V \Rightarrow y}{U, x'x'', V \Rightarrow y}\ \text{GRI 5)}}{U, T', T'', V \Rightarrow y}$$

donde x', x'' son tipos y T', T'', secuencias no vacías, por el "arbol"

$$\dfrac{\dfrac{T' \Rightarrow x' \quad U, x', x'', V \Rightarrow y}{T'' \Rightarrow x'' \quad U, T', T'', V \Rightarrow y}\ \begin{matrix}\text{Regla de Reducción}\\ \text{Regla de Reducción.}\end{matrix}}{U, T', T'', V \Rightarrow y}$$

donde aparecen dos reducciones que tienen grado más pequeño que la reducción inicial.

6) caso: Los últimos paso en las pruebas de ambas premisas $T \Rightarrow x$, $U, x, V \Rightarrow y$, introducen la conectiva principal $x = x'/x''$. La reducción

$$\dfrac{\dfrac{T, x'' \Rightarrow x'}{T \Rightarrow x'/x''}\ \text{GRI 1)} \quad \dfrac{V' \Rightarrow x'' \quad U, x', V'' \Rightarrow y}{U, x'/x'', V', V'' \Rightarrow y}\ \text{GRI 3)}}{U, T, V', V'' \Rightarrow y}$$

donde V', V'' son secuencias, la primera no vacía, y x', x'' son tipos, puede ser reemplazada por

$$\dfrac{\dfrac{T, x'' \Rightarrow x' \quad U, x', V'' \Rightarrow y}{V' \Rightarrow x'' \quad U, T, x'', V'' \Rightarrow y}\ \begin{matrix}\text{Regla de Reducción.}\\ \text{Regla de Reducción.}\end{matrix}}{U, T, V', V'' \Rightarrow y}$$

donde ambas nuevas reducciones tienen grado más pequeño que la reducción inicial.

7) caso: Este último caso es como 6) caso, excepto que $x = x''\backslash x'$ es tratado simétricamente.

Habiendo terminado la demostración del metalema podemos pasar a demostrar el metateorema que nos interesaba, que en compensación a la larga prueba anterior tiene una breve demostración:

TEOREMA DE GENTZEN.

Si $T \Rightarrow x$ y $U, x, V \Rightarrow y$ son demostrables en Σ_G, entonces $U, T, V \Rightarrow y$ es demostrable en Σ_G.

Demostración:

Sea n el grado de la reducción, puesto que ningún grado es negativo, $n \geq 0$, tenemos que aplicando a

lo más un número finito de veces el metalema MT2.4 podemos dar una prueba de $U,T,V \Rightarrow y$ en Σ_G: cuando más el metalema se aplica $\sum_{i=0}^{n-1} 2^i$ veces.

Dejando de lado la faceta formal del cálculo de tipos sintácticos pasamos a ocuparnos de ver qué aplicaciones e interpretaciones puede tener, esto lo haremos en el próximo capítulo.

RIS 71003

III. APLICACIONES E INTERPRETACIONES DEL CALCULO SINTACTICO.

A) APLICACION A UN LENGUAJE FORMAL.

La segunda mitad del siglo actual puede ser el inicio de la "época de las computadoras", debido al gran auge de su utilización en campos cada vez más amplios y diversos de la actividad humana. En particular, a partir de la década de los sesenta (a raíz, especialmente del surgimiento de la "teoría de la instrucción programada") empezó el estudio sistemático, dentro de la "tecnología educativa", de las ventajas y desventajas del uso del computador en el campo educativo. Mc Maurin, citado en [32], propone la siguiente definición de tecnología educativa: *"forma sistemática de diseñar, llevar a cabo y evaluar el proceso total de aprendizaje y enseñanza en terminos de objetivos específicos basados en la investigación sobre el aprendizaje y comunicación humanas y emplear una combinación de recursos humanos y no-humanos para conseguir una instrucción más efectiva".*

Si estamos interesados en el estudio del proceso de enseñanza-aprendizaje de alguna área del conocimiento, en este caso Matemáticas, y en el uso de la computadora durante el proceso, es posible plantearse algunas preguntas interesantes al respecto: ¿para qué partes de las Matemáticas es factible enseñarlas vía un computador?, ¿se obtienen mejores resultados, en términos de aprendizaje, con la enseñanza tradicional o con computadora?, ¿puede programarse una computadora para que plantee problemas de Matemáticas y, además diga si la respuesta del alumno es correcta o no?, etc... Responder a tales preguntas cae en el ámbito de interés de la "Matemática Educativa".

Dejando de lado cuestiones como las anteriores supongamos, en el contexto de la Matemática Educativa, que tenemos una microcomputadora capaz de plantearle a un alumno, vía la pantalla, problemas elementales del álgebra que se enseña en el nivel medio (Veáse [6] para las perspectivas en computación) del tipo: ¿cuánto vale la expresión algebraica $2^a + b$ si a=4, b=-3 ?, desarrolle $(3a + 2b)^2$ y expréselo como un trinomio, etc... Además, la microcomputado-

ra puede calificar de correcta o incorrecta la respuesta dada por el alumno; es claro que una microcomputadora con las anteriores características debe disponer de un algoritmo que le permita "reconocer" las respuestas del alumno como "expresiones algebraicas bien formadas", antes de "procesarlas" y decidir si son correctas o incorrectas.

EL CALCULO DE TIPOS SINTACTICOS puede ser utilizado para obtener, en este caso, tal ALGORITMO DE RECONOCIMIENTO de la manera esbozada en los siguientes paso: nuestro primer paso es asignar a los signos usados para formar las expresiones uno o un número finito de tipos sintácticos, esto lo podemos hacer por medio del listado o diccionario siguiente:

diccionario

números:

1,2,3,4,5,.... → e

variables:

a,b,c,d,e,.... → e

0 → c

) → p

(→ (e/p)/e, e/(ep), (c/p)/c, c/(cp).

+,- → e/e, e\e/e, e\e/c, c\e/e, c\c/c.

x → e\e/e, e\c/c, c\c/e, c\c/c.

/ → e\e/e, c\c/e.

* → e\e/e, e\e/c, c\c/e.

Formalmente el anterior diccionario es una función φ del conjunto, $\mathcal{A}$, de signos del lenguaje de las expresiones algebraicas en el conjunto, $P(\tau)$, de los subconjuntos del conjunto, τ, de los tipos sintácticos, $\varphi : \mathcal{A} \to P(\tau)$, tal que

$\varphi(0)=\{c\}$	$\varphi())=\{p\}$	$\varphi((=\{(e/p)/e, e/(ep), (c/p)/c, c/(cp)\}$
$\varphi(1)=\{e\}$	$\varphi(a)=\{e\}$	$\varphi(+)=\{e/e, e\backslash e/e, e\backslash e/c, c\backslash e/e, c\backslash c/c\}$
$\varphi(2)=\{e\}$	$\varphi(b)=\{e\}$	$\varphi(-)=\quad(+)$
$\varphi(3)=\{e\}$	$\varphi(c)=\{e\}$	$\varphi(x)=\{e\backslash e/e, e\backslash c/c, c\backslash c/e, c\backslash c/c\}$
.	.	$\varphi(/)=\{e\backslash e/e, c\backslash c/e\}$
.	.	$\varphi(*)=\{e\backslash e/e, e\backslash e/c, c\backslash c/e\}$
.	.	

Nótese que para algunos elementos de los valores de la función φ hemos omitido los paréntesis; la omisión está basada en que

$$(x\backslash y)/z \Leftrightarrow x\backslash(y/z),$$

para tipos sintácticos x, y, z, arbitrarios.

Observemos que, dada una cadena del lenguaje de las expresiones algebraicas podemos hacerle corresponder por φ un número finito de cadenas de tipos sintácticos por el sencillo expediente de asignar a cada signo de la cadena un elementos del valor de la función para dicho signo.

EJEMPLO 3.1.- Para la cadena 2/a encontramos las cadenas de tipos sintácticos que le corresponden por φ:

2	/	a	,	2	/	a
e	e\e/e	e		e	c\c/e	e

Vemos que le corresponden dos cadenas de tipos sintácticos.

Como segundo paso, de los elementos del conjunto T distinguiremos dos en especial: el tipo e y el tipo c; con la intención de "reconocer" como "expresión algebraica bien formada" a toda cadena del lenguaje de las expresiones algebraicas para la cual alguna de las cadenas de tipos sintácticos, que el corresponden por φ, tiene tipo e o c después de aplicarle las "reglas de cancelación".

El tercer paso es establecer las "reglas de cancelación", que para este caso serán los 2 teoremas del cálculo sintáctico:

1) $(x/y)\,y \Rightarrow x,$
2) $x\,(x\backslash y) \Rightarrow y.$

Las reglas de cancelación son las únicas reglas permitidas para computar el tipo de una cadena de tipos sintácticos reduciendo su longitud lo más posible. Entendemos por "lo más posible" obtener por medio de las reglas de cancelación una cadena de tipos sintácticos a la cual ya no se le puede aplicar alguna regla de cancelación y con longitud menor que la de la cadena inicial.

Antes de dar explícitamente el algoritmo de reconocimiento ilustremos su aplicación con algunos ejemplos:

EJEMPLO 3.2.- Queremos saber si la cadena

3 + (5 / a) * 2

es una "expresión algebraica bien formada" dentro del lenguaje de las expresiones algebraicas. Para decidir al respecto comenzamos por asignar a la cadena sus correspondientes cadenas de tipos sintácticos por medio de φ, obteniendo para este ejemplo un total de 120 cadenas de tipos sintácticos, entre ellas las siguientes dos:

3	+	(	5	/	a	)	*	2
e	e/e	(c/p)/c	e	e\e/e	e	p	e\e/e	e

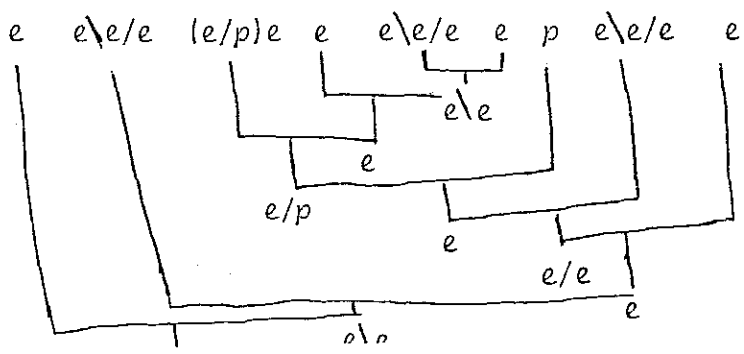
y

3	+	(	5	/	a	)	*	2
e	e\e/e	(e/p)/e	e	e\e/e	e	p	e\e/e	e

Después de computar por reducción el tipo de cada una de las 120 cadenas, si obtenemos para alguna de ellas el tipo $\mathbf{e}$ o el tipo $\mathbf{c}$ reconocemos como "expresión algebraica bien formada" la cadena 3+(5/a)*2. Computando para las anteriores dos cadenas de tipos obtenemos:

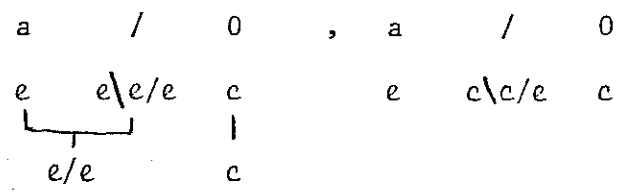
3	+	(	5	/	a	)	*	2
e	e/e	(c/p)/c	e	e\e/e	e	p	e\e/e	e
				e\e				
e	e/e	(c/p)/c	**e**			p	e\e	

donde esta última cadena no puede seguir siendo reducida, en cambio para la segunda cadena,



tenemos que el tipo computado es e y, por tanto, la cadena 3+(5/a)*2 es una "expresión algebraica bien formada".

EJEMPLO 3.3.- Ahora determinamos si la cadena a/0 es una expresión algebraica bien formada computando por reducción el tipo sintáctico de las dos cadenas de tipos sintácticos que le corresponden:



Como no obtuvimos ni el tipo e ni el tipo c concluímos que la cadena a/0 no es una expresión algebraica bien formada.

Habiendo terminado la ilustración de su aplicación procedemos a establecer el algoritmo:

ALGORITMO DE RECONOCIMIENTO.

Dada una cadena del lenguaje de las expresiones algebraicas, para decidir si es una expresión algebraica bien formada aplicamos el procedimiento siguiente:

I.- Encuentre todas las cadenas de tipos sintácticos que se le asignan por φ a la cadena dada.

II.- Para cada cadena de tipos sintácticos compute el tipo sintáctico de longitud mínima.

III.- Si para alguna cadena de tipos el tipo computado es e o c la cadena del lenguaje de las expresiones algebraicas es bien formada. En caso contrario, la cadena no es bien formada.

El algoritmo de reconocimiento establecido se aplica exclusivamente a cadenas del lenguaje de las expresiones algebraicas, aunque, por supuesto, dicho lenguaje formal no es el único para el cual es posible establecer un algoritmo de reconocimiento de expresiones bien formadas. En [18] puede ser encontrada una lista de algunos otros lenguajes formales para los cuales es posible establecer un algoritmo de la naturaleza que aquí nos ocupa.

Entre dichos lenguajes se encuentra el CALCULO PROPOSICIONAL que de acuerdo a una de sus formulaciones posee variables proposicionales de tipo 0, y dos signos para negación y condicional de tipos $0/0$ y $0\backslash 0/0$, respectivamente. Así, para un curso introductorio de Lógica el disponer de un algoritmo de reconocimiento nos permite ofrecer una definición efectiva de FORMULA BIEN FORMADA: toda cadena "reconocida" por el algoritmo es una fórmula bien formada. Pasamos ahora, a ocuparnos de la aplicación del cálculo sintáctico para el establecimiento de un algoritmo de reconocimiento para el español.

B) APLICACION A UN LENGUAJE NATURAL.

Dentro de este material ya hemos visto qué y cómo asignar tipos sintácticos a palabras del español; en lo que sigue nos ocupamos de ilustrar la aplicación del algoritmo de Lambek para el reconocimiento de oraciones gramaticales del español y algunas de las cuestiones que plantea dicha aplicación. Cabe aclarar que nuestro ejemplo de aplicación no es exhaustivo y completo: quedan detalles sin tratar que rebasan los alcances y metas del presente trabajo cayendo en el ámbito de una investigación interdiciplinaria a largo plazo.

Supongamos que tenemos una cadena de palabras en español de la que queremos saber si es una oración gramatical del español. Si tenemos los tipos sintácticos de todas y cada una de las palabras que forman la cadena podemos, siguiendo la idea desarrollada en la aplicación anterior, asignar a cada palabra su tipo sintáctico y simplificar la expresión de tipos sintácticos resultante hasta obtener una expresión no reducible. En el caso de que la expresión última sea 0 la cadena de palabras en español es una oración gramatical, en caso contrario no lo es. El procedimiento aquí esbozado está mostrado en el siguiente ejemmplo:

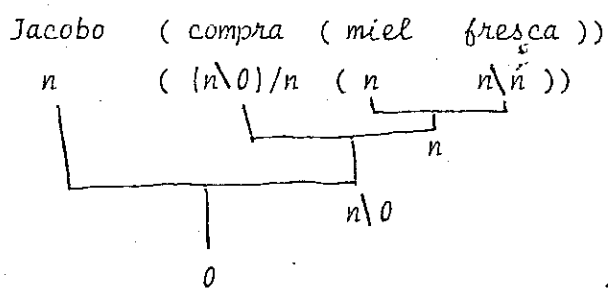
EJEMPLO 3.4.- Queremos saber si la cadena

Jacobo compra miel fresca

es una oración gramatical del español. Partiendo del diccionario:

Jacobo → n
miel → n
fresca → $n\backslash n$
compra → $(n\backslash 0)n$,

y haciendo la asignación y reducciones correspondientes



obtenemos al terminar las reducciones el tipo sintáctico 0, por ende la cadena

Jacobo compra miel fresca

es una oración gramatical del español.

Observemos en el ejemplo anterior que para efectuar la asignación hemos introducido paréntesis con el fin de hacer evidente la estructura de la expresión y facilitar las reducciones. En los lenguajes formales estudiados en Lógica el uso de paréntesus es obligatorio -sólo se omiten cuando su omisión no introduce ambiguedad o confusión- y, por tanto, la estructura de las expresiones de los lenguajes formales es evidente por lo que podemos proceder directamente a efectuar la asignación y las reducciones, cuando queremos saber si determinada expresión esta "bien formada" o no. En cambio, en los lenguajes naturales, como el español, no es usual que aparezcan paréntesis en los textos escritos y dos maneras distintas de introducir paréntesis puede llevarnos a resoluciones sintácticas esencialmente diferentes, las cuales pueden ser acompañadas por diferentes significados. A manera de ejemplo tenemos la expresión

el nido del pájaro marrón

agrupada de las dos maneras siguientes:

(el nido (del pájaro)) marrón,

el nido (del (pájaro marrón)).

Así en nuestro procedimiento mecánico de reconocimiento de oraciones debemos establecer el siguiente primer paso,

I.- Dada una expresión o cadena inserte paréntesis en todas las maneras admisibles;

después para cada una de estas expresiones con construcción paréntetica (fórmulas) procedemos como antes a asignarles tipos y efectuar reducciones. Si para alguna de las fórmulas obtenemos, despues del proceso de reducción, el tipo **O** la expresión original sin paréntesis es una oración gramatical.

La introducción de paréntesis en las expresiones no es suficiente para que el algoritmo trabaje sin problemas ya que las palabras en español pueden tener asignados más de un tipo sintáctico como muestran los ejemplos siguientes:

EJEMPLO 3.5.- Del ejemplo 1.2 sabemos que

$$Pobre \rightarrow n/n,$$

pero a partir de la oración

Juan pobre trabaja

llegamos a que *pobre* tiene otro tipo, es decir,

$$pobre \rightarrow n\backslash n.$$

Esto lo indicamos resumidamente de la siguiente manera

$$pobre \rightarrow n/n,\ n\backslash n.$$

En general, usualmente los adjetivos en español tienen ambos tipos: n/n, $n\backslash n$ aunque para cada tipo corresponde un significado distinto.

EJEMPLO 3.6.- Considerando la oración

él trabaja

y que de acuerdo al ejemplo 1.1

$$trabaja \rightarrow n\backslash O,$$

obtenemos que

$$él \rightarrow O/(n\backslash O).$$

Además, a partir de que

$$Juan\ trabaja\ para \rightarrow O/n$$

y de que

$$Juan\ trabaja\ para\ él \rightarrow O$$

llegamos a obtener otro tipo para *él*:

$$él \rightarrow (0/n)\backslash 0.$$

No asignamos a *él* tipo n ya que no puede sustituir a *Juan*, o a otra expresión de tipo n, en, por ejemplo, la oración

pobre Juan trabaja.

En cambio, lo recíproco si puede ocurrir, por tanto el nombre *Juan*, o cualquier otro nombre, tiene también los tipos del pronombre *él*. La situación anterior la simbolizamos escribiendo

$$n \Rightarrow 0/(n\backslash 0), \quad n \Rightarrow (0/n)\backslash 0.$$

La última expresión es un caso particular del teorema T1.3.

EJEMPLO 3.7.- Partiendo de las oraciones

Juan probablemente trabaja

Juan trabaja probablemente

obtenemos 3 tipos más para *probablemente*, es decir,

$$Probablemente \Rightarrow (n\backslash 0)/(n\backslash 0),\ 0\backslash 0,\ (n\backslash 0)\backslash(n\backslash 0).$$

Los dos últimos tipos se obtienen de las dos maneras distintas de agrupar la última oración y tenemos $0\backslash 0 \Rightarrow (n\backslash 0)\backslash(n\backslash 0)$, que es un caso particular del dual simétrico del teorema T1.5. Como *probablemente* es un adverbio modificador de oración, de la validez del dual simétrico de T1.5 podíamos concluir que toda expresión adverbial modificadora de oración por la derecha, de tipo $0\backslash 0$, también tendría tipo $(n\backslash 0)\backslash(n\backslash 0)$.

No tan sólo hay que aceptar que una palabra puede tener un número finito de tipos sintácticos sino que también hay palabras a las cuales habría que asignarles un número infinito de tipos sintácticos. Tal es el caso de la palabra *y*, de tipo $x\backslash x/x$, la cual enlazaría juntas expresiones de casi cualquier tipo x como parecen sugerir las oraciones siguientes:

Juan corre y María estudia,
Juan y María estudian,

Pedro tiró un libro viejo y despastado,
Jacobo golpea dura y rápidamente,
Ahora y siempre sale el sol,...

Para superar los problemas planteados por la existencia de palabras con varios tipos sintácticos establecemos el siguiente segundo paso en nuestro algoritmo.

II.- Para cada palabra de la expresión asigne uno por uno, todos los tipos permitidos por una lista finita de tipos dada.

En otras palabras, permitimos que en nuestro diccionario finito las palabras tengan asignadas un número finito de tipos. Tomamos un número finito de tipos, inclusive para palabras que potencialmente tendrían una asignación infinita, ya que para propósitos prácticos basta con un número finito, aunque sea bastante grande.

Hasta aquí han sido suficiente para efectuar las reducciones las dos reglas:

$$(x/y)y \Rightarrow x$$

$$y(y\backslash x) \Rightarrow x$$

pero el ejemplo siguiente muestra la necesidad de incorporar otras. Determinar el número mínimo de reglas para la aplicación del algoritmo de reconocimiento en el español sería una de las metas de la investigación interdiciplinaria mencionada antes.

EJEMPLO 3.8.- Si queremos analizar la oración

él ama a él

donde

$$él \rightarrow 0/(n\backslash o),\ (0/n)\backslash 0$$
$$ama \rightarrow (n\backslash 0)/n$$
$$a \rightarrow n/n,\ ((n\backslash 0)/n)\backslash((n\backslash 0)/n)$$

obtenemos cadenas con paréntesis que no pueden ser reducidas hasta obtener el tipo 0, por ejemplo:

$$\text{él} \quad ((\text{ama} \quad a) \quad \text{él})$$

$$0/(n\backslash 0) \quad (n\backslash 0)/n \quad n/n \quad (0/n)\backslash 0.$$

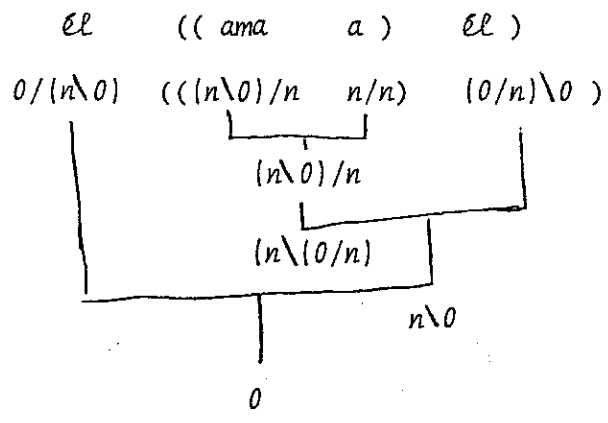
Pero si incorporamos las reglas

$$(x/y)(y/z) \Rightarrow x/z,$$

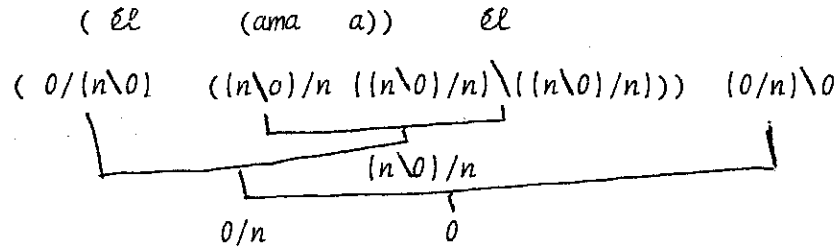
$$(x\backslash y)(y\backslash z) \Rightarrow x\backslash z,$$

$$(x\backslash y)/z \Leftrightarrow z\backslash(y/z),$$

para tipos x,y,z, la situación cambia.



Además, podemos tener resoluciones alternativas como



Habiendo mostrado la necesidad de dar más de dos reglas de cancelación, nos gustaría señalar que las supondremos dadas en un npumero finito. Dicha suposición descansa básicamente en dos razones: una de índole teórica y la otra de índole práctica, ambas razones son igualmente válidas para la aplicación a un lenguaje formal. Posponiendo para el Capítulo IV la razón teórica aclaramos la razón práctica: Si disponemos de un algoritmo de reconocimiento teóricamente susceptible de ser aplicado por un mecanismo artificial, digamos una computadora, en el que desempeñan un papel determinado las reglas de cancelación, tenemos que tomar en cuenta la capacidad limitada de memoria y, en consecuencia,

dar las reglas de cancelación, que aplicará la computadora, en un número finito.

Expuesta la razón de índole práctica pasamos a establecer explícitamente el algoritmo que dada una cadena de palabras en español decidirá si es una oración gramatical o no lo es. Posteriormente veremos una utilización del algoritmo en la Linguística.

ALGORITMO DE RECONOCIMIENTO.

I.- Dada una expresión o cadena de palabras del español inserte paréntesis en todas las maneras admisibles.

II.- Para cada palabra de la expresión asigne, uno por uno, todos los tipos permitidos por una lista finita de tipos dada.

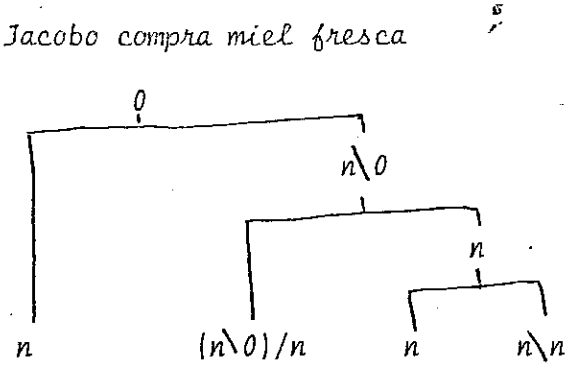
III.- Para cada agrupamiento de la cadena y asignación de tipos correspondiente, compute el tipo de la expresión completa usando las reglas de cancelación establecidas.

IV.- Si para alguno de los agrupamientos el tipo computado es 0, entonces la expresión dada originalmente es una oración gramatical. En caso contrario, no es una oración gramatical.

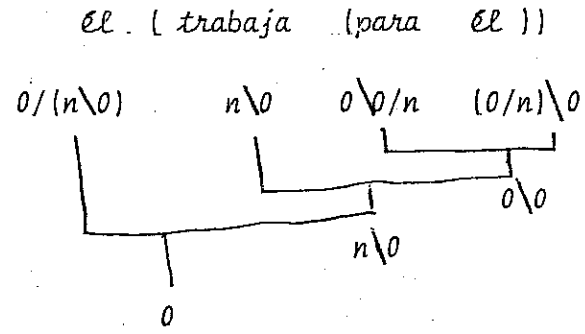
Desde el punto de vista linguístico el algoritmo de reconocimiento no sólo reconoce la corrección gramatical de una cadena sino además, cuando reconoce una oración gramaticalmente correcta, da un análisis sintáctico en grupos sintácticos, es decir, constituyentes, de ella. En efecto, será constituyente toda subcadena de la cadena dada para la cual la correspondiente cadena de tipos sintácticos en cierto paso del proceso de reconocimiento se cancele hasta un tipo sintáctico; este será el tipo sintáctico del constituyente dado. Para la oración gramaticalmente correcta del ejemplo 3.4 tenemos el sistema de 3 constituyentes (sin contar los de un solo elemento) que corresponden a tres pasos del proceso de reconocimiento:

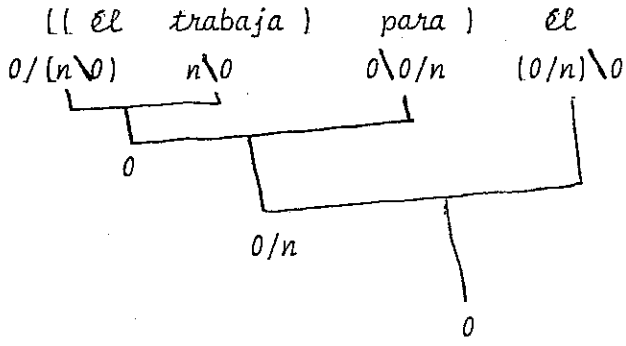
1) *miel fresca* (tipo n).
2) *compra miel fresca* (tipo $n\backslash 0$).
3) *Jacobo compra miel fresca* (tipo 0).

Para obtener el usual "árbol de constituyentes" basta con invertir el "árbol" del proceso de reconocimiento:



Observemos que se puede presentar el caso de que la oración no sea sintácticamente unívoca y que el algoritmo produzca diversos análisis, es decir, que la oración resulte asociada con distintos sistemas de constituyentes. Esto puede pasar, en primer lugar, porque la cadena de tipos sintácticos correspondiente a la oración puede reducirse según distintos procedimientos, por ejemplo:





En este ejemplo tenemos dos sistemas de constituyentes:

1) *para él*	1') *él trabaja*
2) *trabaja para él*	2') *él trabaja para*
3) *él trabaja para él*	3') *él trabaja para él*

En segundo lugar, se pueden asignar a la oración varias cadenas de tipos sintácticos, y esto también puede conducir a distintos análisis sintácticos, por tanto, a diversos sistemas de constituyentes como ocurre con la oración del ejemplo 3.8.

Podemos resumir la aplicación a la Linguística simplemente diciendo que el algoritmo nos provee de una versión rigurosa de la tradicional actividad linguística conocida como ANALISIS EN CONSTITUYENTES. Así en la enseñanza del español puede ser útil disponer de tal procedimiento mecánico de análisis sintáctico. Otra aplicación de tal procedimiento sería en el campo de la traducción automática por medio de computadoras de un lenguaje a otro: no tan sólo sería útil al reconocer el material de entrada como oraciones gramaticales del lenguaje por traducir, sino que también puede ayudar con el análisis sintáctico de dicho material.

Antes de seguir con las interpretaciones del cálculo de tipos sintácticos nos gustaría decir algo sobre otra posible aplicación de los tipos sintácticos en Linguística. Nuestras consideraciones son sobre la polisemia del término "palabra" y seguimos en ellas lo expresado por S. Stati en la primera parte del capítulo 3 del libro [30].

Con respecto a los distintos significados del término "palabra" compararemos los dos siguientes enunciados:

(a) El proverbio *Quien golpea primero, golpea dos veces* contiene seis palabras.

(b) Los matemáticos usan a menudo la palabra *suponer*.

En (a) "palabra" designa a cada uno de los elementos, gráficamente separados, del proverbio citado, *quien, golpea, dos*, etc. Es decir, las palabras son consideradas los elementos mínimos de cuya combinación resultan las oraciones. En este sentido hemos usado en el cap. I, y en este, el término "palabra" al hablar de "cadenas de palabras". Este el sentido que dentro de la Linguística Matemática, en general, se asigna al término "palabra" ya que *generalmente las frases de un lenguaje natural cuando se estudian en el plano de la teoría de las gramáticas formales, se consideran como cadenas de formas de palabras... que desempeñan el papel de símbolos terminales.* (Véase [14] en pag.69).

En cambio, en (b) "palabra" se refiere no al infinitivo *suponer*, sino al conjunto de formas gramaticales de un determinado verbo: *supongamos, hemos supuesto, suponiendo*, etcétera. En otros términos, en el enunciado (b) se habla de la frecuente recurrencia de un concepto verbal en los discursos matemáticos, y no importa en absoluto si los matemáticos lo utilizan en presente, en gerundio, o en otro tiempo verbal.

De acuerdo a Stati, *con el fin de evitar el equívoco, se ha pensado en la introducción en la terminología linguística de dos términos no ambiguos, para reemplazar el vocablo polisémico "palabra"*. Algunos estructuralistas introducen el término "lexema" como nombre de un concepto y como "alolexema" a sus distintas actuaciones formales, contrastantes desde el punto de vista gramatical. Las variantes pueden ser gramaticales, fonéticas y semánticas. Por ejemplo, *supongamos* y *suponiendo* son variantes gramaticales del lexema *SUPONER*; para el lexema *PADRE* tenemos las variantes fonéticas, *padre, papá y apá*; por último, mencionamos las variantes semánticas del lexema *GRAVE* en *la situación es grave* y *lleva el acento grave*. Concluyendo, un lexema es un conjunto de variantes gramáticales, fonéticas y semánticas, y cada una de estas variantes es un alolexema. De lo aquí discutido resulta que todo enunciado está formado por una cadena de alolexemas, cada uno de los cuales pertenece a un determinado lexema.

Con esta terminología ya podemos explicar el hecho de que a una misma forma de palabra, digamos *pobre* se le asignen distintos tipos sintácticos, *pobre* $\longrightarrow$ n/n, $n\backslash n$. Cada uno de los tipos sintácticos está indicando un alolexema del lexema respectivo. Podemos, usando los tipos sintácticos, intentar formalizar los conceptos de lexema y alolexema, definiendo en una primera aproximación los alolexemas como parejas ordenadas (A,x) donde A es una forma de palabra y x un tipo sintáctico. Con esta definición podríamos definir un lexema como un conjunto de parejas ordenadas, es decir, alolexemas.

En esta formalización un enunciado, en lugar de una cadena como

Pobre . Juan . trabaja,

sería ahora una cadena del tipo

$(\ Pobre,\ n/n\ ).(Juan,\ n).(trabaja,\ n\backslash 0),$

donde la pareja resultante es

$(\ Pobre\ .\ Juan\ .\ trabaja,\ n/n.n.n\backslash 0).$

El algoritmo de reconocimiento se aplicaría así al segundo lugar de la pareja ordenada, y si este segundo elemento se puede cancelar por medio de las reglas de cancelación para producir el tipo *0* diríamos que la cadena de formas de palabras del primer lugar es una oración gramaticalmente correcta.

Con respecto a los 3 tipos de variantes gramaticales podemos observar que las variantes fonéticas como *padre, papá* y *apá* se distinguen una de otra por tener distintas imágenes de palabras en el primer lugar de la pareja ordenada,

(*padre*, *n*)
(*papá*, *n*)
(*apá*, *n*),

y el mismo tipo sintáctico en el segundo lugar; en cambio, las variantes gramaticales se diferenciarían por tener distintos el primer y el segundo lugar de la pareja ordenada, por ejemplo,

(*supone*, *n 0*)
(*suponen*, $(0/(n\backslash 0))\backslash 0$);

por último, las variantes semánticas se distinguen por tener iguales el primer lugar y distintos los tipos sintácticos del segundo lugar:

(*pobre*, n/n)
(*pobre*, $n\backslash n$)
(*trabajo*, *n*)
(*trabajo*, $n\backslash 0$)

Nos gustaría señalar que las afirmaciones aquí hechas tienen que ser matizadas, debido a que la formalización de la que hablamos, como una aplicación de los tipos sintácticos, en realidad es una *idea* de formalización que todavía tiene que ser sujeta a un estudio y desarrollo que escapa a los fines de este trabajo: esta tarea se realizaría dentro de la Linguística Matemática Aplicada. Si aquí se ha mencionado dicha formalización ha sido con la intención de ilustrar y señalar un tema abierto a la investigación de interés tanto para linguistas, como para matemáticos.

Posponiendo para el capítulo IV la formalización y exposición del concepto subyacente tras el algoritmo de reconocimiento pasamos a ocuparnos de dar interpretaciones del cálculo sintáctico dentro de la Matemática.

C) INTERPRETACION EN EL SISTEMA AXIOMATICO ORDENADO DE LOS NUMEROS REALES.

Aunque en el capítulo I hayamos establecido el sistema formal $\sum$ para "tipos sintácticos", desde el punto de vista lógico se puede hablar de variables: $\mathcal{N}_1, \mathcal{N}_2, \ldots$ que tomarían sus valor no en el conjunto de tipos sintácticos sino en cualquier conjunto en el que estuvieran definidas tres operaciones correspondientes a $.,\backslash,/$. Al sustituir en el enunciado del cálculo sintáctico el término "tipos sintácticos" por el término "variables" obtendríamos un SISTEMA FORMAL, es decir, una estructura abstracta de relaciones entre términos sin interpretar. Para tales sistemas formales siempre es deseable poder encontrar una INTERPRETACION de ellos; en el caso que nos ocupa tal interpretación es proporcionada por los tipos sintácticos y las operaciones definidas para ellos. Podemos encontrar otra interpretación del sistema formal dentro de la Matemática, en particular, dentro de la aritmética de los números reales.

Obtenemos dicha interpretación si consideramos que los tipos sintácticos son números racionales positivos distintos del cero, el producto $x.y$ es interpretado como el producto habitual de los racionales, las operaciones x/y y $y\backslash x$ son interpretadas, ambas, como la división usual en los racionales $\frac{x}{y}$, y por último, la fórmula $x \Rightarrow y$ es interpretada como la desigualdad $x \leq y$.

Con estas interpretaciones los esquemas de axiomas y las reglas de inferencia se traducen en:

Esquemas de Axiomas:

a) $x \leq x$

b) $(xy)z \leq x(yz)$

c) $x(yz) \leq (xy)z$

Reglas de Inferencia:

RI 1) Si $xy \leq z$ entonces $x \leq \frac{z}{y}$

RI 2) Si $xy \leq z$ entonces $z \leq \frac{z}{x}$

RI 3) Si $x \leq \frac{z}{y}$ entonces $xy \leq z$.

RI 4) Si $y \leq \frac{z}{x}$ entonces $xy \leq z$.

RI 5) Si $x \leq y$ y $y \leq z$ entonces $x \leq z$.

que son todas teoremas válidos de la aritmética de los números reales y por tanto, efectivamente hemos dado una INTERPRETACION del sistema formal Σ dentro de la aritmética de los números reales. Tenemos así, por la LEY DE DEDUCCION (véase en [31] el cap.VI), que todos los teoremas demostrables en Σ al ser interpretados en ka aritmética de los números reales serán desigualdades válidas de los números racionales positivos. Además, como el sistema formal tiene procedimiento de decisión tenemos un modo mecánico para encontrar las demostraciones de las desigualdades expresadas en términos de producto, división y la relación "menor o igual".

Habiendo encontrado esta interpretación del sistema formal Σ ya podemos demostrar el siguiente resultado:

METATEOREMA MT 3.1.- El sistema formal Σ es consistente.

Demostración:

Supongamos que el sistema es inconsistente, eso quiere decir que es posible probar en el sistema una cierta expresión P, para la cual también es posible probar su negación lógica $\sim P$. Por la ley de deducción tendríamos que para toda interpretación del sistema formal encontraríamos que es posible demostrar dos enunciados contradictorios, pero para la interpretación dada en este apartado significa que existiría una desigualdad demostrable $x \leq y$ para la cual también podemos demostrar la desigualdad $x > y$, lo cual no puede pasar por la ley de la tricotomía.

En conclusión, el sistema formal Σ es consistente.

IV. MAS ALLA DEL CALCULO.

Podem ormalizar las conceptos implícitos en la parte del capítulo anterior que tr ba del algoritmo de reconocimiento de la siguiente manera:

Sea $T = \{(,)\} \cup \{x \mid x \text{ es un tipo sintáctico primitivo}\}$, es claro que T es un conjunto finito, por tanto podemos considerar el conjunto T^* de todas las cadenas sobre T y establecer la siguiente definición:

DEFINICION 4.1.- Diremos que un tipo sintáctico x es una PARTE de un tipo sintáctico y si, y solo si, (x) es una subcadena de y, considerando que $(x) \in T^*$ y $y \in T^*$.

EJEMPLO 4.1.- Considerando los tipos sintácticos primitivos del lenguaje de las expresiones algebraicas, tenemos que del tipo sintáctico

$$((p\backslash(c/p))/(c/e))$$

(sin contarlo a el mismo). son partes los tipos siguientes

$$p\backslash(c/p)\ ,\ c/p,\ c/e,\ p,\ c,\ e.$$

La anterior definición nos permite establecer algunas otras definiciones útiles para nuestros propósitos.

DEFINICION 4.2.- Sean x, z, y tipos sintácticos, tales que x, y son partes de z. Diremos que y ES z-MAYOR QUE x si, y solo si, tambien x es parte de y.

Denotamos "y es z-mayor que x" por "$z[y \geq x]$."

EJEMPLO 4.2.- Para los tipos sintácticos del ejem. 4.1 tenemos que,

$z[(p\backslash(c/p)) \geq (c/p)]$, $z[z \geq z]$
$z[(c/p) \geq p]$, $z[z \geq (c/e)]$
$z[(c/p) \geq c]$, $z[(c/e) \geq e]$ y $z[(c/e) \geq c]$,

donde

$z = ((p\backslash(c/p))/(c\backslash e))$.

DEFINICION 4.3.- Sean $x, z, y_1, y_2, \ldots, y_k$ tipos sintácticos tales que x es parte de z, $z[y_1 \geq x]$, $z[z \geq y_k]$, y $z[y_{i+1} \geq y_i]$ $i=1,\ldots,k-1$.

El ORDEN de x en z, denotado $O[x]z$, es el número máximo de veces que aparece en algún tipo de la secuencia finita, $y_1, y_2, \ldots, y_k, z$, el signo del producto ".".

EJEMPLO 4.3.-- En el ejem. 4.2 tenemos que

$O[p]((p\backslash(c/p))/(c\backslash e))=0$, en cambio para el tipo sintáctico

$$z'=((p/(ec))(e\backslash p))/e)c$$

tenemos

$$O[ec]z' = 2$$

DEFINICION 4.4.- Dada una fórmula $x \Rightarrow y$, diremos que es A LO MAS BINARIA si, y solo si, existe un tipo z, tal que es parte de x, es parte de y, y

$O[z]x - O[z]y =$ **0 ó 1.**

EJEMPLO 4.4.- Las fórmulas siguientes son a lo más binarias,

$$(x/y)y \Rightarrow x,$$
$$x(x\backslash y) \Rightarrow y$$
$$(x/y)(y/z) \Rightarrow x/z$$
$$x \Rightarrow y/(x\backslash y),$$

en cambio,

$$x \Rightarrow (xy)/y$$

no es a lo más binaria. Nótese que a partir de una fórmula que sea a lo más binaria, o que no sea a lo más binaria, podemos obtener una infinidad de fórmulas del mismo tipo utilizando el T1.6 que asegura:

$$\frac{x \Rightarrow x' \quad y \Rightarrow y'}{xy \Rightarrow x'y'}$$

Habiendo enunciado las definiciones necesarias pasamos a hacer explícitos los conceptos subyacentes en nuestra exposición del capítulo III.

DEFINICION 4.5.- Una GRAMATICA CATEGORIAL BIDIRECCIONAL es una quíntupla $\langle VT, \tau, \sigma, R, \varphi \rangle$, donde:
VT es un conjunto finito de elementos, se le conoce como el VOCABULARIO; en aplicaciones a la Linguística sus elementos generalmente son formas de palabras.

τ es la cerradura de un conjunto finito de tipos sintácticos primitivos o categorías fundamentales, digamos $t_1, t_2, \ldots, t_n$, bajo las operaciones $.,/,\backslash$.

σ es un elemento distinguido de τ; en nuestra aplicación a la Linguística era 0.

$R=\{(x/y)y \Rightarrow x, \quad x(x\backslash y) \Rightarrow y\} \cup B$ es el conjunto de reglas de cancelación, donde B es un conjunto finito de fórmulas demostrables a lo más binarias de Σ o de una extensión de Σ; puede pasar que $B = \emptyset$.

φ es la función de asignación desde VT en subconjuntos finitos de τ.

Restringiendo el conjunto τ se obtienen distintos tipos de gramáticas categoriales, de las cuales pueden verse algunas en el cap. III de [23] para $B=\emptyset$.

DEFINICION 4.6.- Decimos que una cadena de tipos sintácticos x SE CANCELA DIRECTAMENTE A y, si y resulta desde x por una aplicación de una regla de cancelación. Diremos que x SE CANCELA A y, si existe una secuencia finita de cadenas de tipos sintácticos, $z_1, z_2, \ldots, z_n$ tal que $x=z_1$, $y=z_n$ y z_i se cancela directamente a z_{i+1} para $i=1,\ldots,n-1$

En otras palabras, x se cancela a y, si y resulta de x por la aplicación finita de las reglas de cancelación.

DEFINICION 4.7.- Una cadena α sobre VT es llamada una ORACION si, y solo si, al menos una de las cadenas de tipos sintácticos asignadas por φ a α se cancela a σ.

DEFINICION 4.8.- El conjunto de todas las oraciones de una gramática categorial bidireccional dada es llamada el LENGUAJE DETERMINADO POR LA GRAMATICA CATEGORIAL BIDIRECCIONAL.

DEFINICION 4.9.- Llamaremos LENGUAJE CATEGORIAL BIDIRECCIONAL a un lenguaje dado si, y solo si, existe una gramática categorial bidireccional que lo determina.

Un ejemplo de gramática categorial bidireccional es obtenido fácilmente de los capítulos anteriores si tomamos como VT el conjunto de las formas de palabras del español a las que asignamos tipos sintácticos en los cap. I y cap. III, como categoría distinguida el tipo primitivo 0, y como reglas de cancelación las dadas en la sección B) del capítulo III. Haciéndolo obtenemos el siguiente ejemplo:

EJEMPLO 4.5.- $H = \langle VT, \tau, \sigma, R, \varphi \rangle$

dónde:

VT= {*Jacobo, María, Juan, pobre, fresca, padre, papá, apá, miel, leche, compra, trabaja, él, probablemente, ama, supone, suponen, corre, estudia, para, a, y*} .

τ es la cerradura del conjunto de tipos sintácticos $\{n, 0\}$ bajo operaciones $.,\backslash,/$.

La función φ desde VT en subconjuntos finitos de τ es

$\varphi(Jacobo) = \{n\}$, $\varphi(miel) = \{n\}$, $\varphi(él) = \{0/(n\backslash 0), (0/n)\backslash 0\}$

$\varphi(Juan) = \{n\}$, $\varphi(María) = \{n\}$, $\varphi(pobre) = \{n/n,\ n\backslash n\}$,

$\varphi(leche) = \{n\}$, $\varphi(fresca) = \{n\backslash n\}$, $\varphi(padre) = \{n\}$,

$\varphi(apá) = \{n\}$, $\varphi(papá) = \{n\}$, $\varphi(compra) = \{(n\backslash 0)/n\}$,

$\varphi(trabaja) = \{n\backslash 0\}$, $\varphi(probablemente) = \{0\backslash 0, (n\backslash 0)\backslash(n\backslash 0), (n\backslash 0)/(n\backslash 0)\}$,

$\varphi(ama) = \{(n\backslash 0)/n\}$, $\varphi(a) = \{n/n, ((n\backslash 0)/n)\backslash((n\backslash 0)/n)\}$

$\varphi(supone) = \{n\backslash 0\}$, $\varphi(suponen) = \{(0/(n\backslash 0))\backslash 0\}$,

$\varphi(corre) = \{n\backslash 0\}$, $\varphi(estudia) = \{n\backslash 0\}$, $\varphi(y) = \{(0\backslash 0)/0,\ 0\backslash(0/0)\}$,
$\varphi(para) = \{(0\backslash 0)/n,\ 0\backslash(0/n)\}$.

$\sigma = 0$

$R = \{(x/y)y \Rightarrow x,\ x(x\backslash y) \Rightarrow y,\ (x\backslash y)/z \Leftrightarrow x\backslash(y/z),$
$(z/y)(y/x) \Rightarrow z/x,\ (z\backslash y)(y\backslash x) \Rightarrow z\ x\}$.

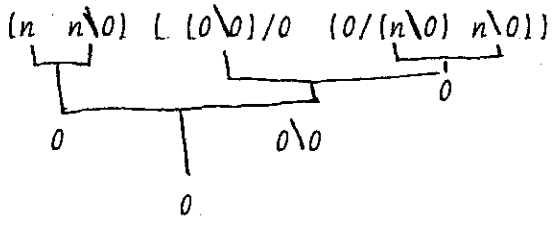
En esta gramática categorial son ORACIONES todas las empleadas hasta aquí como ejemplos; hay otras oraciones en esta gramática, por ejemplo, la cadena sobre VT

María estudia y él trabaja.

es una ORACION en nuestra gramática categorial bidireccional ya que una cadena de tipos sintácticos asignada por φ a la cadena se cancela a 0:

1) $(n\ \ n\backslash 0)\,[(0\backslash 0)/0\ \ (0/(n\backslash 0)\ \ n\backslash 0))$
2) $(n\ \ n\backslash 0)\,[(0\backslash 0)/0\ \ 0)$
3) $(n\ \ n\backslash 0)\ \ 0\backslash 0$
4) $0\ \ 0\backslash 0$
5) 0

con una secuencia de 5 cadenas de tipos sintácticos (ver def. 4.6), tales que z_i se cancela directamente a z_{i+1} para $i=1,\ldots,4$. En forma de "árbol" tenemos



Aunque en la sección A) del capítulo III al aplicar el algoritmo de reconocimiento decíamos que una expresión era "expresión algebraica bien formada" si tenía tipo c o tipo e, podemos ver que es posible dar una gramática categorial bidireccional para las expresiones algebraicas que puede manejar una microcomputadora si tomamos en cuenta, primero, que en tal caso los signos para las variables y el conjunto de números que se utilizan son finitos y, segundo, que agregando las reglas de cancelación $e \Rightarrow E$ y $c \Rightarrow E$, donde E es el tipo para "expresión algebraica bien formada", obtenemos la categoría distinguida $\sigma = E$. Nótese que las reglas de cancelación no son teoremas en Σ sino

que se obtiene de dicho sistema formal incorporando $e \Rightarrow E$ y $c \Rightarrow E$ como AXIOMAS y manteniendo igual los 3 esquemas de axiomas y las reglas de inferencia.

La familia de los lenguajes categoriales bidireccionales es cerrada bajo la unión. En efecto, si L_1 y L_2 son lenguajes categoriales bidireccionales, podemos encontrar gramáticas categoriales bidireccionales.

$$H_1 = \langle VT_1, \tau_1, \sigma_1, R_1, \varphi_1 \rangle,$$
$$H = \langle VT_2, \tau_2, \sigma_2, R_2, \varphi_2 \rangle,$$

tales que $\tau_1 \cap \tau_2 = \emptyset$ y, L_1 es determinado por H_1, L_2 es determinado por H_2.

Sea σ un elemento que no pertenece a $VT_1 \cup VT_2$, y construimos una gramática categorial bidireccional

$$H = \langle VT, \tau, \sigma, R, \varphi \rangle$$

con

$$VT = VT_1 \cup VT_2$$

$$\tau = \tau_1 \cup \tau_2$$

$$R = \{(x/y)y \Rightarrow x, x(x\backslash y) \Rightarrow y\} \cup B_1 \cup B_2 \{\sigma_1 \Rightarrow \sigma_1\ \sigma_2 \Rightarrow \sigma\}$$

φ es una función desde VT en subconjuntos finitos de τ, tal que, para $\alpha \in VT$

$$\varphi(\alpha) = \begin{cases} \varphi_1(\alpha) & \text{si } \alpha \in VT_1 - VT_2 \\ \varphi_2(\alpha) & \text{si } \alpha \in VT_2 - VT_1 \\ \varphi_1(\alpha) \cup \varphi_2(\alpha) & \text{si } \alpha \in VT_1 \cap VT_2 \end{cases}$$

entonces si L es el lenguaje determinado por H tenemos que $L = L_1 \cup L_2$.

Resulta que la anterior no es la única propiedad de los lenguajes categoriales bidireccionales ya que para todo lenguaje cateorial bidireccional es posible construir una gramática generativa libre de contexto que lo genera. Antes de demostrar el teorema correspondiente recordemos que una gramática libre de contexto es una cuádrupla $G = \langle V, VT, \sigma, P \rangle$, donde:

- V es un conjunto finito; llamado el vocabulario.
- VT es un subconjunto de V; llamado el vocabulario terminal y sus elementos símbolos terminales: en aplicaciones linguísticas a la sintaxis por lo general son formas de palabras.
- σ es un elemento distinguido de V-VT, llamado variable inicial.

P es un conjunto de reglas, llamadas producciones o reglas de reescritura de la forma $u \rightarrow v$ con $u \in V-VT$ y $v \in V^*$.

TEOREMA T4.1.- Para cualquier gramática categorial bidireccional $H=\langle VT,\sigma,R,\varphi\rangle$ es posible construir una gramática libre de contexto G, tal que L(G) es precisamente el lenguaje determinado por H.

Demostración:

Para construir $G=\langle V',VT',\sigma',P\rangle$ tenemos que $\sigma'=\sigma$, además $VT'=VT$, es decir, el elemento distinguido es el mismo en ambas gramáticas, y el vocabulario de H es igual al vocabulario terminal de G.

Para construir V', recordando que $\varphi(\alpha)$ es un subconjunto finito de $\tau, \forall \alpha$ del conjunto finito VT, tenemos que la unión D de todos los valores de $\varphi(\alpha)$ es un conjunto finito. Además el conjunto de las partes de $\varphi(\alpha)$ $\forall \alpha \in VT$ es finito y así la unión E de todas las partes de los valores de $\varphi(\alpha)$, para $\alpha \in VT$ es finita, $\therefore$ $F=D \cup E$ es finito y haciendo $V'=VT \cup F$ conseguimos el conjunto finito que necesitamos para vocabulario de G.

El conjunto P de las reglas de reescritura consistiría de reglas de 3 tipos:

1) del tipo $x \rightarrow (x/y)y$, si $(x/y) \in F$,
   y del tipo $y \rightarrow x(x\backslash y)$, si $(x\backslash y) \in F$.
2) del tipo $x \rightarrow \alpha$, si $x \in \varphi(\alpha), \forall \alpha \in VT$.
3) del tipo $x \rightarrow y$ si x y $y \in F$,
   y del tipo $x \rightarrow y.z$ si x,y y $z \in F$.

El conjunto P es finito ya que tanto F, VT y R lo son. Claramente L(G) es el lenguaje determinado por H.

Este es el momento oportuno para recordar que en el cap. III mencionamos que había dos razones: una práctica y la otra teórica, para pedir que las reglas de cancelación formaran un conjunto finito. La razón práctica ya fue señalada y la teórica es que si R, al igual que F y VT, no fuera un conjunto finito no podríamos obtener el conjunto finito de reglas de reescritura necesario

para poder construir nuestra gramática libre de contexto, en otras palabras, sin este requisito no tendríamos el teorema T4.1.

Nótese que las relgas del tipo del inciso 3) se presentan si en H el conjunto $B \neq \emptyset$, es decir, si alguna regla de cancelación a lo más binaria, distinta de $(x/y)y \Rightarrow x$, $x(x\backslash y) \Rightarrow y$, es utilizada. En efecto, en una fórmula a lo más binaria $x \Rightarrow y$ siempre es posible encontrar:

1) Dos partes z, w de x tales que $x = z.w$ y la regla de cancelación se convierta en $z.w \Rightarrow y$, si, para algún t parte de x y y, $0[t]x - 0[t]y = 1$. o en caso contrario,

2) si $0[t]x - 0[t]y = 0$, para t parte de x y y, es posible encontrar al menos una parte z de x, y al menos una parte w de y, tales que ni z y w estan compuestos a partir del producto y, digamos, $x = z/t$, $y = t\backslash w$, o algún tipo compuesto análogo, entonces $x \Rightarrow y$ se convierte en $z/t \Rightarrow t\backslash w$.

En caso de que $B = \emptyset$ entonces no se tendrían las reglas del tipo inciso 3). Para este caso se menciona en [14] que una gramática categorial bidireccional H es siempre encontrada para cualquier gramática G libre de contexto tal que L(G) es precisamente el lenguaje determinado por H.

Dado un lenguaje que podemos ver como generado por una gramática libre de contexto o como determinado por una gramática categorial bidireccional, el adoptar una de las gramáticas significa tomar partido por uno de los lados de la diferencia profunda existente entre ambos tipos de gramáticas.

Dentro de la Linguística la diferencia se presenta en el procedimiento de representar la combinalidad sintáctica de las formas de palabras. Por un lado en las gramáticas generativas libres de contexto la información está contenida en las reglas de reescritura. Por otro lado, en las gramáticas categoriales dicha información se encuentra en el vocabulario, es decir, en los tipos sintácticos de las formas de palabras. Las reglas en las gramáticas categoriales también son necesarias, pero hay relativamente pocas y, por otra parte, no contienen ninguna información sobre un lenguaje concreto; en realidad, ellas son teoremas de un sistema formal que en este trabajo hemos llamado CALCULO DE TIPOS SINTACTICOS.

Todos los conceptos que hemos examinado en este trabajo se han referido, de hecho, a un solo aspecto de los lenguajes: el sintáctico. Ahora bien, el lenguaje posee otro aspecto no menos importante que el sintáctico: el semántico. Sobre este último aspecto casi nada se ha dicho en los capítulos anteriores y, nos gustaría terminar nuestro trabajo con la presentación sucinta de algunas ideas que se manejan en el estudio de la Semántica. Estas ideas se desarrollan dentro de la clase de análisis semántico del lenguaje que ha sido seguido dentro de las llamadas GRAMATICAS DE MONTAGUE.

Suponemos que todo lenguaje natural sirve para dos cosas: nombrar entes y decir algo sobre ellos, además por lo general cuando queremos decir algo lo hacemos por medio de una oración declarativa. Sin embargo, si queremos dar una cuenta precisa del significado de una expresión linguística no es bastante relacionarla con un objeto o conjunto de objetos; debemos también proveer un SENTIDO o CONCEPTO para la expresión como muestran las dos lecturas de la oración

Juan piensa en su futura esposa,

1) en una primera lectura, Juan está pensando en una *persona*, es decir un ente.
2) En una segunda lectura tenemos que Juan estpa pensando, digamos a los 15 años, en las *características* que debe tener la persona que en a futuro será su esposa.

En una terminología que ha llegado a ser la más generalmente aceptada en los últimos años, nos referiremos en lo subsecuente, al sentido o concepto relacionado a una expresión como su INTENSION, y por la EXTENSION de la misma expresión queremos significar aquellos objetos en el mundo a los cuales se refiere la expresión.

Ahondando em estos conceptos tenemos que la extensión de una expresión linguística depende del MUNDO POSIBLE considerado, es decir, del contexto en que se sitúe la expresión. Por ejemplo, la expresión *rubio* tiene una extensión en el mundo posible de los Estados Unidos Americanos y otra extensión distinta en el mundo posible de México, ya que es común considerar como rubias a personas que en E.U.A. no lo son. Por otro lado, ¿qué tienen en común las varias extensiones de una expresión linguística, digamos *rubio* en diferentes mundos posibles?

¿qué es lo que hace que todas ellas sean extensiones, por ejem., de *rubio* y no de *moreno?*. La respuesta será "la intensión" ya que la consideramos como el "pegamento" que une el lenguaje con el mundo, es decir, una intensión es algo que relaciona una expresión linguística con su extensión. Ella determina la extensión de una expresión: si tenemos conocimiento de la intensión de una expresión tenemos, por lo tanto, una herramienta para escoger los objetos en el mundo que tiene como extensión la expresión linguística. En terminología matemática, podemos decir que la intensión es una función desde mundos posibles en extensiones. Volviendo a la primera pregunta planteada, podemos decir que lo que las varias extensiones de una expresión en mundos posibles diferentes tienen en común es que todas ellas son valores de una función intensional, la cual para todo mundo posible escoge exactamente la extensión relevante.

Consideramos que la extensión de una oración declarativa es su valor de verdad y que la proposición respectiva, la cual es la designación usual para la intensión de una oración declarativa, es una función desde mundos posibles a valores de verdad. Las intensiones de predicados y términos individuales son funciones desde mundos posibles en conjuntos de objetos y objetos, respectivamente. Señalemos que este análisis de la intensión ha hecho el concepto esencialmente independiente del lenguaje, puesto que una intensión tiene entidades extralinguísticas como su dominio (mundos posibles) y como su rango (objetos y valores de verdad). Para conectar las intensiones con el lenguaje, suponemos que existe una *función interpretadora general* I la cual para toda expresión linguística nos da la intensión apropiada. Para la expresión *rubio* tenemos

$$I(\textit{rubio}) = \text{intensión}_{\textit{rubio}}$$

$$\text{Int}_{\textit{rubio}}(W_n) = \text{Ext}_{W_n}(\textit{rubio}),$$

BIBLIOTECA DE CIENCIAS EXACTAS Y NATURALES
EL SABER DE MIS HIJOS HARA MI GRANDEZA

donde:

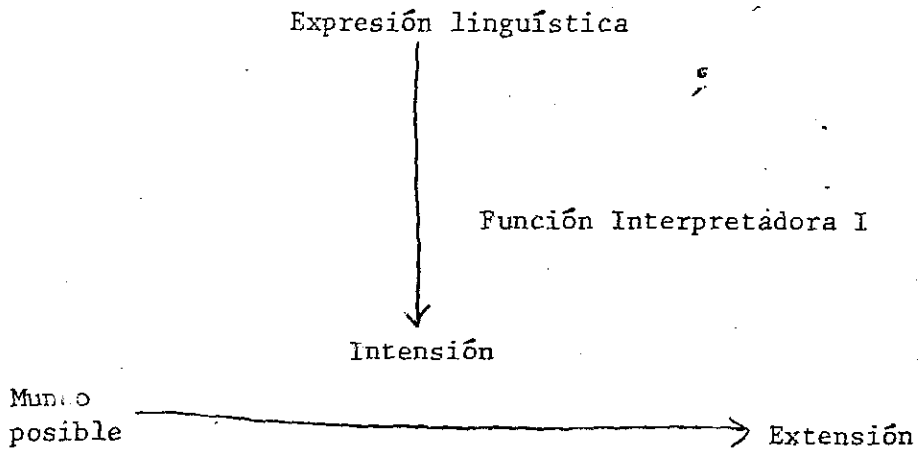
W_n = mundo posible con índice n.

I = función interpretadora general.

$\text{Int}_{\textit{rubio}}$ = la intensión para *rubio*

$\text{Ext}_{W_n}(\textit{rubio})$ = la extensión para *rubio* en W_n.

Así pues, la función interpretadora general escoge la intensión para una expresión linguística, dicha intensión es, a su vez, una función tal que nos da para cualquier mundo posible arbitrario W_n la extensión de la expresión en ese mundo. En forma de diagrama lo podemos representar como:

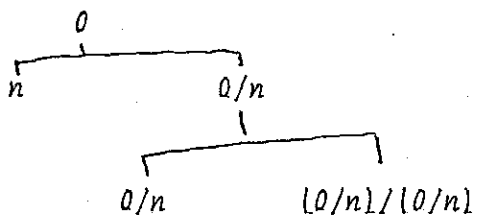


En el análisis semántico del lenguaje realizado dentro de las gramáticas de Montague se parte de dos ideas fundamentales: 1) el significado de una expresión linguística compleja es una función del significado de sus partes constituyentes, y 2) hay un isomorfismo entre la sintaxis y la semántica de un lenguaje, es decir, la [illegible] es una clase de mapa que muestra como los significados de las partes se [illegible]an en el significado del todo. Esto significa que ante dos análisis sintáctico preferimos uno de ellos por criterios semánticos, por ejemplo, al analizar la oración

Juan corre velozmente

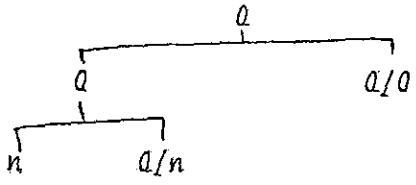
preferimos el análisis

Juan corre velozmente



al análisis

Juan corre velozmente



ya que *correr velozmente* tiene "sentido".

Observemos que los tipos sintácticos no son los que les asignaríamos, de acuerdo a nuestro análisis de los capítulos anteriores, a las formas de palabra que aparecen en la oración anterior. Lo que pasa es que simplemente se considera el hecho de que una forma de palabra se concatena con otra para obtener una expresión sin tomar en cuenta el "lado" de la concatenación: si se concatena por la derecha o por la izquierda es irrelevante. Así, en este análisis sintáctico *pobre* → n/n, y no los tipos $n\backslash n$ y n/n . La gramática categorial de este tipo es llamado *unidireccional* y este tipo de gramática es la isomorfa con la gramática esbozada a continuación.

Para llevar a cabo las ideas mencionadas antes partimos de *tipos semánticos primitivos* que corresponden a los tipos sintácticos básicos de los nombres y las oraciones declarativas. Las extensiones de las oraciones son valores de verdad, con tipo extensional t, y las extensiones de los nombres son entes, de tipo extensional e. Las intensiones de oraciones (proposiciones) son funciones desde mundos posibles a valores de verdad. Tales funciones son indicadas por el tipo intensional $\langle s,t\rangle$, donde s es una etiqueta para el tipo intensional correspondiente al tipo extensional t. En general, si α es un tipo extensional, $\langle s,\alpha\rangle$ es el tipo intensional de la intensión correspondiente a α. Las intensiones de nombres son funciones desde mundos posibles a entes y, así, son de tipo intensional $\langle s,e\rangle$. Las intensiones de los nombres son muchas veces llamadas *conceptos individuales*.

Antes de ver cómo construir intensiones y extensiones para otras expresiones, queremos señalar que para abarcar en el análisis semántico casos como el de la segunda lectura de la oración

Juan piensa en su futuro esposa

debemos considerar que en algunas ocasiones la intensión de una expresión lingüística está actuando como extensión. Para incluir tales casos, Montague (ver el artículo [7]) considera que en todos los contextos, los nombres y las oraciones tienen como extensiones conceptos individuales y proposiciones, respectivamente.

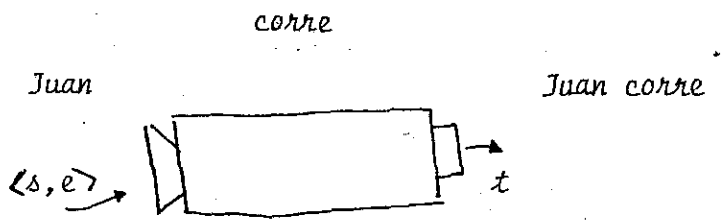
Hecho el anterior señalamiento, tenemos que análogamente al proceso seguido en el cap. I construiremos todas las intensiones y extensiones de todas las otras expresiones sobre la base de los nombres y oraciones. Todas las expre

siones de tipo semántico derivado tendrán como su extensión una función desde la *intensión* de un tipo a la *extensión* de otro, y como intensión una función desde mundos posibles en extensiones. Lo anterior es ilustrado en los siguientes ejemplos:

EJEMPLO 4.6.- Si tomamos *corre* su extensión será una función desde las intensiones de nombres a las extensiones de oraciones. En efecto, en la oración

Juan corre,

con extensión un valor de verdad de tipo semántico t, tenemos que *corre* se ha concatenado con *Juan*, que tiene una intesión (una función) de tipo semántico $\langle s,e\rangle$. Considerando la figura siguiente que ilustra como es obtenida la extensión de la oración



tenemos que la extensión de *corre* tiene tipo semántico extensional

$$\langle\langle s,e\rangle, t\rangle$$

y su intensión, (una función de mundos posibles en extensiones, a su vez funciones), tiene tipo semántico intensional

$$\langle s, \langle\langle s,e\rangle, t\rangle\rangle$$

EJEMPLO 4.7.- Del hecho que *velozmente* se concatena con *corre* produciendo una expresión linguística,

corre velozmente,

que actua como verbo intransitivo al concatenarse con un nombre, digamos *Juan*, tenemos que la extensión de *velozmente*, como es ilustrado en la siguiente figura,

es una función desde la intensión de un expresión con tipo intensional $\langle s, \langle\langle s,e\rangle,t\rangle\rangle$ a la extensión de una expresión con tipo semántico $\langle\langle s,e\rangle,t\rangle$. El tipo sempantico extensional de *velozmente* lo denotamos por

$$\langle s, \langle\langle s,e\rangle , t\rangle\rangle, \langle\langle s,e\rangle,t\rangle$$

y el correspondiente tipo semántico intensional para la intensión es denotado

$$\langle s, \langle\langle s, \langle\langle s,e\rangle, t\rangle\rangle , \langle\langle s,e\rangle, t\rangle\rangle\rangle$$

En general, tenemos que si una expresión linguística tiene como extensión una función desde la intensión de tipo intensional $\langle s,\alpha\rangle$ a la extensión de tipo extensional β, el tipo sempantico extensional que le corresponde será denotado por

$$\langle\langle s,\alpha\rangle,\beta\rangle$$

La intensión correspondiente será una función que a los mundos posibles les asocia extensiones de tipo semántico extensional $\langle\langle s,\alpha\rangle ,\beta\rangle$, así que le asignamos el tipo semántico intensional

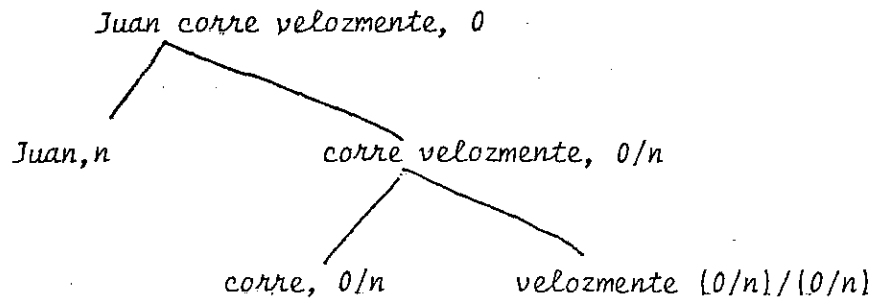
$$\langle s, \langle\langle s,\alpha\rangle,\beta\rangle\rangle$$

Antes de dar explícitamente el isomorfismo entre sintaxis y semántica, ilustramos el paralelismo que se presenta entre la estructura sintáctica y la estructura sempantica cuando una oración es analizada utilizando tipos sintácticos y tipos semánticos.

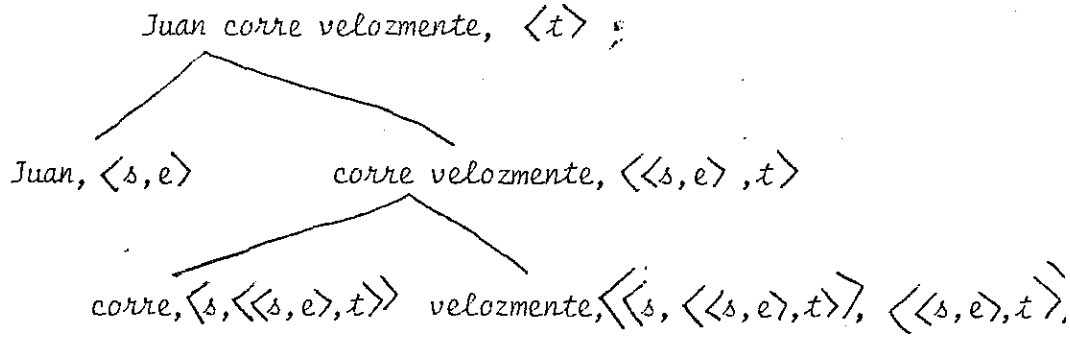
EJEMPLO 4.8.- Para la oración

Juan corre velozmente

tenemos el análisis sintáctico



que podemos comparar con el análisis semántico



para darnos cuenta del paralelismo entre las estructuras.

La función F que nos permite establecer el isomorfismo entre sintaxis y semántica es una que asocia tipos sintácticos de una gramática cateonal unidireccional con tipos semánticos de una gramática intensional. La función es definida recursivamente de la siguiente manera:

$$F(0) = t$$

$$F(n) = e$$

$$F(x/y) = \langle\langle s, F(y)\rangle, F(x)\rangle$$

Por ejemplo, al tipo sintáctico $(0/n)/n$ le hacemos corresponder el tipo semántico extensional que a continuación obtenemos por medio de la función F:

$$F((0/n)/n) = \langle\langle s, F(n)\rangle, F(0/n)\rangle$$

pero como

$$F(n) = e$$

$$F(0/n) = \langle\langle s, F(n)\rangle, F(0)\rangle$$

tenemos que

$$F((0/n)/n) = \langle\langle s, e\rangle, \langle\langle s, e\rangle, t\rangle\rangle$$

En lo anterior hemos hablado de una gramática categorial unidireccional que sería el caso especial de la definición de gramática categorial bidireccional donde T es la cerradura del conjunto finito $\{n, 0\}$ bajo las operaciones $.,/$. Las reglas de cancelación se convierten en

$$(x/y)y \Rightarrow x$$

$$x(y/x) \Rightarrow y$$

ya que de hecho las operaciones / y \ son una misma operación. Un desarrollo análogo para obtener una gramática intensional puede ser efectuado, donde las reglas de cancelación tendrían la forma:

$$\langle \delta, \beta \rangle \; \langle \langle \delta, \beta \rangle, \alpha \rangle \Rightarrow \alpha$$

$$\langle \langle \delta, \beta \rangle, \alpha \rangle \; \langle \delta, \beta \rangle \Rightarrow \alpha$$

para tipos semánticos α, β.

El desarrollo de tal gramática intensional se encuentra fuera de los propósitos de este trabajo. Para una presentación, cercana a la esbozada aquí, de las gramáticas intensionales puede consultarse [1] y para mayor profundidad en el estudio de ellas, veánse el artículo [7] y las notas de clase [2]. La pretensión de mencionar los conceptos anteriores en este trabajo, ha sido mostrar que los métodos y conceptos desarrollados para tratar los tipos sintácticos no se agotan en el campo de la sintaxis sino que, por el contrario, pueden extenderse al campo de los estudios semánticos.

BIBLIOGRAFIA:

1.- Allwod Jens, Lars-Gunnor Andersson, Osten Dahl. Logic in Linguistics. Cambridge University Press, 1977.

2.- Avila Murillo, Fernando, Notas de clase del curso "Lógica Formal II" impartido en SEM 83-1.

3.- Bur-Hillel, Y. "A Quasiarithmetical notation for syntactic description". Language. vol 29 (1453) pp. 47-58

4.- Birkhoff, Garret y Saunders Mac Shane. Algebra. Macmillan Publishing Co. Inc. New York, 1967.

5.- Calderón Enrique y Maria Isabel García. "La estructura de los lenguajes" Revista Matemática. Sociedad Matemática Mexicana. Número 7, 1970.

6.- Calderon Alzati, Enrique. La próxima generación de computadoras. Ciencia y Desarrollo #54, 1984.

7.- Cooper, Robin. Review article: formal filosophy of R. Montague. Language. Vol. 53, 1977.

8.- Chevalley, Claude, Fundamental concepts of algebra. Academic Press inc. 1956.

9.- Chomsky, Noam. Aspectos de la teoría de la sintaxis. Ed. Aguilar,Madrid 1970.

10.- Chomsky, Noam. Estructuras sintácticas. Siglo XXI ed., 1974.

11.- Chomsky, Noam. "On ceitan properties of grammars". Information and control. Vol. 2, (1959) p.p. 137-167.

12.- Chomsky Noam. Three Models for tje description of language. I.R.E, trans., Inform, theory IT2 (1956) p.p.113-124.

13.- Fowler Roger, PARA COMPRENDER EL LENGUAJE, una introducción a la Linguística. Ed. Nueva Imagen.

14.- Gladkij.A.V. y I.A. Melcuk. Introducción a la Linguística Matemática. Ed. Planeta. Barcelona, 1972.

15.- Gross, Maurice. Modelos matemáticos en linguística. Ed. Gredos, Madrid, 1972.

16.- Hilton, Peter y Wu, Yel-Chiang. A course in Modern Algebra. John Wiley & Sons. 1974.

17.- Lambek, "On the calculus of syntactic types". Structure of language and its mathematical aspects, vol. XII, symposia in applied mathematics. R. Jakobson ed, 1961, Amer. Math. Soc.

18.- Lambek "The mathematics of sentence structure, Amer. Math. Monthly", vol. 65 (1958 p.p. 154-169.

19.- Leroy Maurice. Las grandes corrientes de la Linguística, Fondo de Cultura Económica, México, 1974.

20.- Lehmann, winfred. ¿Qué es la linguística? -hojas mimeografiadas.

21.- Marcus, Solomón, Nicolau, Edmond y Stati Sorin. Introducción en la linguística matemática. Ed. Teide, Barcelona, 1978.

22.- Marcus, Solomón, Aspectos matemáticos de la linguística. Linguística y Comunicación. Greimas et.al. Ed. Nueva Visión, Buenos Aires, 1967.

23.- Marcus, Solomón. Algebraic linguistics, Analytical Models.

24.- Mounin Georges, Historia de la Linguística, desde los orígenes al siglo XX. Ed. Gredos. Biblioteca Románica Hispánica. Madrid 1968.

25.- Malmberg, Bertil, Los nuevos caminos de la Linguística. Siglo XX ed. 12º ed., 1980.

26.- Nelson, R.I. Introduction to Automata. John Wiley & Sons Inc., New York, 1968.

27.- Quezada, J. Daniel. La Linguística generativo transformacional: supuestos e implicaciones. Alianza Editorial, S.A., Madrid, 1974.

28.- Serrano, Sebastian, Lógica, Linguística y matemáticas. Ed. Anagrama, Barcelona, 1977.

29.- Serrano, Sebastian. Elementos de linguística matemática. Ed. Anagrama, Barcelona, 1979.

30.- Stati, Sorin, La sintaxis. Ed. Nueva Imagen, México, 1979.

31.- Tarski, Alfred. Introducción a la lógica y a la metodología de las ciencias deductivas. Espasa-Calpe, S.A., 1977.

32.- Material de la sección de Matemática Educativa del CINVESTAV del IPN para la materia de "Teorías del Aprendizaje y la Instrucción".

33.- Kuhun, La Estructura de las Revoluciones Científicas, F.C.E.

34.- Spanter.E. Grammars and Language. Amer. Math. Monthly, Abril 1969, p.p. 335-342.