

LA UTILIZACION DE FORMALISMOS BASADOS EN UNIFICACION PARA EL ANALISIS DE LAS LENGUAS NATURALES(*)

Celso Vargas

ABSTRACT

Insights from Artificial Intelligence and Computer Science are very fruitful in the development of formalisms for natural language analysis and comprehension. In this paper I present some general aspects of a theoretical perspective widely used in the description of natural languages. This approach is based on computational resources, specially those that derive from logic programming, in which the Unification algorithm plays an important role. In the first section I describe the origins of unification. In the second section I give a mathematical perspective that underlies formalisms for natural languages analysis. In the third and fourth sections, I present two different approaches that utilize unification as the only operation, namely, categorial grammars and phrase structure grammars.

0. Introducción

Para muchos autores (véase Karttunen (1986), Barlow (1988)), la construcción de gramáticas y formalismos para el análisis de las lenguas naturales basados en el procesamiento de información, constituye algo así como una segunda revolución en lingüística. Esto por varias razones, entre ellas las siguientes: a- el concepto de información es mucho más factible de ser formalizado. Y es, en un sentido importante, independiente de la estructura de la lengua, esto es, permite generalizaciones no expresables en otros formalismos; b- a diferencia de enfoques tradicionales, como el transformacional, están totalmente basados en la estructura superficial, esto es, en el orden de aparición de los constituyentes en oración; c- utiliza los mismos instrumentos computacionales utilizados en el diseño y construcción de lenguajes de programación, fundamentalmente programación lógica. Esto constituye una importante generalización; d- la codificación de información de las entradas léxicas desempeña un papel muy

importante. En efecto, como señala Barlow (1988) a diferencia de los enfoques tradicionales, la estructura sintáctica proporciona menor información que las entradas léxicas. Esto posibilita la codificación de información tanto sintáctica, semántica como fonológica, sin necesidad de considerar estos aspectos como componentes separados; e- consecuentemente, se observa un cambio significativo "de la representación estructural a la representación relacional de la información" (Barlow 1988: 2). Esto es, lo más importante no es codificar las relaciones sintácticas por sí mismas, sino las restricciones que la transmisión de información conlleva; f- dado que la noción de "información" es neutral, puede ser compatible con enfoques tanto conceptualistas como realistas. Este último aspecto es, desde el punto de vista filosófico, bastante significativo, ya que, como señalan Sag y Pollard (1987), pueden desarrollarse diferentes gramáticas y formalismos sin tener que esperar a que los problemas filosóficos de fondo, relativos a la naturaleza última de los formalismos, sean resueltos.

Los formalismos basados en la transmisión de información, por lo menos del tipo que consideramos aquí, son bastante recientes. Su introducción en lingüística se remonta a 1979, fecha en que Martin Kay introduce, por primera vez, operación de unificación, en el contexto de su "Functional Grammar".

Ahora bien, a diferencia de muchos de los enfoques existentes, los algoritmos y gramáticas basadas en información pueden establecer las condiciones bajo las cuales tipos de información pueden ser transmitidos, combinados, cuando hay información incompatible, etc., sin que haya una explosión en el poder computacional de los modelos, o sin que tengamos que abandonar la base matemática de las gramáticas libres de contexto. De hecho uno de los resultados más importantes sobre los formalismos basados en información es que sobre una base libre de contexto es posible generar clases de gramáticas que recorran toda la jerarquía de Chomsky (véase Shieber (1986)). Esto tiene gran importancia desde el punto de vista lingüístico, ya que con el propósito de tratar lenguas naturales es importante no limitar el formalismo por requerimientos computacionales.

En los enfoques basados en la transmisión de información, el algoritmo de unificación desempeña un papel fundamental. En este artículo quiero ocuparme de la estructura general de los formalismos que utilizan unificación como mecanismo fundamental. No es mi propósito evaluar las teorías en cuestión, sino más bien, presentar un enfoque que está tomando mucha importancia en este momento.

Como se verá, existen diferentes formas de especificar estos formalismos. En este artículo me propongo caracterizar dos de estas formas de especificarlos: 1-aquellos basados en reglas de reescritura y 2- los basadas en gramáticas categoriales. Ambos enfoques como indicaremos más adelante, aunque son equivalentes, al menos eso se cree, difieren en aspectos significativos.

En la primera sección señalo el origen de unificación. En la segunda, caracterizo la estructura matemática que subyace al enfoque basado en la transmisión de información que utiliza la teoría de grafos como estructura de representación. En la tercera, la incorporación de dominios informacionales en gramáticas de estructura sintagmática. En la cuarta su incorporación en el contexto de las gramáticas categoriales.

1. Origen de unificación

El algoritmo de unificación fue introducido por Robinson (1965) en el contexto de su método de resolución. El problema planteado por Robinson consistía en establecer un procedimiento mecánico (computacional) para determinar cuando de un conjunto de fórmulas de la lógica de predicados de primer o segundo orden, podemos inferir otra fórmula o fórmulas como consecuencia.

Como se recordará, la lógica de predicados de primer orden se construye con los siguientes elementos:

- 1- Un conjunto de constantes individuales, Pedro, Juan, roca, etc.
- 2- Un conjunto variables que pueden ser reemplazadas por constantes.
- 3- Un conjunto de funciones, '+', '-', 'sucesor', etc.
- 4- Un conjunto de predicados, amar a, querer a, vestir a, etc.
- 5- Un conjunto de conectivas lógicas, 'y', 'o', 'si,...., entonces,....', y 'no es el caso que'
- 6- Dos cuantificadores: uno universal y otro existencial.

A partir de estos elementos podemos construir fórmulas de diferente complejidad. Las fórmulas simples, aquellas que resultan de la combinación de un predicado y una constante o variable (o funciones), se designan fórmulas atómicas. El tipo de lógicas así construidas se llaman de predicados de primer orden. Las de segundo orden permiten cuantificar sobre predicados.

Sin embargo, las lógicas de predicados plantean un problema muy grave: son indecidibles. Esto es, no existe un procedimiento mecánico para determinar si una fórmula arbitraria de esa lógica tiene cierta propiedad (generalmente, si es una consecuencia de un sistema particular). Sin embargo, algunos subconjuntos de esta lógica son decidibles.

Sin embargo, a pesar de este resultado general, sí existen procedimientos de decisión parcial; el de Robinson es uno de ellos. Se llama método de resolución porque dado un conjunto de fórmulas se intenta encontrar el conjunto de resolventes. Un resolvente es de la A y -A (la afirmación y su negación). Se dice que un conjunto de fórmulas o una fórmula es insoluble (sin solución) si el conjunto vacío puede ser derivado mediante resolución, esto es, si para cada fórmula del conjunto existe su negación.

El método de resolución funciona sobre un subconjunto de fórmulas de la lógica de predicados, esto es, aquellas que tienen forma clausal. Se dice que una expresión está en forma clausal si reúne las siguientes condiciones:

1- Contiene únicamente los operadores lógicos de conjunción, disyunción y negación.

2- Todos los cuantificadores universales aparecen a la izquierda de la expresión eliminando todos los cuantificadores redundantes.

3- No posee cuantificadores existenciales. Estos son eliminados en favor de las funciones Skolem.

4- La fórmula en su totalidad consta de conjunciones con disyunciones internas y negaciones de fórmulas atómicas.

Uno de los teoremas de esta lógica o lógicas, debido a Skolem, establece que para cada fórmula de primer o segundo orden que es satisficible, existe otra de segundo orden que es equivalente a la primera. (Sin embargo, en la práctica la fórmula resultante, con frecuencia, es de mayor complejidad).

El método de resolución es un procedimiento de decisión parcial y se aplica únicamente a fórmulas en forma clausal. Dada una fórmula en forma clausal B, el método de resolución toma negación de B y determina si es insatisficible, en cuyo caso, A es decidible. A es insatisficible si y solo si, la fórmula vacía puede ser derivada. Esto significa que todas sus fórmulas atómicas son de la forma A y -A.

En la aplicación del método de la resolución se hace uso de dos importantes elementos: la interpretación Herbrand y la operación de unificación. Quisiéramos referirnos brevemente a cada uno de ellos antes de ver la utilización de la operación unificación en las teorías lingüísticas.

El lenguaje de predicados de primer y segundo orden se construye a partir de los siguientes elementos: a- un conjunto de constantes, de dos tipos: individuales y constantes de predicado de adicidad n; b- un conjunto de funciones de adicidad n; c- un conjunto de variables individuales y de predicado; d- un conjunto de conectivas diádicas y monádicas y e- un conjunto de cuantificadores (universal y existencial). Se define el conjunto de términos de esta lógica como el conjunto que contiene todas las constantes, todas las variables individuales y el conjunto de funciones con sus respectivos argumentos.

La idea de la interpretación Herbrand es la siguiente: toda fórmula de lógica de predicados puede ser interpretada en términos de sí misma, es decir, reemplazando las variables por constantes (un conjunto finito de instanciaciones). Una fórmula no se satisface si, y solo si, es falsa en todas las instanciaciones.

Uno puede ver el conjunto de términos como el conjunto base de un monoide libre; las instanciaciones, selección de subconjuntos del monoide. Cada fórmula puede admitir un conjunto infinito de instanciaciones. En programación lógica el conjunto de instanciaciones está restringido por las bases de datos del programa.

En general, el algoritmo de la unificación es un caso especial de sustitución. De todo el conjunto de instanciaciones o sustituciones que yo puedo realizar, son de interés aquellas que me permitan aplicar el método de resolución, es decir, que me permitan obtener fórmulas del tipo A y -A (recuérdese que el objetivo del método de resolución es derivar la cláusula nula cuando una fórmula sea insatisficible). Dos términos t_1 y t_2 unifican si, y solo si, existe una sustitución s, tal que $s(t_1) = s(t_2)$. Así formulado, sin embargo, no me asegura que la unificación para los dos términos sea la más general. Podemos definir al **unificador más general** de dos términos del siguiente modo:

Un unificador u de los términos s y t es llamado **unificador más general** (UMG) de s y t si para cualquier otro unificador m, existe una sustitución h tal que $hu = m$. (Knight 1989 : 95)

Al procediendo para encontrar el UMG se denomina algoritmo de la unificación. En relación con este algoritmo señala Robinson (1979):

Este algoritmo, que es el corazón del principio de resolución, es de considerable interés por sí mismo. Opera sobre cualquier conjunto S de expresiones y tiene dos maneras de terminar (y siempre termina): o termina con la indicación de que el conjunto S no es unificable; o termina indicando que es unificable y señalando el unificador más general de S (Robinson 1979: 189).

El algoritmo de unificación que introduce Robinson contiene dos partes: un chequeador para evitar infinitas unificaciones y el programa que busca el unificador más general, si existe. Sin embargo, tiene varios problemas, uno de ellos es que requiere tiempo exponencial para decidir si un conjunto de expresiones es unificable o no.

Muchísima investigación se ha realizado, y se sigue realizando, para optimizar dicho algoritmo y extenderlo a diversas áreas de investigación. El artículo de Knight, recién citado, incluye consideraciones sobre unificación en las siguientes áreas: complejidad computacional, algoritmos y estructuras de datos, prueba de teoremas, programación lógica, lógicas de orden superior, estructuras de rasgos, procesamiento de lenguas naturales, algoritmos de unificación en paralelo, teorías de ecuaciones, inferencia tipo, lenguajes de programación (en particular en la familia de lenguajes de programación lógica) y máquinas de aprendizaje.

2. Estructura del formalismo para lingüística

Quisiera presentar la estructura matemática de los formalismos basados en la transmisión de información de más amplio uso en lingüística. En éstos podemos distinguir dos componentes: a- información sobre ítems léxicos y b- el conjunto de reglas que establecen las condiciones bajo las cuales conjuntos de información sobre ítems léxicos, sobre constituyentes, etc., pueden ser combinados. Respecto al primer componente, estos formalismos utilizan como recursos de representación la teoría de grafos aplicada a un dominio de información (informacional) determinado, en este caso, información lingüística. Sobre estos dominios informacionales opera unificación. Los dominios informacionales contienen toda aquella información lingüística relevante (sintáctica, semántica, pragmática y fonológica) sobre los ítems léxicos. Respecto al segundo componente, existen, como se verá más adelante, diferentes formas de establecer el conjunto de reglas.

El dominio informacional está constituido por dos elementos fundamentalmente: a- rasgos y b- valores. Por ejemplo, género, número y persona constituyen rasgos, femenino, singular y tercera persona sus valores respectivamente. Cada uno de los rasgos tiene asociado un conjunto de valores. El ejemplo anterior puede ser expresado de cualquiera de las dos maneras siguientes:

- (1) a. género: femenino b. < género, femenino >
 número: singular < número, singular >
 persona: tercera < persona, tercera >

Cada uno de estos rasgos, con su respectivo valor, se denomina estructura. Un estructura, entonces, consta de un rasgo y de un conjunto de valores (que puede ser vacío, []). Podemos definir formalmente lo dicho hasta el momento de la siguiente manera:

Sea A un conjunto de rasgos (R) y valores (V) sobre un dominio lingüístico particular. A partir de A podemos definir el conjunto de estructuras atómicas E' de la siguiente manera:

$$E' = \{ d_i : R \rightarrow V : i \in N \}$$

donde N es el conjunto de los números naturales. La función d_i es una función que toma cualquier argumento como entrada, no importa si está bien definida o no (esto es, si está lingüísticamente justificado o no), y realiza una operación sobre ella. Dicha operación consiste, fundamentalmente, en formar ecuaciones cuyos componentes son rasgos valores para formar estructuras.

Ahora bien, una estructura puede tener como argumento un conjunto de estructuras (sean atómicas o no), esto es, un rasgo puede tomar como valor un conjunto de rasgos y valores (otras estructuras). Por ejemplo, una lengua como el español en la que el verbo debe concordar con el sujeto en los dos rasgos, número y persona, puede representarse de la siguiente manera:

$$(2) \quad SN \left[\begin{array}{l} \text{Concordancia: } \boxed{1} \left[\begin{array}{l} \text{número: singular} \\ \text{persona: tercera} \end{array} \right] \end{array} \right]$$

$$V \left[\begin{array}{l} \text{Concordancia: } \boxed{1} \end{array} \right]$$

Los índices en este ejemplo indican la co-referencialidad de los rasgos de concordancia. (2) puede expresarse también como un conjunto de ecuaciones de la siguiente forma:

(3). concordancia (número) = singular y concordancia (persona) = tercera.

Antes de formalizar este hecho es importante definir lo que entendemos por Estructura. Esta puede definirse recursivamente del siguiente modo:

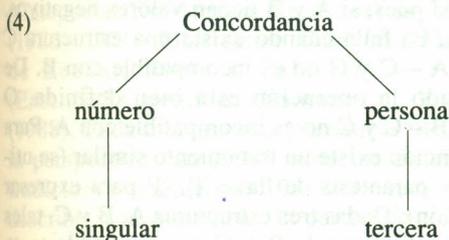
$$E = \{ d_i : R \rightarrow E ; i \in N \}$$

Se dice que el conjunto $\langle E, d_i \rangle$ en N genera un álgebra libre a partir de los elementos (estructuras) de A .

Ahora bien, el conjunto de valores que el conjunto R puede tomar es definido de la siguiente manera:

$$R_i : R \rightarrow E, i \text{ en } N.$$

Cada una de estas estructuras define un conjunto de niveles (L). El nivel más alto de la estructura está constituido por el rasgo y los niveles inferiores por los valores de ese rasgo (conjunto de estructuras). Dichas estructuras son ubicadas, también, en diferentes niveles. En cada una de las estructuras pueden introducirse nuevos valores. Por ejemplo, parte del ejemplo 2 puede ser representado del siguiente modo:



Para cada nivel de una estructura como 4, tiene un conjunto único de ecuaciones asociado y que constituye su descripción. Dicho conjunto de ecuaciones determina el conjunto de valores que ese nivel puede tomar. Por otro lado, dada la naturaleza composicional de las estructuras, es posible describir cada uno de los nodos de ese nivel. En efecto, el valor n , de un conjunto $\{l_1, l_2, l_3, \dots, l_n\}$ de niveles, denotado como $n(l_1, l_2, l_3, \dots, l_n)$ es equivalente a $n(l_1), n(l_2), n(l_3), \dots, n(l_n)$.

Es importante llamar la atención sobre el modo de representación que hemos estado utilizando. Este elimina la distinción entre argumento y función. En efecto, desde el punto de vista tradicional (de la lógica de predicados de primer orden) se hace necesario, tanto desde el punto de vista sintáctico como semántico, distinguir entre funciones y argumentos. Los argumentos denotan valores constantes, mientras que las funciones designan operaciones sobre argumentos. En los formalismos basados en unificación, los argumentos de una función pueden ser otras funciones (estructuras), con lo cual la distinción deja de ser válida.

Dicho lo anterior, podemos precisar un poco mejor el concepto de dominio informacional I : I está constituido por el conjunto de funciones,

$$d_i : E \rightarrow (E + L) + C$$

donde C es un conjunto de constantes, esto es, constantes que pueden ser introducidas en algunos niveles y L es un conjunto de niveles.

Ahora bien, sobre I pueden definirse, siguiendo a Scott (1981) y Shieber (1986), como el conjunto de sistemas de información $D = \langle D, D, \text{Con}, \rightarrow \rangle$

donde D es un conjunto de proposiciones, Con es el conjunto finito de subconjuntos consistentes, \rightarrow es la relación de implicación entre los elementos de Con y elementos de D , \perp es el elemento neutro.

Decimos que un subconjunto S de D es deductivamente cerrado si cada proposición implicada por un subconjunto consistente de S está en S . La cerradura deductiva C de S , $D C$ es el subconjunto de D más pequeño deductivamente cerrado que contiene a S (Pereira y Shieber 1986: 44).

Para completar el modelo matemático que subyace al tipo de formalismos que estamos considerando, es necesario que introduzcamos algunas definiciones.

1- Definición de subsunción:

Sean A y B dos estructuras cualesquiera, decimos que A subsume B (denotado como $A \supseteq B$) si, y solo si, toda estructura de A está incluida en B .

2- Definición de substitución.

Una substitución s es una operación sobre estructuras, por ejemplo, A, B, A' y B' tal que

$$\begin{aligned} a- & \text{ si } A' = A \text{ y } B' = B, \\ b- & s(A') = s(B'), \end{aligned}$$

entonces, $A = B$, donde B se obtiene a partir de A reemplazando A' por B' .

La operación substitución describe un conjunto de ecuaciones sobre estructuras. Substitución tiene un conjunto de propiedades.

Sean s y r dos substituciones. Entonces,

- 1- Reflexividad. Para toda substitución s , y estructura X , $s(X, X) = X$.
- 2- Simetría. Sea X e Y dos estructuras cualquiera si $s(X) = s(Y)$, entonces $s(Y) = s(X)$.
- 3- Transitividad. Sean X , Y y Z estructuras, si $s(X) = s(Y)$ y $s(Y) = s(Z)$, entonces, $s(X) = s(Z)$.
- 4- Asociatividad. Para toda estructura X , $r(s(X)) = s(r(X))$.

En los formalismos basados en Unificación, existe una clase particular de substitución que es de especial importancia: aquella en la que es posible obtener conjuntos mayores de información. La operación unificación permite definir esta clase.

Dadas dos estructuras A y B , decimos que A unifica a B , denotado como $\text{Unif}(A, B)$ si, y solo si, existen substituciones s y r tal que si $s(A)$ y $r(B)$, entonces, existe un C tal que $s(A) = C$ y $r(B) = C$.

Se dice que s es el unificador más general de dos estructuras A y B si, y solo si, para toda substitución s' (A), si $s'(A) = B$, entonces, $s'(A) = s(A)$.

La operación inversa a unificación es generalización. Se dice que dos estructuras A y B son generalizables, si y solo si para toda substitución s y r , $s(A) = r(B)$. En términos de conjuntos acotados, unificación corresponde al Sup del conjunto y generalización al Inf .

Ahora bien, en el análisis de las lenguas naturales es muy importante expresar generalizaciones sobre algunos fenómenos lingüísticos. Por ejemplo, en español existe una regla muy simple que rige los cambios de 'z' a 'c'. Esta regla afirma que se cambia la 'z' por la 'c' cuando le sigue una 'e' o una 'i'. En este caso la introducción de disyunciones es muy importante. Por otro lado, existe también una regla que rige la aplicación del imperativo. En general, parece que los imperativos no se aplican a la primera persona singular o plural, pero sí al resto de las personas (?). Es suficiente con indicar que no se aplica en estos casos. Aquí el uso de '+' o '-' para indicar la presencia o ausencia del rasgo en cuestión es importante.

No existen valores que sean intrínsecamente negativos, sino que lo que llamamos valores nega-

tivos remite a restricciones o a generalizaciones sobre el tipo de fenómenos a tratar.

Dado que la lógica de predicados de primer orden es indecidible, no puede proporcionarse un tratamiento completo del fenómeno de la negación y de la disyunción. Sin embargo, sí existe un procedimiento de decisión parcial para ellos.

Tanto las restricciones como las generalizaciones son expresadas en el tratamiento de Karttunen mediante una operación denominada negación. Dadas dos estructuras A y B .

Si A y B son distintas, esto es, contienen un valor diferente para algún rasgo, entonces ($\text{negar}(A, B)$) no hace nada sobre ellas. Si A y B no son distintas, entonces, A es marcada con $-B$ y B con $-A$. Estas restricciones evitan que los dos nodos lleguen a ser semejantes. Cuando A es unificado con C , la unificación tiene éxito sólo si el resultado es diferente de B (Karttunen 1986: 27)

Así pues, si A y B tienen valores negativos, $\text{Unif}(A, B)$ falla cuando existe una estructura C tal que $A = C$ y C no es incompatible con B . De otro modo la operación está bien definida. O cuando $B = C$ y C no es incompatible con A . Para la disyunción existe un tratamiento similar (se utilizan los paréntesis de llave '{', '}' para expresar disyunción): Dadas tres estructuras A , B y C , tales que A es disyunta de B y C es compatible tanto con A como con B , $\text{Unif}(\{A, B\}, C)$ (unificar a A o B con C) si, y solo si, existe otra estructura C' , tal que $C' = \{(A, C), (B, C)\}$. (Véase Karttunen 1986 para un tratamiento más completo de estos aspectos).

Estos aspectos nos permiten caracterizar con mayor precisión la definición de los sistemas de información introducidos anteriormente. En efecto, dado un conjunto de estructuras E , un sistema de información sobre E , se obtiene mediante las operaciones generalización y unificación. Esto es, un sistema de información es un conjunto de ecuaciones obtenidas mediante substitución y tal que, este conjunto de substituciones cumple con las condiciones de unificación y generalización. Expresado en términos de Pereira y Shieber:

Las ecuaciones expresan restricciones entre rasgos y valores en una estructura de rasgos, y la implicación codifica el carácter reflexivo, simétrico, transitivo y de substitución sobre ecuaciones. Más precisamente, decimos que un conjunto finito de ecuaciones E implica una ecuación e si

- Pertenencia: $e \in E$.
- reflexividad: e es D o e es $d = d$ para algún descriptor (²)
- Simetría: e es $d_1 = d_2$ y $d_2 = d_1$ está en E .

- Transitividad: e es $d_1 = d_2$ y existe un descriptor d tal que $d_1 = d$ y $d = d_2$ están en E.
 - Sustitución: e es $d_1 = p_1 \cdot d_2$ y ambos $p_1 = p_2$ y $d_1 = p_2 \cdot d_2$ están en E.
 - Iteración: Existe un E' c E tal que E' l- e y para todo e' e E', E l- e'.
 - Falsedad: Si e es inconsistente, {e} implica todo elemento de D.
- (Pereira y Shieber 1986: 44- 45).

Me interesa ahora ilustrar el uso de este formalismo en dos teorías lingüísticas contemporáneas: Gramáticas Catoriales y la Gramática de Estructura Sintagmática "Head-Driven".

3. Gramáticas catoriales

Diferentes tipos de gramáticas catoriales pueden ser desarrollados dentro de la perspectiva del enfoque informacional. En particular, CUG (Categorial Unification Grammar) de Uszkoreit (1986), UCG (Unification Categorial Grammar) de Zeevat, Klein y Calder (1986) y ECG (Extended Categorial Grammar) de Yoo y Lee (1988). Cada una de estas gramáticas incorpora, a la perspectiva categorial, aspectos de diferentes tradiciones lingüísticas. En esta sección quiero describir, como ejemplo, la perspectiva de Yoo y Lee (1988).

Las gramáticas catoriales tienen su origen en Ajdukiewicz (1960) "Lenguaje y Conocimiento". Fueron desarrolladas y modificadas por Lewis (1970) "General Semantics" y por Montague en diferentes artículos (véase la compilación de Thomason (1974)). La perspectiva de Montague fue la conocida y de mayor riqueza (véase Vargas (1989) para una descripción del enfoque; véase Josew (1986) para una formulación de la Semántica de Montague en términos de la lógica de predicados de primer orden).

Entre las características de las gramáticas catoriales, que surgen en el contexto de la Semántica de Montague, podemos señalar las siguientes: 1- Su punto de partida, a diferencia del enfoque transformacional, está basado enteramente en la estructura superficial, esto es, en el orden de aparición de los constituyentes de una oración. 2- Incorpora un conjunto de operaciones estructurales que cumplen el mismo papel que las transformaciones en la gramática de Chomsky. Pero a diferencia de Chomsky, no existe tal cosa como una estructura profunda. 3- Es de naturaleza fundamentalmente composicional. Esto es, una

expresión compleja es una función de los elementos constituyentes y de las reglas utilizadas en su construcción. 4- La especificación de la semántica se hace en términos de una lógica intensional cuyos valores son establecidos en una semántica de teoría de modelos (aunque existen otras alternativas).

Las tres primeras características están presentes en las gramáticas catoriales más recientes, pero incorporadas desde otra perspectiva. La última característica es reemplazada normalmente en favor de un tratamiento basado en la Semántica de las Situaciones de Perry y Barwise.

La ECG de Yoo y Lee incorpora dos elementos que desempeñan un papel muy importante en su propuesta, y que derivan de GPSG: el formato ID/LP (Immediate Dominance/ Linear Precedence) y el rasgo u operador '/' Slash.

Las reglas de una gramática, como vimos más arriba, son de la forma:

$$(5) X \text{ ----} \rightarrow Y + Z$$

En esta regla el orden de aparición de los constituyentes es muy importante. En efecto, esta regla indica que X domina inmediatamente a Y y a Z y que Y y Z aparecen en ese orden. Sin embargo, para lenguas con orden de aparición de las palabras más libre, este formato requiere la incorporación de una gran cantidad de reglas. En el formato de ID/LP las reglas pueden simplificarse de la siguiente forma:

$$(6) X \text{ ----} \rightarrow Y, Z$$

donde $Y < Z$ (Y precede a Z).

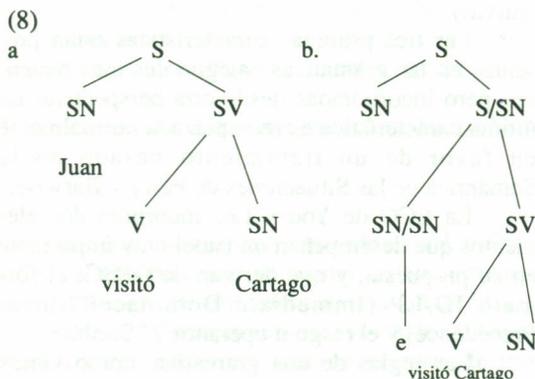
La precedencia puede establecerse mediante un conjunto de principios independientes al tipo de reglas utilizadas. ECG no incorpora ID, pero sí LP, como veremos más adelante.

El rasgo '/' fue introducido para dar cuenta de oraciones con QU (Quién, Cuándo, Dónde, Qué, etc.), y para casos de topicalización, en los que hay un constituyente (normalmente un Sintagma Nominal) que es desplazado.

En GPSG existe un conjunto base de categorías, por ejemplo, SN, SV, PP, y un conjunto expandido de categorías, SN/SV, SV/SV, SN/SN, etc. y que se lee "X menos Y". Estas son introducidas con el propósito de dar cuenta de dependencias no ligadas. Como en la oración b

del siguiente ejemplo, en el que existe una dependencia entre "Juan" y "quién".

- (7) a-Juan visitó Cartago
b-Quién visitó Cartago.



Como puede observarse en b hay un constituyente que tiene un vacío (e) y que debe ser llenado en alguna posición del árbol. Las categorías que dominan sobre vacío deben tener la forma X/X. Una de las características de Slash es que siempre indica en cuál posición éste es llenado.

Estos dos aspectos son incorporados en ECG. En este sentido, ECG puede ser considerada, al igual que HPSG, una extensión o reformulación de GPSG (3).

Para la caracterización de ECG es necesario tener en cuenta tres aspectos: 1- el conjunto de rasgos y categorías utilizadas; 2- las operación de cancelación y de concatenación y 3- los principios de precedencia lineal (PL).

Con respecto al primer aspecto, ECG utiliza tres categorías básicas: SN, S y PP, cada una de ellas posee un núcleo. El núcleo del SN es n (nombre), el de S es s (oración) y el de PP es p (pre o posposición). Utiliza dos atributos, MAX y QU con valores + y -, con el propósito de distinguir entre categorías con máxima proyección y aquellas que no la tienen. En términos generales, una categoría con proyección es aquella que puede dominar sobre otras categorías (del vocabulario no terminal). Las otras categorías son aquellas que dominan sobre elementos terminales. Por ejemplo, N tiene como núcleo n , -MAX y -QU. SN es una categoría maximal con núcleo n y que puede estar marcada como +QU o -QU dependiendo de la naturaleza del ítem. Normalmente,

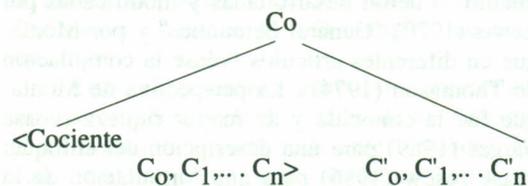
estos aspectos son expresados en términos de ecuaciones de la siguiente forma:

- (9) $N = \{ \langle \text{núcleo: } n \rangle \langle \text{MAX -} \rangle \langle \text{QU -} \rangle \}$.

Tenemos, en tercer lugar, los rasgos que los autores denominan contextuales y las formas. Los primeros incluyen: número que puede ser *sing*, *plu* o # (este último cuando el rasgo está indefinido); persona: *masc*, *fem*, *neutro* y % (para indefinido); caso: *nomi*, *acus*, % (indefinido); índice: 1, 2, f, g. Las formas o *uform* incluye: *fin* (finita), *infin* (infinitiva), *bas* (base), *part* (participio) y *pas* (pasiva).

Finalmente, las categorías pueden ser de dos tipos: de *cociente* (Quotient) y de *almacenamiento* (Stored). En los enfoques lingüísticos actuales se asigna a algunos elementos de la oración un papel activo en la construcción, esto es, el componente que, desde el punto de vista informacional, subsume mayor cantidad de información. Por ejemplo, el verbo desempeña el papel activo a nivel de oración (la cabeza, como veremos en el caso de HPSG). La categoría cociente se introduce con la finalidad de expresar este hecho. Las construcciones en las que están involucradas categorías cociente tienen la siguiente forma:

- (10) $\langle \text{Cociente } C_0, C_1, C_2, \dots, C_n \rangle$ (4)



La categoría cociente se combina con n argumentos ($n \neq 1$) para producir una expresión de categoría C_0 . Esta última es denominada categoría madre.

La introducción de las categorías de almacenamiento tiene que ver con el tratamiento de algunos fenómenos relacionados con lo que se conoce como "Dependencias no ligadas" (Unbounded dependencies). Estas construcciones tienen la característica de que existe un constituyente extra desde el punto de vista de la oración, pero cuando se observa dentro de la oración falta un constituyente. Las dependencias no ligadas tienen una variedad de formas. Entre ellas, topicalización, preguntas y cláusulas relativas.

- (11) i- El hombre que María encontró ... (relativización).
 ii- A quién encontró María (preguntas).
 iii- A María, la golpeó su esposo (topicalización).

Las categorías de almacenamiento se aplican fundamentalmente a SN o SP, pero nunca a los V (verbos). Sin embargo, como se recordará del ejemplo 10.b., el vacío generado por el SN o SP que ha sido almacenado se propaga por el árbol hasta que sea llenado (a ese proceso se le denomina "por colación" (percolation)). La diferencia es que en cada nodo se indicará el vacío en cuestión.

Se puede definir las categorías de almacenamiento de la siguiente manera:

Si d es una expresión de la categoría cociente de la forma $\{ \langle \text{Cociente } C_0, C_1, C_2, \dots, C_i, \dots, C_n \rangle \}$, donde $n \geq 1$, entonces, debe existir una categoría de almacenamiento de una de las dos formas:

i- para $n > 1$

$\{ \langle \text{Cociente } C_0, C_1, C_2, \dots, C_{i-1}, C_{i+1}, \dots, C_n \rangle \} \gg \{ \langle \text{almacenado}, C_i \rangle \}$;

ii- para $n = 1$

$C_0 \gg \{ \langle \text{almacenado}, C_i \rangle \}$
 bajo la condición de que C_i sea una categoría maximal indizada (Yoo y Lee 1988: 9).

A nivel de lexicón los ítemes contienen toda la información relacionada con sus restricciones sintácticas, sus características léxicas, su información semántica y su información fonológica. Estos ítemes son combinados de acuerdo con algunas reglas que son de tres tipos: a nivel sintáctico o categorial las operaciones de cancelación; a nivel semántico las operación de juntura (merge) y a nivel fonológico mediante operaciones de concatenación.

Las operaciones de cancelación son tres fundamentalmente: la de aplicación funcional (f^{cancel}), la de cancelación de almacenamiento ($f^* \text{cancel}$) y la cancelación por colación ($f^{\Delta*} \text{cancel}$). La primera se aplica a construcciones que involucran categorías cociente, las segundas a categorías en las que hay almacenamiento y las últimas a procesos en los que el vacío de una expresión almacenada se conserva en los diferentes niveles del árbol.

La primera clase de operaciones es formalizada del siguiente modo:

Para cualquier categoría C , y cualquier secuencia no vacía G de categorías, y S cualquier conjunto, entonces, $f^{\text{cancel}}((C, G)_i \gg E, S) = C \gg E$ bajo la condición de que G subsuma S , donde E es o un conjunto vacío de categorías o una categoría $\{ \langle \text{almacenado}, D \rangle \}$ (pág. 14).

Recuérdese que una categoría A subsume B si, y solo si, B contiene al menos igual cantidad de información que A .

La operación de cancelación de almacenamiento toma la forma siguiente:

Para cualquier categoría no-cociente C y categorías D y D' , $f^* \text{cancel}(C \gg \{ \langle \text{almacenado}, D \rangle \}, D') = C$ bajo la condición que D subsuma D' .

Finalmente, las operaciones por colación toman la forma siguiente:

Para cualesquiera categorías C, A, A' y D , $f^{\Delta*} \text{cancel}((C, A), A'/D) = C/D$ bajo la condición de que A subsuma A' . (pág. 15).

Las operaciones de juntura combinan mediante la operación subsunción la información que cada uno de los ítemes léxicos y constituyentes conlleva, siempre que se cumplan las dos condiciones siguientes:

- 1- La información de cada uno de ellos es compatible.
- 2- La juntura de información respeta las operaciones de cancelación de categorías anteriormente enunciadas.

Las operaciones de concatenación se aplican a los componentes fonológicos de la oración en el contexto de los dos aspectos siguientes:

- 1- Restricciones sintácticas o categoriales de combinación de categorías o constituyentes;
- 2- Los principios de precedencia que rigen la lengua en estudio.

Los principios de precedencia establecen el orden de aparición de los distintos componentes de la oración. Por ejemplo, para una lengua del tipo SVO, el SN sujeto precede al SV y la preposición al SN correspondiente. Cada lengua tiene principios de precedencia diferentes, pero que es necesario considerarlos para una com-

prensión adecuada de su estructura y funcionamiento.

4. Gramática de estructura sintagmática

La incorporación de conjuntos de rasgos y valores dentro de gramáticas que utilizan reglas libres de contexto fue utilizada de manera central, y por primera vez, en la gramática léxico-funcional propuesta por Kaplan y Bresnan (1983); lo mismo las operaciones de unificación y generalización. También, desempeña un papel muy importante en la Gramática de Estructura Sintagmática Generalizada (GPSG: Generalized Phrase Structure Grammar) propuesta por Gazdar y otros (1982 y 1985), y en otras teorías sintácticas. Sin embargo, estas gramáticas no son estrictamente basadas en unificación, sino que los incorporan como un subcomponente.

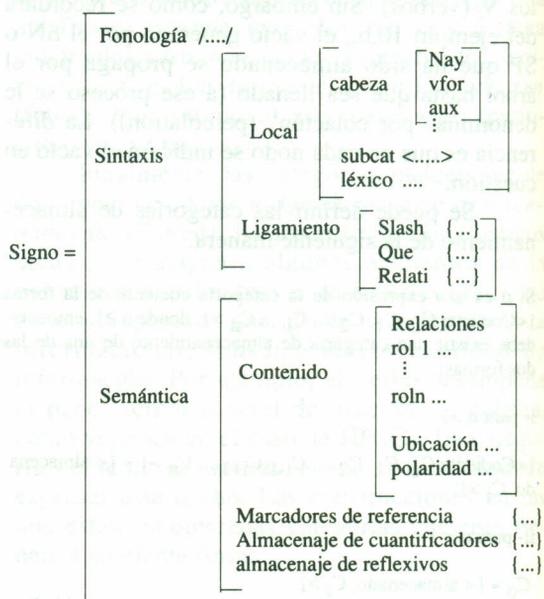
PATR-II es un formalismo basado enteramente en unificación y que es utilizado para especificar gramáticas o componentes de gramáticas. Este formalismo incorpora la estructura descrita en la sección 2. PATR-II fue desarrollado por Shieber en el Laboratorio de Palo Alto y en la Universidad de Stanford. Posteriormente, Karttunen desarrolló una versión de PATR-II en la Universidad de Texas. PATR-II es una fuente importante de intuiciones para los investigadores en las áreas de computación, inteligencia artificial y procesamiento de lenguas naturales.

El ejemplo de gramática basada en unificación que queremos presentar se conoce con el nombre de HPSG (Head-Driven Phrase Structure Grammar). Se denomina así porque la noción de 'cabeza' desempeña un papel fundamental en el modelo. HPSG fue desarrollada por Pollard y Sag en la Universidad de Stanford. Tiene su origen en la tesis doctoral de Pollard (1984) relacionada con la discusión de GPSG de Gazdar y otros. En este momento es la que mejor expresa los requerimientos del enfoque unificacional.

Para HPSG la caracterización de una lengua involucra los aspectos siguientes: 1- un conjunto de principios universales, válidos para todas las lenguas; 2- un conjunto de principios particulares (válidos para esa lengua); 3- un conjunto de operaciones o reglas y 4- un conjunto de rasgos y valores. Estos últimos permiten, entre otras cosas, imponer las restricciones que los distintos ítems léxicos conllevan, distinguir constituyentes de otros y establecer generalizaciones.

Quisiera presentar brevemente cada uno de estos componentes, comenzando en orden inverso, del último al primero.

Para HPSG cada pieza, sea léxica o no, o conjunto de piezas que conlleve información se denomina "signo". Cada signo conlleva información de al menos tres tipos: fonológica, sintáctica y semántica. La estructura de cada signo en el lexicon toma la forma siguiente:



Tomado de Pollard y Sag (1987); pág. 10

El aspecto fonológico registra la realización fonológica del signo si la tiene, así como sus variaciones y restricciones. Este componente ha sido poco estudiado dentro de HPSG. El componente sintáctico está regido por dos rasgos: el local, que registra la información sobre aspectos como los siguientes: si el signo en cuestión es elemento activo o no (cabeza), cuál es el constituyente sintáctico al que pertenece (MAYOR); si es un verbo, si está en forma conjugada o no. El rasgo de subcategorización codifica las restricciones que este signo impone sobre otros constituyentes. Por ejemplo, cierta clase de verbos en español, los verbos de deseo o expectativa implican que la oración subordinada esté en subjuntivo (**Deseo que María venga, espero que María venga**). El rasgo léxico puede tomar dos valores + o -, dependiendo de si el signo es un ítem o una frase. Los rasgos de ligamiento se proponen para dar cuenta de varios fenómenos de dependencias

no ligadas, del tipo señalado en el apartado anterior. Con la diferencia de que el rasgo SLASH se utiliza más en el sentido de las gramáticas categoriales clásicas. Para estas gramáticas una construcción como S/SN significa a partir de la categoría S/SN podemos obtener un S (oración) mediante la agregación de un SN. En ECG no hay necesidad de incluir estos mecanismos ya que la operación de concatenación se aplica sobre conjuntos de categorías y no sobre una de ellas. El rasgo de ligamiento, como se observa, no es local sino que se propaga a diferentes partes del árbol.

El componente semántico de un signo viene determinado por el conjunto de relaciones que puede establecer con otros signos y con el tipo de roles que puede desempeñar. Por ejemplo, si el signo es un SN, éste puede ser el sujeto de la oración, el objeto o el instrumento. Por ejemplo, "el martillo" puede ser el sujeto de la oración ("el martillo es pesado"), el objeto, ("Juan botó el martillo") e instrumento ("Juan colgó la puerta con el martillo"). Estos aspectos deben ser acoplados con los de subcategorización. El componente semántico también incluye los marcadores de referencia que indican las restricciones sobre el orden de aparición de los constituyentes, la interpretación, respecto al número y tipo de cuantificadores y la aparición de reflexivos.

Las reglas de una gramática pueden ser formuladas de varias maneras. La siguiente es una posible formulación:

- (12) $X \rightarrow X_0, X_1$
 $\langle \text{Cat}, X \rangle = S$
 $\langle \text{Cat}, X_0 \rangle = SN$
 $\langle \text{Cat}, X_1 \rangle = SV$
 $\langle \text{cabeza}, SV \rangle = \langle \text{cabeza}, S \rangle$
 $\langle \text{cabeza}, SN \rangle = N$
 $\langle \text{cabeza}, SN \rangle = V$
 $X_0 < X_1$

La regla $X \rightarrow X_0, X_1$ puede ser instanciada de diversas maneras. El mismo esquema de regla puede ser utilizado para definir otros constituyentes. En lugar de S, SN y SV pudimos haber utilizado SN, Det y N, con los mismos resultados. Se requieren pocos esquemas para dar cuenta de una gran cantidad de reglas.

HPSG utiliza también esquemas de reglas pero de una forma ligeramente diferente. Un

esquema libre de contexto como el siguiente permite dar cuenta de un conjunto significativo de reglas:

- (13) $[\text{SUBCAT} \langle \rangle] \rightarrow H [\text{LEX} -, C$

Esta regla indica que cualquier constituyente para el cual el rasgo de subcategorización es vacío, puede reescribirse como la cabeza no léxica (esto es, no puede ser verbo o nombre) y un complemento. Si imponemos que $H [\text{LEX} -] < C$, entonces, esta regla permite abreviar:

- (14) $S \rightarrow SN, SV$
 $SN \rightarrow SNs, \text{Nombre}$

Esta última regla es generable ya que no se especifica cuál debe ser el número de constituyentes que conforme la cabeza no léxica. Si la condición anterior no se impone, entonces, de (13) abrevia: $SN \rightarrow \text{det Nombre}$.

La siguiente regla (donde C^* se interpreta como la cerradura, esto, cero o cualquier número finito de complementos) permite la generación de otro número significativo de reglas; algunas de ellas enumeradas en (16):

- (15) $[\text{SUBCAT} \langle [] \rangle] \rightarrow H [\text{LEX} +], C^*$

- (16) $SV \rightarrow V$
 $SV \rightarrow V SN,$
 $SV \rightarrow V SN SN SP$
 $SV \rightarrow V SP$

Las reglas léxicas, como son denominadas, no tienen por qué generar únicamente construcciones bien formadas. La idea de que una gramática basada en reglas de reescritura debe generar solamente el conjunto de expresiones bien formadas, es abandonada cada día por mayor número de lingüistas. Lo que se busca, más bien, son principios generales que permitan restringir el conjunto de expresiones generables por un conjunto de reglas. Estos principios constituyen el marco dentro del cual una lengua determinada puede "moverse" y evolucionar.

Como indicamos al principio de esta sección, para HPSG existen dos conjuntos de principios que complementan las reglas utilizadas por una gramática. Los primeros son principios válidos para una o un conjunto de reglas; existe otro conjunto que es válido para todas las lenguas (principios

universales). Dentro del primer conjunto, existen principios como los siguientes:

1- Principios de precedencia (como hemos indicado, estos principios tienen que ver con el orden de aparición de los constituyentes). En español, tanto el artículo como la preposición preceden al N.

2- Principios de subcategorización (requerimientos impuestos por algunos constituyentes sobre otros en la oración). En español, un nombre siempre tiene que estar acompañado de un artículo.

Entre los principios universales propuestos por HPSG, podemos indicar los siguientes:

HFP (Head Feature Principle)
SUBCATEGORIZACION
FFB (Foot Feature Binding)

El primer principio establece que el conjunto de rasgos de los hijos (en árboles los hijos son los descendientes directos del nodo, conocido como madre) del constituyente cabeza, y los de este último, deben concordar entre sí. Esto es, Si S es el conjunto de rasgos de los hijos y P el del constituyente cabeza, entonces $S = P$. "Si el signo X es un constituyente fraseal e y es el conjunto de hijos de X, entonces, X e Y comparten los valores para la ruta (path) SYNTAX/LOCAL/CABEZA" (Pollard y Sag 1987:17 y Pollard y Sag 1987a: 45 y ss). Este principio siguiendo a estos autores puede ser formalizado como sigue:

(17) Signo-fraseal [] ==>

Sint/Local/ cabeza	1
Hijas/Cabeza-hijas/Local/cabeza	1

Este principio tiene muchas aplicaciones en español. En efecto, en español, el SN sujeto y el verbo pueden compartir un conjunto de sus rasgos, aquellos del sujeto; también el adjetivo y el artículo deben tener en común un conjunto de rasgos con el N.

El segundo principio, el de la SUBCATEGORIZACION establece que el conjunto de rasgos de la madre y de los hijos tienen que ser acoplados. La unificación del conjunto de rasgos de los elementos subcategorizados se da cuando no existe incompatibilidad con los de la madre. El

principio puede ser establecido del siguiente modo:

(17) Signo-fraseal [] ==>

Sint/Local/ subcategorización	2						
hijos	<table border="1"> <tr> <td>hijos-cabeza/ Sint/Local/ subcat</td> <td>Unir</td> <td>(1 2)</td> </tr> <tr> <td>compl-hijos</td> <td>1</td> <td></td> </tr> </table>	hijos-cabeza/ Sint/Local/ subcat	Unir	(1 2)	compl-hijos	1	
hijos-cabeza/ Sint/Local/ subcat	Unir	(1 2)					
compl-hijos	1						

Donde Unir es un operador que toma dos listas L_1 y L_2 y retorna la concatenación de ellas. Esto es,

$Unir(L_1, L_2)$ (donde L_1 consta de $\langle l_1, l_2, \dots, l_n \rangle$ y L_2 de $\langle l_m, \dots, l_x \rangle$ son elementos no necesariamente disyuntos) = $\langle l_1, l_2, \dots, l_n, l_m, \dots, l_x \rangle$.

El tercer principio establece que si $X \rightarrow YZ$ es una regla, entonces, el conjunto de rasgos de Y y Z son heredados por X. Esto es, recorren el árbol de abajo hacia arriba respetando las restricciones impuestas por las reglas.

Formalmente expresado,

Signo-fraseal [] ==>

SINTAX/LOCAL/ SUBCAT HIJOS/SIN-
TAX/LOCAL/SUBCAT

La anterior constituye una caracterización bastante breve de dos importantes formulaciones lingüísticas en las que la Unificación desempeña el papel fundamental. El paradigma informacional desempeña un papel fundamental

Notas

- (*) Quiero agradecer al Dr. Jack Wilson, por la lectura y observaciones a este artículo.
1. Sin embargo, es una tarea de naturaleza empírica determinar cuáles conjuntos de estructuras son los adecuados para una descripción lingüística particular. El formalismo no debe ser restringido. Por ello presentaremos aquí una perspectiva general.
 2. Un descriptor es un elemento del conjunto de rasgos que restringe el valor un elemento cualquiera en ciertos niveles.
 3. La UCG desarrollada por Zeevat, Klein y Calder (1986) comparte muchas características con ECG.

Entre las diferencias está el hecho de que UCG se aproxima a la teoría de la representación del discurso propuesta por Kamp (1981) y no a la semántica de las situaciones.

4. A diferencia de enfoques como el de Montague y otras gramáticas categoriales, ECG permite que múltiples argumentos sean unificados al mismo tiempo. Los otros enfoques permiten la aplicación de una operación a un argumento por vez.

Bibliografía

- Barlow (1988) *Unification and agreement*. CSLI, Stanford University.
- Gazdar y otros (1985) *Generalized phrase structure grammar*. The MIT Press, Cambridge.
- Josew (1986) "Montague Grammar and First-Order Logic" en: *Categorial grammar, unification grammar and parsing*. Edinburgh University; pp 143- 194.
- Kaplan y Bresnan (1982) *The mental representation of syntactic structure*. MIT.
- Karttunen (1986) "Features and Values" en Shieber et al. (1986) *A compilation of papers on unification-based grammar formalisms*. Parts I and II, CSLI, Stanford University; pp 17-33.
- Knight (1989) "Unification: An Interdisciplinary Survey" En: *Computer survey*.
- Lewis (1970) "General Semantics". En Partee (editor) (1976) *Montague grammar*. Academic Press, USA; pp 1-53.
- Pereira y Shieber (1986) "The semantics of Grammar Formalisms Seen as Computer Languages" En Shieber et al. (1986); 37-54.
- Pollard (1984) *Generalized context-free grammars, head grammars and natural language*. Tesis de doctorado, Stanford University.
- Pollard y Sag (1987) *Head-driven phrase structure grammar: An informal synopsis*. CSLI, Stanford University.
- (1987a) *Information-based syntax and semantics*. Vol. 1. CSLI, Stanford University.
- Robinson (1965) "A Machine Oriented Logic Based on the Resolution Principle" En: *Journal of the Association of Computer Machinery*. Marzo (12).
- (1979) *Logic: form and function*. Elsevier North Holland Inc., USA.
- Scott (1982) "Domains for Denotational Semantics", ICALP, July
- Shieber (1986) *An introduction to unification-based formalisms*. CSLI, Stanford University.
- Thomason (1976) *Formal philosophy: a compilation papers of Richard Montague*.
- Uzkoreit (1986) *Unification categorial grammar*. CSLI, Stanford University.
- Vargas, Celso (1989) "Aspectos del Programa Lingüístico General de Montague" En *Revista de Filología y Lingüística de la Universidad de Costa Rica*.
- Yoo y Lee (1988) *Extended categorial grammar*. CSLI. Stanford University.
- Zeevat, Klein y Calder (1986) "Unification Categorial Grammar". En: *Categorial grammar, unification grammar and parsing*. Center for cognitive science. Edinburgh University. Vol. I.

