

## **Transformación de algunas estructuras sintácticas en un sistema de traducción automática**

T.Redondo, P.Rodríguez, L.Sopeña, I.Zapata  
Centro de Investigación UAM-IBM  
Madrid

### **Resumen**

Dentro del proyecto de Traducción Automatizada **Multilingual English Translator (M88/S)**, cuyo objetivo es la construcción de un prototipo de traducción de textos del inglés al castellano, se describen brevemente los criterios seguidos en el tratamiento de algunas de las estructuras complejas más frecuentes. Se detallan los fundamentos lingüísticos y las decisiones y acciones llevadas a cabo en la fase de *transferencia* del sistema.

## *umen*

entro del proyecto de Traducción Automatizada **Multilingual English Translator (M88/S)**, cuyo o es la construcción de un prototipo de traducción de textos del inglés al castellano, se describen ente los criterios seguidos en el tratamiento de algunas de las estructuras complejas más frecuentes. allan los fundamentos lingüísticos y las decisiones y acciones llevadas a cabo en la fase de *transfe*- del sistema.

## *oducción*

ultilingual English Translator (M88/S) es un proyecto de Traducción Automática Asistida ac- nte en desarrollo en el Centro de Investigación UAM-IBM, Madrid. El objetivo es construir un po que traduzca automáticamente al castellano textos escritos en inglés.

omo ocurre en la mayoría de los sistemas de TA, el dominio de la aplicación está restringido; en o caso, se trata de los manuales de IBM, decisión tomada también por otros proyectos similares ara la elaboración del prototipo nos basamos en un corpus de 500 oraciones extraídas de uno de di- anuales.

as características generales del sistema han sido ya presentadas con anterioridad [2]. En este docu- describimos más en detalle el tratamiento efectuado sobre algunas estructuras complejas del inglés, ricular atención a las que aparecen con más frecuencia en nuestro corpus.

uestro sistema de TA sigue el modelo *transfer*, en consecuencia, el proceso de traducción se divide fases: análisis del texto fuente, transferencia y generación del texto objeto.

ara la etapa de análisis utilizamos una gramática del inglés, PEG [3], escrita en el lenguaje PLNLP |, y un amplio léxico (basado en los diccionarios Webster-7 y Longman) [5], componentes ya des- rollados previamente. PEG genera, para cada oración, una descripción sintáctica o *registro*, en rma de una colección de estructuras atributo-valor, y, a partir de ella, construye un árbol de análisis ás legible, utilizando una parte de la información de dicha estructura.

proceso de transferencia consta fundamentalmente de dos aspectos: *transferencia léxica*, que se- ciona el significado específico de cada palabra, y *transferencia estructural*, que decide cuáles son las rmas sintácticas adecuadas. Ambas están íntimamente relacionadas y deben interactuar a lo largo l proceso. Incluyen siempre acciones de desplazamiento y reordenación de los elementos de la acción, junto con la desaparición de algunos elementos y, en algunos casos, la aparición de nuevos dos.

. fase final de generación produce la oración de salida a base de recorrer la estructura previamente nstruida, flexionando las palabras en la forma adecuada y empleando un diccionario específico [6].

lo que sigue utilizaremos los resultados que PEG proporciona como punto de partida para estudiar intos casos tratados, tomando, entre otros, ejemplos del manual que constituye nuestro corpus.

## *ucturas complejas*

r estructuras complejas entendemos aquellas en que intervienen elementos cuya combinación en sí podría constituir una frase (*clause*). Las estructuras complejas que tratamos en este apartado son denominamos **INFCL** (frases de infinitivo) y las **THAT-CL** (relativas y completivas). En el apar- *emplos de transformación* se muestran esquemas de algunos árboles resultantes del análisis y las

# NFCL

En inglés podemos detectar diversas formas en que se presenta la oración de infinitivo: ir o no precedida por "to"; llevar o no antepuesta una preposición (lo que implica que el verbo aparezca en la forma del *present participle*), etc. Igualmente, la frase de infinitivo puede llevar a cabo diversas funciones: ser la *complementiva* de un grupo nominal, que es a su vez el sujeto de la misma, y que está en acusativo (*direct object*) por tratarse del objeto de la acción expresada por el verbo principal (flexionado); puede expresar finalidad o motivo de una acción (*to ... = = = > para ...*); etc. Ejemplos pertinentes a todos estos casos están detallados más abajo.

Por su propia naturaleza (verbo no flexionado), la INFCL está supeditada a aparecer dentro de una estructura de rango superior: *Noun Phrase (NP)*, *Adjective Phrase (AJP)*, etc., donde el núcleo es el propio de estas estructuras, es decir, un *nombre* en NPs, un *adjetivo* en AJs, etc.

Hasta ahora hemos tratado las siguientes estructuras de rango superior en las que se inscribe una NFCL:

**VERB + INFCL(OBJECT) = = = > VERB + INFCL(OBJECT)**

P.e.: *I want to go = = = > (yo) quiero ir*

**INFCL(SUBJECT) + VERB = = = > INFCL(SUBJECT) + VERB**

P.e.: *To go to the cinema is funny = = = > Ir al cine es divertido*

**VERB + NP(OBJ INFCL) = = = > VERB + NP(COMPL SUBJUNC)**

NP(OBJ INFCL) significa que el NP que sigue al verbo es Objeto Directo y contiene una INFCL. NP(COMPL SUBJUNC) significa que el Objeto Directo del verbo es una oración complementiva cuyo núcleo verbal adopta el modo subjuntivo.

En este caso, el verbo puede:

- Pertener a una cierta clase (marcada con TOSUBI en el diccionario, y que incluye a *want*, *wish*, etc.):

*I want him to go to the cinema = = = > (yo) quiero que (él) vaya al cine*  
*You expect the user to issue a command = = = >*  
*usted espera que el usuario proporcione un mandato*

- No pertenecer a esa clase (estos verbos tienen una estructura superficial similar, pero su traducción no es idéntica):

*I asked him to go to the cinema = = = > (yo) le pedí que (él) fuera al cine*  
*It lets your program call macros = = = >*  
*(esto) permite que su programa llame a macros*  
*It allows the system to query variables = = = >*  
*(esto) permite que el sistema examine variables*  
*You wait for the user to issue a command = = = >*  
*usted espera a que el usuario proporcione un mandato*

**IT + VERB + AJP(ADJ+INFCL) = = = > VERB + AJP(ADJ+INFCL)**

AJP(ADJ+INFCL) significa que el AJP está compuesto de un adjetivo y una INFCL. Además, IT es un sujeto vacío y el verbo es copulativo.

P.e.: *It is convenient to have a routine = = = >*  
*es conveniente tener una rutina*

**NP + VERB + AJP(ADJ+INFCL) = = = > NP + VERB + AJP(ADJ+PP INFCL)**

AJP(ADJ+PP INFCL) significa que el AJP está compuesto de un adjetivo y un sintagma preposicional, en el que una preposición introduce una INFCL. Ahora, el NP es el sujeto de la oración y el verbo es copulativo.

*la fórmula es imposible de aprender*

(que es una variante de: *es imposible aprender(se) la fórmula*).

5. NP + VERB + NP(NOUN + INFCL) = = = > NP + VERB + NP(NP + PP INFCL)

NP(NP + PP INFCL) significa que el NP está compuesto por un NP y un sintagma preposicional, en el que una preposición introduce una INFCL.

P.e.: *It is a simple way to issue a command* = = = >  
*es un modo simple de dar un mandato*

P.e.: *John knows a way to improve it* = = = >  
*Juan conoce una manera de mejorarlo*

7. Expresión de la Finalidad mediante INFCL (*purpose clauses*)

Si comparamos entre sí las oraciones siguientes:

*I want him to go* (del tipo 3) = = = > *(yo) quiero que él vaya*  
*I want a car to go home* = = = > *(yo) quiero un coche para ir a casa,*

la diferencia que observamos es más semántica que sintáctica y, por lo tanto, no es reconocible (ni procesable) en este nivel. Compárese también:

a. *I invoke it to get a program* = = = > *(yo) lo llamo para obtener un programa*  
b. *they need a bigger apartment to move in* = = = >  
*(ellos) necesitan un piso más grande al que trasladarse*

y podremos observar que en a) el NP (objeto) puede anteponerse al verbo en castellano, con lo cual hay una transformación adicional que no se presenta en el tipo 3 antes analizado. Compárese también b) con

c. *they need a friend to move in* = = = > *(ellos) necesitan que un amigo se traslade*

donde la misma combinación estructural anterior (que además también comparte el verbo principal y el de la INFCL) da como resultado una traducción diferente por no tratarse en este caso de una expresión de finalidad (aunque ésta sea también una de las posibles interpretaciones).

La forma en que de momento detectamos esta estructura es buscando el objeto de la oración, ya que la expresión de finalidad no puede en ningún caso ser objeto. Por lo tanto, si la INFCL es objeto se procesará tal como se ha descrito en los puntos precedentes. Si no lo es, expresará finalidad:

P.e.: *I use it to get a compiled version of my program* = = = >  
*(yo) lo uso para obtener una versión compilada de mi programa*

Un caso especial (por no aparecer explícita la INFCL en el texto fuente) son las oraciones que están formadas por una preposición seguida de un verbo. La caracterización del inglés en este caso es que el verbo aparezca en *participio presente (prespart)* en vez de en *infinitivo*. En términos generales lo traduciremos por una oración introducida por una preposición y seguida de una INFCL:

• **PREP + VERB (PRESPART) = = = > PREP + INFCL**

P.e.: *without doing ...* = = = > *sin hacer ...*

## THAT-CL

Las *that-cl* (aquí unificamos todas aquellas *clauses* que presentan, implícita o explícitamente, un *that* como pronombre relativo o conjunción completiva) pueden ser básicamente de dos tipos:

- posmodificador de un *noun* en NP, caso que PEG analiza como *relcl* (oración de relativo) y al *that* como *pronombre* (relativo)
- objeto directo del verbo transitivo correspondiente, donde PEG ofrece como resultado del análisis una

Debe recordarse aquí el fenómeno peculiar que presenta el inglés en el primer tipo, donde el pronombre se puede elidir en un número bastante elevado de ocasiones sin que por ello se pierda la relación que se establece, caso que la gramática analiza correctamente.

En cualquiera de ambos tipos señalados se requiere la presencia de un verbo sobre el que se articulan los demás elementos. La partícula *that* aparece en dos modalidades:

1. *pronombre relativo* (en este caso similar a 'which', 'who', ...)

- explícito  
P.e.: *The car that is white ...* = = = > *El coche que es blanco ...*
- implícito  
P.e.: *The man (that) I saw yesterday ...* = = = > *El hombre que vi ayer...*

Esta ausencia del pronombre se resuelve introduciendo un nodo en la estructura de la oración que describa la existencia del mismo tal y como PFG lo analizaría en caso de aparecer. Se ha considerado que la elección del pronombre *that* es correcta, ya que sirve tanto para antecedentes animados como inanimados. En el apartado siguiente se muestra un ejemplo de esta transformación.

2. *conjunción completiva*

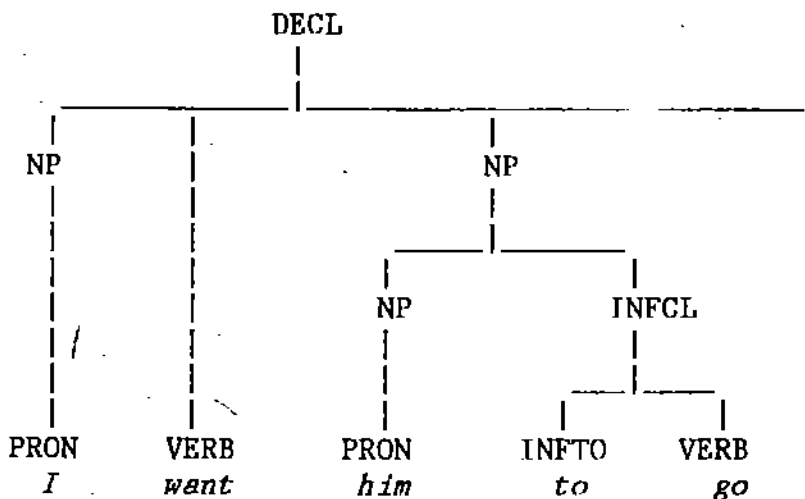
- explícita  
P.e.: *ARG can determine that the number has passed* = = = > *ARG puede determinar que el número ha pasado*
- implícita  
P.e.: *He told me it was wrong* = = = > *El me dijo que estaba mal*

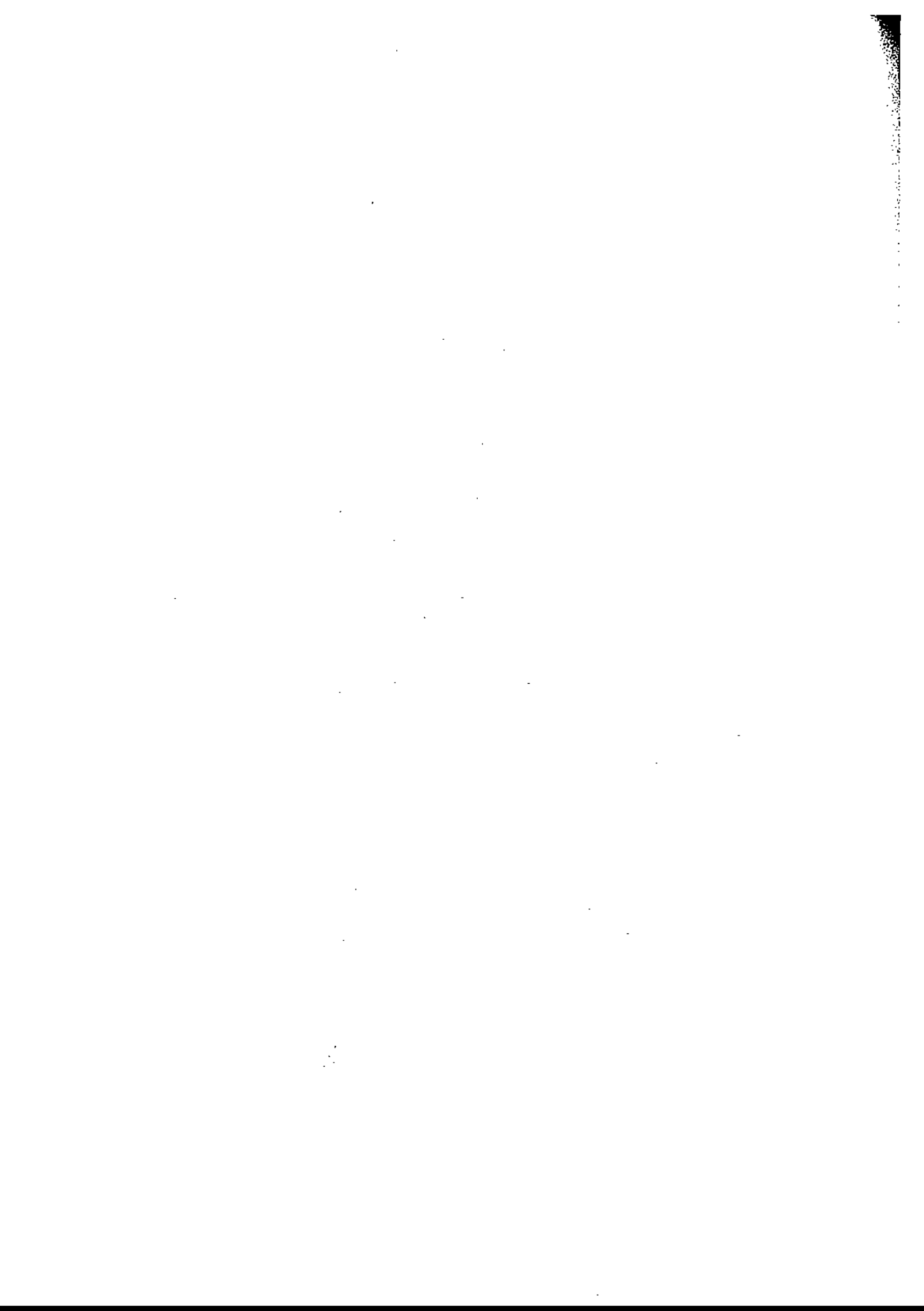
El criterio adoptado es el mismo que para las subordinadas de relativo, es decir, construir un nodo que incluya la conjunción completiva.

## Ejemplos de transformación

Para hacer más gráfica la explicación de cómo se ha procedido a transformar las estructuras complejas utilizamos la representación arbórea como la más conveniente. Los árboles que se muestran reflejan algunos de los ejemplos mencionados en el apartado anterior.

**EJEMPLO 1** "I want him to go"





## **Notas sobre el formalismo de la GPSG.**

*J.A.Alonso, J.C.Ruiz*

## NOTAS SOBRE EL FORMALISMO DE LA GPSG

J.A. Alonso (SIEMENS)  
J.C. Ruiz Anton (EUROTRA)

### 1.1. - Fundamentos de la GPSG

La Gramática Sintagmática Generalizada (*Generalized Phrase Structure Grammar* o GPSG) es una gramática generativa en el sentido clásico, i.e. una teoría formal que asigna descripciones estructurales a las oraciones. Una Gramática generativa consta de reglas que determinan explícitamente todas y (sólo) las oraciones bien formadas de una lengua.

La GPSG (Gazdar, Klein, Pullum y Sag 1985, a partir de aquí abreviado en GKPS) insiste en la preocupación por el rigor formal y la precisión matemática que a juicio de los autores ha de tener la teoría gramatical. Uno de los objetivos de este formalismo es alcanzar un metalenguaje restringido capaz de definir las gramáticas de las lenguas naturales.

Básicamente, la GPSG asume lo siguiente:

- Una gramática con reglas *sintagmáticas* es capaz de generar los lenguajes naturales. Al asumir esta hipótesis, la GPSG rechaza la conocida opinión contraria de Chomsky (1956), en la cual se fundamentó la introducción de transformaciones en la teoría generativa.
- En consecuencia, la GPSG no hace uso de transformaciones, entendidas como reglas que transforman (proyectan) representaciones en otras transformaciones.
- La GPSG parte de un único nivel de representación oracional. Se trata de una estructura de constituyentes, muy similar a la estructura superficial de Chomsky (1965). La GPSG es, por tanto, una teoría *monoestratal*, frente a otros modelos como la Gramática Léxico Funcional (LFG)<sup>1</sup> o la Teoría de la rección y del ligamen (GB)<sup>2</sup>, que admiten varios niveles de representación.

A diferencia de las versiones clásicas de la Gramática Generativa, la GPSG no define directamente sus reglas sintagmáticas. En su lugar, recurre a técnicas cuyo efecto es hacer la gramática más compacta de lo que sería una simple lista de reglas, y a la vez permiten extraer importantes generalizaciones lingüísticas. Estas técnicas son:

<sup>1</sup> cf. Kaplan / Bresnan (1982).

<sup>2</sup> *Government and Binding Theory* (GB) cf. Chomsky 1981, 1985.

- una teoría muy elaborada de los rasgos sintácticos.
- factorización de las reglas sintagmáticas en reglas de dominio, reglas de precedencia y metarreglas.
- principios de instanciación de rasgos.

## 1.2.- El concepto de categoría en la GPSG; Teoría de Rasgos

En GPSG, una categoría se define como un símbolo complejo compuesto de un conjunto de pares atributo-valor (rasgos). Es importante tener en mente que esto no significa que una categoría contenga un conjunto de rasgos, sino que es éste el que conforma una categoría dada. Las etiquetas utilizadas en las reglas para referirse a diferentes categorías (NP, VP, PP, etc.) son únicamente siglas mnemotécnicas que se refieren a determinados conjuntos de rasgos. Así por ejemplo:

NP se refiere al conjunto [+N,-V,BAR 2,...]

VP se refiere al conjunto [-N,+V,BAR 2,...]

etc.

Cada rasgo se compone de un atributo, que representa una cierta propiedad gramatical relevante para la teoría (género, número, persona, tiempo, nivel de barra, etc.), y un valor asociado (singular, primera, presente, 2, etc.).

Hay diferentes tipos de rasgos, según éstos tomen valores atómicos o categoriales. Algunos ejemplos de rasgos con valores atómicos son:

PER [1,2,3]

PAST [+,-]

BAR [0,1,2]

etc.

Los rasgos con valores categoriales, como por ejemplo SLASH, AGR, RE o WH, toman como valor un conjunto de rasgos que especifican (en menor o mayor medida) una categoría. Así, por ejemplo:

AGR[+N,-V,PER 3,-PLU,CA NOM,...]

se refiere al rasgo AGR (concordancia) con un valor que representa un NP singular acusativo.

## 1.3.- Concepto de unificación.

La GPSG se apoya extensamente en el concepto de unificación. Se denomina unificación a la unión de dos categorías (conjuntos de rasgos) en el caso de que los valores de sus respectivos rasgos no sean incompatibles. Por ejemplo; sean las categorías (1), (2) y (3):

(1) [+N,-V,-PLU,CA NOM]

(2) [+N,BAR 2, PER 3]

(3) [+N,-V,+PLU]

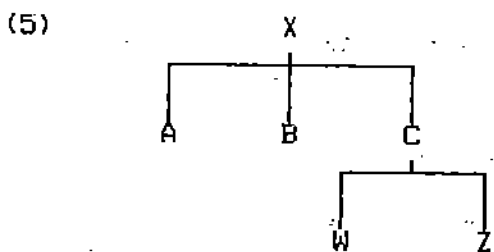
(1) unificaría con (2) en la categoría extendida (4), mientras que en (1) y (3), al tener un valor incompatible para PLU fallaría la unificación.

(4) [+N,-V,-PLU,CA NOM,BAR 2,PER 3]

#### 1.4.- Rasgos instanciados y rasgos heredados.

Un sub-árbol formado por un nodo raíz y todos y sólo los nodos dominados inmediatamente por éste recibe el nombre de árbol local.

Así, en (5), A,B,C conforman un árbol local dominado por X. Lo mismo sucede con W,Z respecto a C.



Un rasgo que forme parte de una categoría en un árbol local se dice que es heredado si la regla ID que induce ese sub-árbol introduce explícitamente dicho rasgo.

Por el contrario, un rasgo se dice que es instanciado cuando su presencia en una categoría de un árbol local se debe a la aplicación de mecanismos como FCRs, FSDs, o los principios de instanciación de rasgos FFP, CAP o HFC, no siendo introducido directamente por ninguna regla ID (cf. punto 2). Así, en el árbol

(6): VP[SLASH[NP]] (= VP/NF)  
V[40]  
VP[FIN,-PLU]

inducido por la regla ID

(7): VP/NF --> H[40], VP[FIN]

los rasgos SLASH y FIN son heredados, mientras que [-PLU] es instanciado.

#### 2.- Mecanismos del formalismo.

Los mecanismos de construcción y validación de árboles con los que trabaja la GPSG son los siguientes:

- \* Reglas de dominancia (ID)
- \* Reglas de precedencia (LP)

- \* Metarreglas
- \* Restricciones de coocurrencia de rasgos (FCR)
- \* Principios universales de instanciación de rasgos (FFP, CAP y HFC)
- \* Especificaciones de rasgos por defecto (FSD)

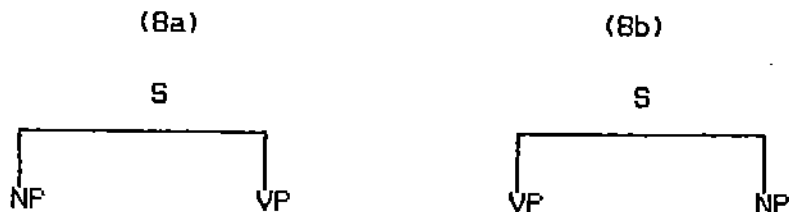
Seguidamente, daremos un breve resumen de cada uno de estos mecanismos:

### 2.1.- Reglas de dominancia (reglas ID).

Las reglas ID (*Immediate Dominance*) indican relaciones de dominancia entre una categoría raíz y una serie de categorías a las que ésta domina, sin dar ninguna información sobre el orden de precedencia de las categorías hijas. Por ejemplo, la siguiente regla ID:

(8): S  $\rightarrow$  NP, VP

generaría dos posibles árboles locales:

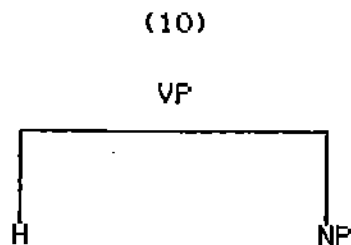


Como convención, la GPSG supone que toda regla ID introduce un núcleo (HEAD) en los árboles locales que induce. Se define como núcleo de un árbol local una categoría H cuyos rasgos N y V coincidan en valor con los de la categoría raíz, y cuyo valor para el rasgo BAR sea igual o menor que el de ésta.<sup>3</sup>

Así, la regla ID

(9): VP  $\rightarrow$  H, NP

introduce árboles del tipo



<sup>3</sup> Esta definición está simplificada respecto a la dada en GKPS (pág. 51). Más estrictamente, H es una abreviatura que representa un conjunto de rasgos que se instanciarán por la aplicación ulterior de la convención HFC (cf. 2.5.3).

donde H es una categoría [+V,-N,BAR 0,...], es decir, un verbo léxico.

## 2.2.- Reglas de precedencia lineal (reglas LP)

Las reglas LP establecen relaciones de precedencia lineal entre constituyentes dentro de árboles locales.

Así, una regla LP como:

(11): [+N] < V[BAR 2] < P[BAR 2]

indica que cualquier constituyente nominal debe preceder a cualquier VP, y ambos deben preceder a cualquier PP dentro de un árbol local. Una regla LP como ésta restringiría los árboles posibles generados por (8) a (8a), ya que (8b) viola (11).

## 2.3.- Metarreglas.

Las metarreglas son un mecanismo para ampliar el conjunto de reglas ID del que dispone la gramática para evitar tener que especificar todas ellas explícitamente.

Una metarregla se aplica sobre una regla ID léxica para dar como resultado otra regla ID léxica, sin anular la primera. Las metarreglas se diferencian de las transformaciones estructurales típicas de las gramáticas generativas en cuanto que:

- \* No se aplican sobre árboles, sino sobre reglas.
- \* No convierten una estructura en otra, sino que a partir de unas reglas ID ya existentes, amplían el conjunto de éstas creando otras nuevas.

Un ejemplo de metarregla es la llamada Metarregla de pasivización, que para el inglés se define como:

(12):           VP --> W, NP  
                  |  
                  V  
          VP(PAS) --> W, (PP[by])

Esta metarregla especifica que para cada regla ID que permita a un VP dominar un NP y algo más (W), debe haber otra regla ID que introduzca una categoría VP pasiva que domine las categorías englobadas en W, junto con (opcionalmente) un PP con un valor by para el rasgo FF0RM.

## 2.4.- Restricciones de Coaparición de Rasgos (FCR)

Las FCRs (*Feature Co-occurrence Restrictions*) establecen restricciones sobre especificaciones de rasgos.

Por ejemplo, una FCR como

(13): FCR 1: [+INV] > [+AUX,FIN]

establece que si se da una especificación de rasgos [+INV] en una categoría, forzosamente se tienen que dar las especificaciones [+AUX,FIN] para esa categoría. En otras palabras, siempre que se dé un caso de cláusula con el verbo en primera posición (inversión), éste deberá ser finito.

## 2.5.- Principios universales de instanciación de rasgos.

La GPSG define tres principios que determinan la buena formación de árboles locales en cuanto a presencia, ausencia o coaparición de rasgos en ciertos nodos clave de dichos árboles. Estos tres principios son los llamados Foot Feature Principle (FFP, o principio de rasgos "foot"), el Control Agreement Principle (CAP, o principio de concordancia de control) y el Head Feature Convention (HFC, o convención de rasgos nucleares).

Seguidamente daremos un breve resumen de cada uno de ellos, utilizando más ideas intuitivas sobre su función y utilidad que el formalismo matemático que se utiliza en el libro para definirlos.

### 2.5.1.- FFP (Foot Feature Principle).

El FFP establece la obligación de que en todo árbol local válido la categoría raíz y todas las categorías hijas unifiquen entre sí sus respectivas especificaciones para los rasgos que pertenecen al subconjunto FOOT, esto es [SLASH,WH,RE].

Una importante restricción del principio es que es válido sólo para rasgos del subconjunto FOOT que estén instanciados, no aplicándose para ningún rasgo heredado (cf. 1.4).

El FFP se ocupa de asegurar la presencia en los nodos necesarios de los rasgos que determinan el control de fenómenos anafóricos, de topicalización o de movimiento (en frases WH, por ejemplo).

### 2.5.2.- CAP (Control Agreement Principle).

El CAP es un principio de buena-formación de árboles que asegura la concordancia entre constituyentes dentro de una frase.

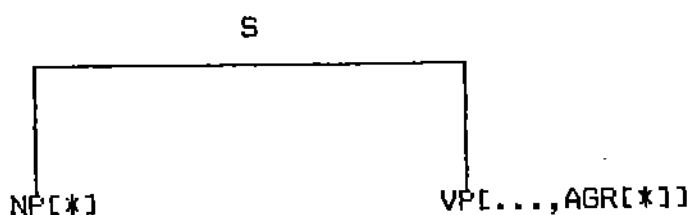
Para ello, se define el concepto de categorías controladoras y controladas, en términos del tipo semántico que cada una de ellas tenga asignado. Sin entrar en detalles, baste mencionar que dentro de la teoría, solo las categorías nominales pueden ser controladoras, mientras que categorías controladas son todas aquellas que en la forma semántica sean funtores cuyos argumentos sean categorías controladoras, es decir, verbos, adjetivos o determinantes.

Dentro de un árbol local, una categoría hija dada puede tener un controlador local o no tenerlo. En el primer caso, el CAP obliga a que el valor del rasgo de control (AGR o SLASH) de la categoría controlada se unifique con la categoría controladora local. En el segundo caso, el CAP obliga a que las

especificaciones para el rasgo de control de la categoría controlada coincide con las de la categoría raíz.

Así, en el árbol (14a)

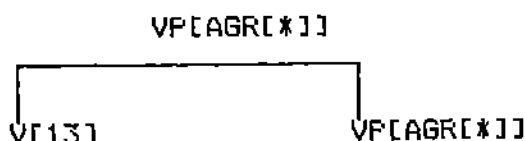
(14a):



el nodo NP controla a VP, y el árbol sólo cumplirá el CAP sólo si las especificaciones de NP, (\*), coinciden con las del rasgo AGR de VP, (\*).

Por otra parte, el árbol (14b)

(14b):



al no tener controlador local, sólo cumplirá el CAP si el valor para el rasgo AGR en el nodo VP raíz y el nodo VP hijo son el mismo.

### 2.5.3.- HFC (Head Feature Convention).

El objetivo del HFC es asegurar la unificación de los rasgos pertenecientes al subconjunto HEAD entre las categorías raíz y núcleo dentro de un árbol local.

Básicamente, el HFC establece que, para todo árbol local que cumpla el FFP y el CAP, todos los rasgos HEAD presentes en la raíz que sean libres en el núcleo del árbol (cf. 2.1) (es decir, que puedan pasar a formar parte de la categoría núcleo sin que haya conflictos de oposición de valores con rasgos ya existentes) deben aparecer en el núcleo, y viceversa: todas las especificaciones de rasgos HEAD que aparezcan en el núcleo del árbol y sean libres en la raíz, deben pasar a formar parte de ésta.

### 2.5.6.- Especificaciones de rasgos por defecto (FSD)

Las FSDs son mecanismos que dotan a las categorías de unas determinadas especificaciones de rasgos en caso que ningún otro mecanismo especifique lo contrario (ya sea otro valor que el marcado por la FSD, o bien la no aparición o aparición del rasgo en cuestión).

Por ejemplo, la siguiente FSD

(15): FSD:     -[CONJ]

establece que si es posible que una categoría carezca del rasgo CONJ (es decir, si ningún otro mecanismo de la teoría obliga a ello), el rasgo CONJ debe estar en efecto ausente de la categoría en cuestión.

### 3. Implementación de GPSG

Veremos aquí las formas de tratar computacionalmente las reglas y principios de la GPSG, y algunos de los problemas que plantean.

#### 3.1.- Reglas ID/LP

Hay dos posibilidades de tratar computacionalmente el reparto de trabajo entre reglas ID y reglas LP:

\* compilar una gramática objeto extendida, que exprese el efecto de combinar ambos tipos de generalización en el formato más familiar de las reglas libres de contexto (CF), que disponen de algoritmos de *parsing* probados y eficaces (Earley, CYK, etc).

Este punto de vista es el adoptado por Phillips y Thompson (1987) del Departamento de IA de la Universidad de Edimburgo. Su proceso de expansión es el siguiente:

- las reglas ID se aumentan por metarreglas
- las reglas ID resultantes se combinan (teniendo en cuenta las reglas LP) en sus posibles órdenes.
- el nuevo conjunto de reglas se completa por medio de las convenciones de instanciación de rasgos.

El problema de esta estrategia es la explosión combinatoria que se produce al compilar gramáticas grandes. Al tener que calcular todas las permutaciones de las reglas ID (filtrando algunas de ellas por las reglas LP), la expansión puede producir listas de miles de reglas CF.

\* El enfoque opuesto consiste en diseñar un *parser* que procese directa y simultáneamente las reglas ID y las reglas LP. Shieber (1983) ha adaptado con algunos retoques el algoritmo de Earley a esta nueva función. En este análisis, el algoritmo no genera todas las combinaciones de una regla ID, sino que examina las posibilidades un paso cada vez, a medida que las va necesitando.

La implementación de Evans y Gazdar (1984) sigue esta línea. Su técnica es compilar primero las reglas ID básicas, las metarreglas, y una versión simplificada de los principios HFC y CAP en un conjunto sencillo de reglas ID, dejando al *parser* la comprobación de las reglas LP y de las restantes convenciones de rasgos.

