

1. Recapitulación

Hasta ahora vimos algunas nociones básicas para formalizar de manera rigurosa las lenguas naturales:

- **Teoría de conjuntos y funciones:** Conjuntos como colecciones de elementos; función como objeto matemático que toma elementos de cierto dominio y devuelve un resultado; posibilidad de traducir todo conjunto como una función y viceversa
- **Cadena:** Una secuencia de símbolos tomados de un alfabeto Σ .
- **Lenguaje:** Conjunto de cadenas; posibilidad de entender las lenguas naturales como lenguajes.
- **Gramáticas (formales):** Sistema deductivo que permite derivar como teoremas todas las cadenas de un lenguaje en particular a partir de un conjunto de axiomas y de reglas de inferencia. Posee capacidad generativa débil (capacidad de generar cadenas) y, fundamentalmente, capacidad generativa fuerte (capacidad de generar estructuras).
- **Jerarquía de Chomsky:** Los lenguajes y sus gramáticas asociadas se pueden clasificar de acuerdo a su poder expresivo en distintas clases según la tabla 1
- **Costo computacional:** El problema del reconocimiento universal de cada gramática acarrea un costo de procesamiento diferente según de qué clase de gramática se trate, tal como también se especifica en la tabla 1
- **Gramáticas independientes de contexto:** Gramáticas que generan lenguajes independientes de contexto.

Bibliografía en la que está basada la clase de hoy:

- Bibliografía obligatoria

Lenguajes	Gramáticas	Tipo de tiempo
Lenguajes regulares	Gramáticas regulares	tiempo lineal
Lenguajes independientes de contexto	Gramáticas independientes de contexto	tiempo polinómico
Lenguajes sensibles al contexto	Gramáticas sensibles al contexto	tiempo exponencial (intratable)
Lenguajes irrestrictos	Gramáticas irrestrictas	indecidable

Tabla 1: Equivalencia entre lenguajes y gramáticas y costos de procesamiento en términos de tiempo para el parseo de una cadena

- Müller, S. (2016). *Grammatical Theory: From transformational grammar to constraint-based approaches*. Language Science Press, Berlin. Cap. 5. “Generalized Phrase Structure Grammar”, pp. 175-196.
- Gazdar, G. (1982). Phrase Structure Grammar. En Jacobson, P. y Pullum, G., editores, *The Nature of Syntactic Representation*, pp. 131–186. Reidel, Dordrecht
- Gazdar, G., Klein, E., Pullum, G., y Sag, I. (1985). *Generalized Phrase Structure Grammar*. Harvard University Press, Cambridge, Massachusetts
- Moreno Sandoval, A. (2001). *Gramáticas de Unificación y Rasgos*. Antonio Machado, Madrid. Cap. 6. “Gramática de Estructura Sintagmática Generalizada”, pp. 113-119.
- Bennet, P. (1995). *A Course in Generalized Phrase Structure Grammar*. University College London, London
- Ristad, E. S. (1987). Gpsg-recognition is np-hard. *Linguistic inquiry*, 18(3):530–536
- Ristad, E. S. (1986). Computational complexity of current gpsg theory. En *Proceedings of the 24th annual meeting on Association for Computational Linguistics*, pp. 30–39. Association for Computational Linguistics

- Ristad, E. S. (1990). Computational structure of gpsg models. *Linguistics and Philosophy*, 13(5):521–587

2. Motivaciones y contexto histórico

Generalized Phrase Structure Grammar (GPSG) es una teoría gramatical desarrollada por un conjunto de investigadores que pretendía desarrollar un marco teórico con las siguientes características:

- Que fuese descriptivamente adecuado
- que fuese una gramática explícita que pudiera ser implementada en computadora de modo eficiente

Para entender las motivaciones que hicieron que GPSG adoptara la forma que adoptó, hay que tener en cuenta algunas cuestiones del contexto histórico:

- Luego de fracasos en el intento de construir algoritmos de parsing para gramáticas transformacionales y del trabajo de [Peters y Ritchie \(1973\)](#) que demostraba que el modelo de [Chomsky \(1965\)](#) era indecidible, existía bastante escepticismo respecto de que una gramática en los lineamientos de la gramática generativa transformacional de aquellos años pudiera implementarse computacionalmente.
- Se había demostrado recientemente que las gramáticas independientes de contexto permiten parsear oraciones en tiempo polinómico, más precisamente, en $O(n^3)$ (ver [Valiant 1975](#) [Graham y Harrison 1976](#)). Incluso, [Earley \(1970\)](#) había propuesto un algoritmo que permitía parsear algunos lenguajes independientes de contexto en tiempo polinómico y otros en tiempo lineal.
- [Chomsky \(1957\)](#) había demostrado que las lenguas naturales no son lenguajes regulares, pero no había demostrado que no fueran lenguajes independientes de contexto. Su motivación para postular una gramática transformacional era conceptual. Para esta época no se había demostrado tajantemente que

las lenguas naturales no fueran lenguajes independientes de contexto.

Las conclusiones a las que arriban los propulsores de GPSG a partir de esto son las siguientes:

- Hay que armar una gramática que respete las convenciones de las gramáticas independientes de contexto, para de ese modo conservar la decidibilidad en $O(n^3)$.
- Hay que enriquecer esa gramática con algo que, sin aumentar su capacidad generativa, permita solucionar los problemas conceptuales señalados en [Chomsky \(1957\)](#).

2.1. Convenciones de las Gramáticas independientes de contexto

Como vimos en clases anteriores, las gramáticas se definen como cuádruplas de la forma $\langle NT, ST, SI, RP \rangle$ ¹, en la que los símbolos NT, T, O y P deben entenderse del siguiente modo:

- **NT = Símbolos no terminales o símbolos auxiliares:**
Los símbolos
- **T = Símbolos terminales:** Se los suele escribir en minúscula para distinguirlos de los símbolos no terminales.
- **O = Símbolo inicial:** Un símbolo no terminal que encabeza la estructura de la cadena a analizar tal que $O \in NT$.
- **P = Reglas de Producción:**

Ver Hasta acá, la definición de arriba aplica no solo a las gramáticas ejercicio independientes de contexto, sino también a todo tipo de gramática

4.1 formal. La característica definitoria de las gramáticas independientes de contexto está dada fundamentalmente por el tipo de reglas de reescritura que aceptan. Estas reglas pueden reescribir solo un símbolo no terminal del lado izquierdo y no poseen restricciones respecto de lo que pueda aparecer del lado derecho de la regla.

¹Según el autor, puede variar el orden de los elementos de la cuádrupla o la definición de los elementos.

Los autores ligados a GPSG interpretan entonces que las reglas de su gramática tienen que respetar esta restricción básica de que del lado izquierdo solo puede aparecer un no terminal.

2.2. Problemas conceptuales de las Gramáticas independientes de contexto

2.2.1. Problemas de linealidad

Como vimos, las reglas independientes de contexto permiten Ver asignar una estructura a una cadena. La estructura se puede repre- ejercicio sentar como un grafo acíclico dirigido (generalmente orientado de 4.2 arriba hacia abajo) en el que cada bifurcación se corresponde a una regla de reescritura en la que el elemento a la izquierda de la regla es representado por el nodo superior y los elementos a la derecha por los nodos inferiores. Estos árboles se construyen de tal modo que respetan el orden lineal de cada lado derecho de la regla y no permite que los arcos se crucen.

Ahora bien, existen al menos dos argumentos para preferir que las reglas independientes de contexto no codifiquen relaciones de precedencia lineal

- Existen lenguas que permiten órdenes alternativos de pala- Ver bras, y eso implicaría la necesidad de escribir distintas reglas ejercicio independientes de contexto para las mismas relaciones jerár- 4.3 quicas.
- Dos lenguas que se diferencien en si colocan el núcleo a la derecha o a la izquierda precisarían reglas independientes de contexto diferentes, aunque las relaciones jerárquicas entre los constituyentes fueran las mismas en las dos.

2.2.2. Problemas de parentesco

Otro problema de las gramáticas independientes de contexto son los problemas de parentesco (*relatedness*)

Un ejemplo típico es el de la pasiva:

- (1) a. Borges escribió *Ficciones*.
b. *Ficciones* fue escrito por Borges

Podemos generar una gramática que genere las dos oraciones:

- (2)
- a. $O \rightarrow \text{SNanim SVact}$
 - b. $O \rightarrow \text{SNinanim SVpas}$
 - c. $\text{SNanim} \rightarrow \text{Borges}$
 - d. $\text{SNinanim} \rightarrow \text{Ficciones}$
 - e. $\text{SVact} \rightarrow \text{V SNinanim}$
 - f. $\text{SVpas} \rightarrow \text{PerPas SP}$
 - g. $\text{V} \rightarrow \text{escribió}$
 - h. $\text{PerPas} \rightarrow \text{fue Part}$
 - i. $\text{Part} \rightarrow \text{escrito}$
 - j. $\text{SP} \rightarrow \text{P SNanim}$
 - k. $\text{P} \rightarrow \text{por}$

Sin embargo, este tipo de formalización pierde la intuición que tienen los hablantes de que, en líneas generales, la oración (1b) se corresponde con la versión pasiva de la oración (1a). Existe una relación de parentesco (*relatedness*) entre activas y pasivas que un gramática independiente de contexto es incapaz de reflejar. Por el contrario, una gramática transformacional como la de Chomsky (1957), explica esa relación de parentesco a partir de la apelación a la estructura profunda.

3. GPSG: el formalismo

GPSG es una de las primeras gramáticas que complementa la enunciación de reglas con la enunciación de principios generales que todas las reglas deben cumplir, reduciendo así el sistema axiomático, aunque incrementándole la ontología de reglas. El sistema pasa de tener solamente reglas de reescritura a tener en su lugar el siguiente tipo de reglas:

- reglas ID
- reglas LP

- metarreglas
- reglas léxicas
- postulados de significado
- rasgos y sus convenciones asociadas:
 - restricciones de coaparición de rasgos
 - especificación de rasgos por *default*).

3.1. Las reglas ID/LP

Para evitar el problema relacionado con la linealidad, en GPSG se separan las reglas que introducen las relaciones jerárquicas de las reglas que introducen las relaciones de precedencia lineal. De este modo, existen dos grandes tipos de reglas:

- Reglas ID (Reglas de dominancia inmediata o *ID rules*): establecen exclusivamente las relaciones de jerarquía.
- Reglas LP (Reglas de precedencia lineal o *LP rules*): restringen los órdenes posibles.

Las reglas ID se anotan separando los símbolos del lado derecho con comas.

(3) Reglas de Dominancia Inmediata:

- a. $S \rightarrow NP, VP$
- b. $NP \rightarrow N$
- c. $VP \rightarrow V, NP_{[a]}$
- d. $NP_{[a]} \rightarrow a, SN$
- e. $V \rightarrow AUX, VLEX$

Estas comas pretenden indicar que los elementos no se suceden el uno al otro, sino que simplemente son hijos de la categoría a la izquierda de la flecha.

Las reglas LP codifican la linealidad con el símbolo \prec que indica que lo que está a su izquierda debe aparecer a la izquierda de lo que aparezca a su derecha.

(4) Reglas de precedencia lineal

- a. $\text{NP} \prec \text{VP}$
- b. $\text{AUX} \prec \text{VLEX}$
- c. $a \prec \text{NP}$

¿Qué órdenes posibles da a lugar la combinación de las reglas ID de (3) y las reglas LP de (3)

3.2. El uso de rasgos y sus convenciones asociadas

En GPSG, los nodos no terminales ya no son símbolos primitivos, sino que son conjuntos formados por rasgos, es decir, pares de <atributo, valor>. Estos rasgos y sus posibles valores asociados incluyen los siguientes:

- $\text{CAT} \times \{\text{N}, \text{V}, \text{P}, \text{A}...\}^2$
- $\text{CASE} \times \{\text{ACC}, \text{NOM}\}$
- $\text{PER} \times \{1, 2, 3\}$
- $\text{PLU} \times \{+, -\}$
- $\text{BAR} \times \{0, 1, 2\}^3$

²En realidad, en GPSG las categorías se obtienen mediante la combinación de los valores positivo y negativo para los rasgos N y V.

³Müller (2016) utiliza 1, 2 y 3 como valores, puesto que sigue en este punto la notación de Uszkoreit (1987), que es una de las figuras principales de la implementación de GPSG para el alemán. La notación originaria, presente en Gazdar *et al.* (1985), es 0, 1 y 2.

- $\text{COMP} \times \{\text{que, si}\}$

Así, el nodo correspondiente a un nombre como *vaca*, en un árbol de derivación clásico se representa como en (5a). En GPSG se representa como en (5b)

- (5) a. Nodo correspondiente a *vaca* en una gramática independiente de contexto típica

$$\begin{array}{c} \text{N} \\ | \\ \text{vaca} \end{array}$$

- b. Nodo correspondiente a *vaca* en una gramática al estilo de GPSG

$$\begin{array}{c} \{ \langle \text{CAT}, \text{N} \rangle, \langle \text{PLU}, - \rangle, \langle \text{GEN}, \text{fem} \rangle, \langle \text{BAR}, 0 \rangle \dots \} \\ | \\ \text{vaca} \end{array}$$

Los valores para el rasgo BAR indican el nivel de X con barra. Se suelen anotar simplificadaamente como un superíndice, dando lugar a la equivalencia en (6):

$$(6) \quad \text{VP} \rightarrow \text{V}, \text{NP} = \text{V}^2 \rightarrow \text{V}, \text{N}^2$$

- Las proyecciones mínimas (y solo ellas) contienen un rasgo de subcategorización al que llaman SUBCAT. El valor de SUBCAT es un número.
- Las proyecciones intermedias (X^1) y máximas (X^2) tienen el rasgo de SUBCAT indefinido. Esto se escribe como $\sim[\text{SUBCAT}]$
- El rasgo SUBCAT, en una notación simplificada, se marca entre corchetes como un subíndice.
- En una regla de la forma “ $X \rightarrow \dots H \dots$ ”, H es el núcleo.
- La *Head Feature Convention* es un principio general que determina que la categoría de la proyección intermedia y de las proyecciones máximas deben coincidir con la del núcleo. De este modo, en una regla de la forma “ $\text{V}^2 \rightarrow \text{H}$ ”, $\text{H} = \text{V}$

- Cuando un rasgo tiene la forma $\langle X, +, - \rangle$ (i.e., es un rasgo binario), se lo puede simplificar como $+X$ o $-X$.

Ver El rasgo SUBCAT permite agrupar los distintos patrones de ejercicio subcategorización.

en 4.4

- $VP \rightarrow H_{[1]} (die)$
- $VP \rightarrow H_{[2]}, NP (love)$
- $VP \rightarrow H_{[3]}, NP, PP_{[to]} (give)$
- $VP \rightarrow H_{[4]}, NP, PP_{[for]} (buy)$
- $VP \rightarrow H_{[5]}, NP, NP (spare)$
- $VP \rightarrow H_{[6]}, NP, PP_{[+LOC]} (put)$
- $VP \rightarrow H_{[8]}, NP, S[FIN]$

Las entradas léxicas son cuádruplas formadas por una forma fonológica, un conjunto de rasgos, información de irregularidad morfológica e información semántica:

(7) $\langle weep, [[CAT\ V], [BAR\ 0], [SUBCAT\ 1]], wept, weep' \rangle$

A los fines de esta exposición, vamos a considerar solo los dos primeros elementos siguiendo la siguiente notación:

(8) $\langle weep, [[CAT\ V], [BAR\ 0], [SUBCAT\ 1]] \rangle$

3.3. Las metarreglas

Las metarreglas son funciones que mapean una regla ID en otra regla ID. Supongamos las oraciones de (9).

- (9) a. John bought the vegetables.
 b. Jim put the book on the table.
 c. Jim believed Bob to be the best.

Si nos concentramos en el VP, para generar estas estructuras se necesitan respectivamente las siguientes reglas ID

- (10) a. $VP \rightarrow H_{[2]}, NP$ (Regla ID para el VP de 9a)
 b. $VP \rightarrow H_{[6]}, PP_{[+LOC]}, NP$ (Regla ID para el VP de 9b)
 c. $VP \rightarrow H_{[17]}, VP_{[INF]}, NP$ (Regla ID para el VP de 9c)

Supongamos ahora que queremos dar cuenta de cómo se forma el VP en las siguientes oraciones:

- (11) a. The vegetables were bought by John.
 b. The book was put on the table by Jim.
 c. Bob was believed by Jim to be the best.

Para eso necesitaríamos recurrir a las siguientes reglas:

- (12) a. $VP[Pas] \rightarrow H_{[2]}, PP_{[BY]}$
 b. $VP[Pas] \rightarrow H_{[6]}, PP_{[+LOC]}, PP_{[BY]}$
 c. $VP[Pas] \rightarrow H_{[17]}, VP_{[INF]}, PP_{[BY]}$

Ahora, asumir estas reglas implica perder la relación de parentesco entre (11) y (9). Para dar cuenta de esto, se utiliza una metarregla como la siguiente:

- (13) **Metarregla pasiva:** $VP \rightarrow \alpha, NP_{[a]} \Rightarrow VP[Pas] \rightarrow \alpha, (PP_{[BY]})$

Ahora, ya no es necesario estipular las reglas de (12, puesto que estas se siguen automáticamente de combinar las reglas de (10 con la metarregla de (13).

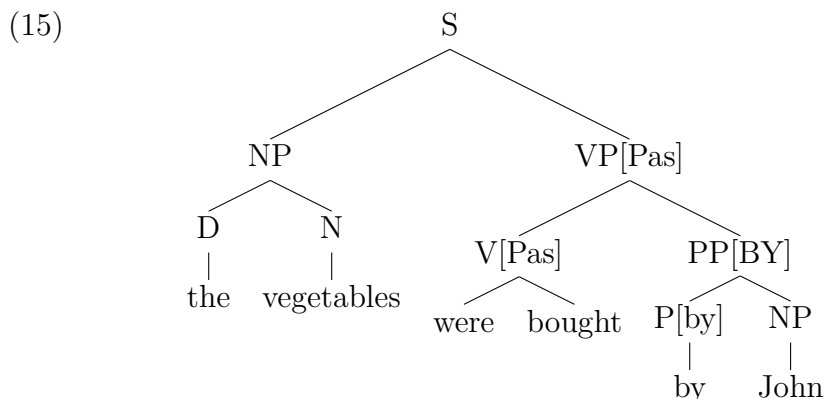
En Gazdar et al (1985) se reconocen 22 tipos de reglas de dominancia inmediata diferentes según el patrón de subcategorización del verbo. De estos 22, 10 son compatibles con la metarregla pasiva. en 4.5

3.4. Reglas léxicas

La metarregla nos permite reducir las reglas ID y establecer relaciones de parentesco entre reglas. Ahora bien, para introducir la forma activa o la forma pasiva del verbo, necesitamos entradas léxicas diferentes. Para resolver esto, GPSG recurre a reglas léxicas. Vamos a ver las reglas léxicas más en detalle más adelante en la

cursada. Por el momento, supongamos que una regla léxica actúa del siguiente modo:

- (14) Si α es un verbo activo, $P(\alpha)$ es una perífrasis pasiva formada por el verbo ser conjugado más el participio pasivo de α .



3.5. La semántica

GPSG asume una semántica en la línea de los trabajos de Montague. A diferencia de lo que vamos a ver cuando veamos implementaciones semánticas en gramáticas de rasgos, GPSG utiliza un sistema combinatorio orientado a teoría de tipos. La semántica orientada a tipos es la más frecuente en los trabajos de lingüística teórica, y la vamos a ver aplicada cuando lleguemos a la unidad de gramáticas categoriales. Por el momento, basta con ver que este tipo de abordajes semánticos utilizan recurren a modelos de las cosas que existen. El significado de las expresiones lingüísticas es su referencia, es decir, su correspondencia con elementos en estos modelos. Una teoría semántica involucra básicamente dos cuestiones:

1. especificar las denotaciones posibles para cada expresión de una categoría sintácticamente determinada, y
2. especificar la manera en la que las denotaciones de las expresiones complejas se producen en función de las denotaciones de sus constituyentes.

(Gazdar *et al.* 1985: 184)

El modelo está conformado por distintos tipos de cosas que se organizan en conjuntos. El modelo que asume GPSG consta de tres grandes tipos de elementos atómicos:

- Las entidades: Es el conjunto de todos los individuos, objetos y animales que existen en el mundo. Corresponde al tipo e.
- Los valores de verdad: Incluyen lo verdadero y lo falso. Corresponde al tipo t.
- Los mundos posibles: Es el conjunto de todos los modos en que las cosas pudieron ser. Permiten relativizar la verdad de las oraciones a determinados mundos (e.g., *Juan Peterson es un médium* es falsa en el mundo actual w_1 , pero verdadero en el mundo w_1 en que transcurre *Nuestra parte de noche* de Mariana Enríquez. Corresponde al tipo s.

En otros términos:

- (16) a. $D_e = \{x: x \text{ es una entidad existente en el modelo}\}$
 b. $D_t = \{0, 1\}$
 c. $D_s = \{w: w \text{ es un mundo posible}\}$

En su exposición, Müller (2016) obvia los mundos posibles. Vamos a hacer lo mismo.

A partir de los tipos simples, es posible formar recursivamente tipos complejos. Por ejemplo, un verbo transitivo es de tipo $\langle e, t \rangle$, porque es una función que toma una entidad y devuelve un valor verdad. GPSG, en una versión simplificada, asume los siguientes tipos para algunas de las expresiones lingüísticas:

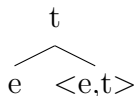
- (17) a. $TYP(S) = t$
 b. $TYP(NP) = \langle \langle e, t \rangle, t \rangle$
 c. $TYP(N') = \langle e, t \rangle$
 d. $TYP(Det) = \langle TYP(N'), TYP(NP) \rangle$
 e. $TYP(VP) = \langle e, t \rangle$

Existe una regla general para obtener una denotación a partir de las denotaciones de dos expresiones lingüísticas que se combinan: Aplicación funcional

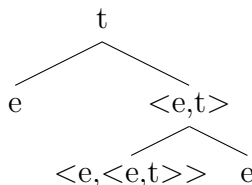
- (18) **Aplicación funcional:** Si α es de tipo $\langle b, a \rangle$ y β es de tipo b , entonces $\alpha(\beta)$ es de tipo a .

Ver ejercicio 4.6 Para poder aplicar esta regla de manera generalizada, GPSG estipula ciertas reglas que hacen que las reglas ID que no sean binarias, como la que reescribe a los verbos ditransitivos, acaben siendo binarias en la práctica. De este modo, al utilizar aplicación funcional, se obtienen árboles como los que se especifican a continuación.

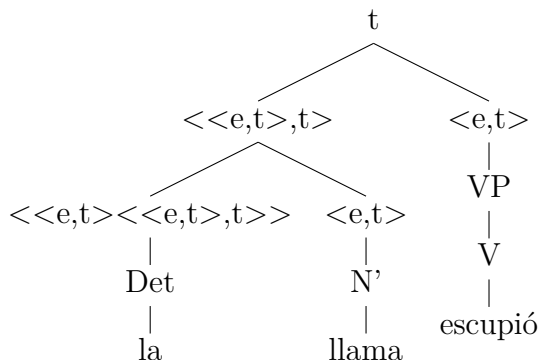
- (19) a.



- b.



- c.



De este modo, se usa teoría de conjuntos y funciones para modelar el significado de las expresiones lingüísticas y una regla básica que permite deducir el significado de una expresión compleja a partir de sus partes constituyentes, la regla de aplicación funcional en (18).

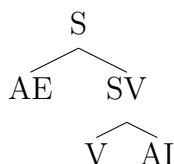
3.6. Los postulados de significado

El recurso de la metarregla permite relacionar dos formas. Pero no relaciona en sí mismo el significado para estas formas. Para eso vamos a agregar al sistema formal un mecanismo que se conoce como postulado de significado, que permite establecer sinonimia entre dos formas. Una versión absolutamente simplificada e informal del postulado de significado pasivo sería:

- The first nominal sister of a passive verb is interpreted as the last nominal argument of the active verb; the last nominal sister of a passive verb is interpreted as the first argument of the active verb. (Bennet 1995: 140)

En términos más formales, esto significa que si el verbo activo V es una función que toma dos entidades x e y en ese orden (siendo x el argumento interno e y el argumento externo, por lo tanto, más lejano, como se ve en (20)) y devuelve como resultado $V(x,y)$, en donde el primer elemento es el agente y el segundo el tema, la traducción semántica de V en $V[\text{Pas}]$ mediante este postulado de significa lo convierte en una función que ahora toma y y x en ese orden y devuelve $V[\text{Pas}](x,y)$.

(20)



3.7. Las dependencias no acotadas

Supongamos que queremos dar cuenta de la siguiente oración, que, como vimos, parece obedecer a una estructura que no puede ser generada por una gramática independiente de contexto.

(21) Qué dijiste que vio.

GPSG va a hacer uso de las herramientas que vimos hasta ahora más un principio adicional, el Principio del rasgo Foot (FFP por sus siglas en inglés), para dar cuenta de esta oración sin violentar

el formato de reglas esperado para una gramática independiente de contexto. Supongamos que tenemos la siguiente gramática:

- (22) Reglas para un fragmento de GPSG2
- a. $VP \rightarrow V[2], NP[-Q]$
 - b. $VP \rightarrow V[8], S[+FIN, <COMP \text{ que}>]$
 - c. $S[+FIN, <COMP \text{ que}>] \rightarrow \text{que}, S$
 - d. $S \rightarrow NP[-Q] VP$
- (23) Entradas léxicas para un fragmento de GPSG2
- a. $<\text{dijiste}, V[8]>$
 - b. $<\text{vio}, V[2]>$
 - c. $<\text{caballos}, NP[-Q, -Null]>$
 - d. $<\epsilon, N[+Q, +Null]>$
 - e. $<\text{qué}, NP[+Q, -Null]>$
 - f. $<\text{que}, C>$

Suponiendo ciertas reglas LP, esta gramática genera la oración (24):

- (24) a. Dijiste que vio caballos

Para generar dependencias cruzadas como la de (21), podemos utilizar una metarregla que convierte a toda regla de reescritura del siguiente modo⁴:

- (25) Metarregla para introducir preguntas parciales:
 $VP \rightarrow W, X \Rightarrow VP \rightarrow W, X[+Null]/X[+Q]$

W es, como en el caso de la metarregla pasiva, una variable. Por la HFC, debe incluir un H de categoría V. Esta metarregla nos

⁴En el modelo más estándar presentado en Gazdar *et al.* (1985), se utiliza un principio general en lugar de una metarregla. Vamos a seguir en este punto a Müller (2016), que retoma una metarregla que se había postulado en los inicios de la teoría

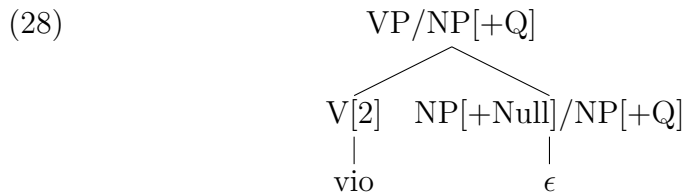
permite traducir cualquier regla de nuestra gramática en una regla válida con slash. Aplicado a (22a) da como resultado la siguiente equivalencia:

$$(26) \quad VP \rightarrow V[2], NP[-Q] \Rightarrow VP \rightarrow V[2], NP[+Null]/NP[+Q]$$

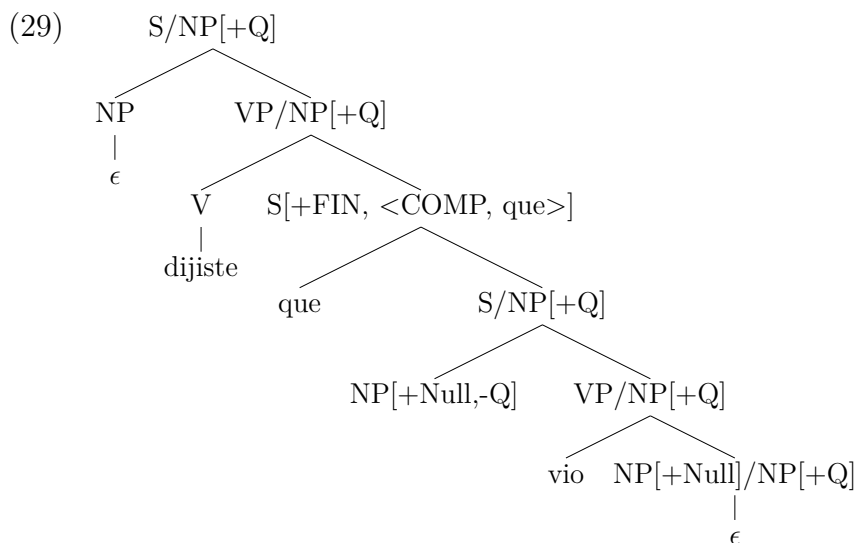
Un principio denominado Principio del rasgo Foot determina que el nodo madre hereda los nodos del subconjunto que GPSG denomina rasgos Foot. Estos incluyen, por ejemplo, el rasgo Slash. Por este principio, entonces la regla que se da como resultado de (26) se traduce en los siguientes términos:

$$(27) \quad VP/NP[+Q] \rightarrow V[2], NP[+Null]/NP[+Q]$$

Esto, combinado con las entradas léxicas de (23) da lugar a la siguiente estructura:



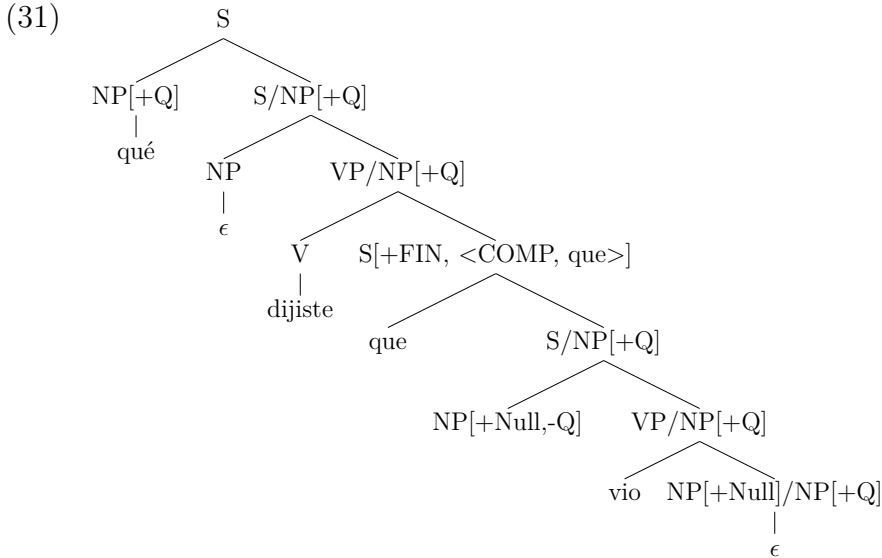
El Principio del rasgo Foot aplica hacia arriba en el árbol y da lugar a la siguiente estructura:



Supongamos ahora que tenemos una regla del siguiente tipo:

$$(30) \quad S \rightarrow X[+Q] \ S/X[+Q]$$

Al combinar esta regla con la derivación previa, se obtiene el siguiente árbol:



El rasgo Slash fue muy influyente en el campo de las gramáticas basadas en rasgos. De hecho, en unas clases vamos a ver una implementación de esto para una gramática enriquecida con rasgos en NLTK.

3.8. Sobre el poder expresivo de GPSG

GPSG, al conservar el formato de reglas de las gramáticas independientes y al contar con un número reducido de rasgos atómicos, es débilmente equivalente a una gramática independiente de contexto. Puesto que las gramáticas independientes de contexto no tienen el poder expresivo suficiente para dar cuenta del lenguaje natural, como se probó con posterioridad, eso hace que GPSG no sea una teoría descriptivamente adecuada.

Vimos en las clases previas que se sabe que es posible parsear una gramática independiente de contexto en tiempo polinómico.

Como GPSG es débilmente equivalente a las Gramáticas Independientes de Contexto, se asumió que conservaban también un bajo costo computacional. Sin embargo, se ha demostrado que esto no es cierto: GPSG es probablemente intratable. (ver Ristad 1986, 1987, 1990).

- En Ristad (1986) se muestra que la fuente de este excesivo costo computacional proviene del uso de categorías vacías y de la convención del rasgo Head. El hecho de que GPSG lleve tanto tiempo de procesamiento a pesar de ser equivalente a una gramática independiente de contexto tiene que ver con el tamaño de la gramática. Normalmente, el costo de procesamiento se mide solo en términos de la longitud de la cadena, pero en realidad, el tamaño de una gramática también lo afecta. GPSG es equivalente a una gramática independiente de contexto lo suficientemente grande para que el impacto en el costo computacional sea relevante.
- Por otro lado, Ristad (1990) muestra que el problema de la clausura finita de metarreglas (*i.e.*, el problema de determinar si una regla r pertenece al conjunto de las reglas formadas por la interacción de las metarreglas y las reglas ID de base) es demasiado complejo.
- Ristad (1990) también demuestra que pertenece es demasiado complejo el problema de reconocimiento universal para árboles locales no ordenados (*i.e.*, la posibilidad de decidir si una cadena x dada puede ser derivada de un conjunto de árboles locales no ordenados).

Estas características hicieron que GPSG terminara perdiendo su atractivo. Por otro lado, GPSG se implementó en el lenguaje de programación PC-PATR (<https://software.sil.org/pc-patr/>). Este lenguaje de programación carece de soporte desde hace ya muchos años.

4. Ejercicios

4.1. Reglas para gramática independientes de contexto

viene de ¿Cuáles de estas reglas pueden pertenecer a una Gramática In-
página 4 dependiente de contexto y cuáles no?

- (32) a. $X \rightarrow Z Y$
b. $X \rightarrow Z x w Y$
c. $X Z \rightarrow Y$
d. $X Z \rightarrow y$
e. $X a \rightarrow Y b a$
f. $X \rightarrow X$

4.2. Gramáticas independientes de contexto y linealidad

Viene de Supongamos las siguientes reglas de reescritura independientes
página 5 de contexto:

1. $S \rightarrow SN SV$
2. $SN \rightarrow N$
3. $SV \rightarrow V SN_{[a]}$
4. $SN_{[a]} \rightarrow a SN$
5. $V \rightarrow AUX VLEX$

¿Cuáles de los siguientes órdenes son posibles?

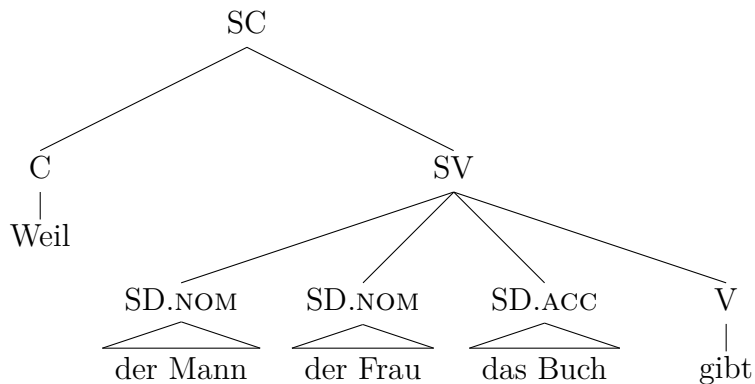
1. $AUX VLEX N a N$
2. $N AUX VLEX N a$
3. $N AUX VLEX a N$

Demuéstrelo proveyendo la derivación correspondiente.

En alemán, los argumentos en el *Mittelfeld* pueden aparecer en viene de cualquier orden, tal como se ve en (33). Este fenómeno se llama página 5 *local reordering*.

- Suponiendo el siguiente árbol para la oración (33a), escriba una gramática independiente de contexto que dé cuenta de todos los órdenes. A los fines del ejercicio tome las subcadenas resumidas con triángulos como si se tratasen de un nodo terminal indivisible.

(34)



¿Qué problemas encuentra en esta formalización del *local reordering*?

4.4. Un fragmento para GPSG (rasgo SUBCAT)

Viene de Elabore un fragmento de reglas de reescritura usando las convenciones de GPSG, particularmente la convención asociada al rasgo SUBCAT, que dé cuenta de los patrones de subcategorización en español que se ejemplifican en (35):

- (35) a. Marta se arrepentió de sus palabras.
b. Marta murió.
c. Maca le dio a Pablo el regalo.
d. Maca compró un regalo.

Elabore el léxico como un par ordenado para los verbos arrepentir, morir, dar y comprar.

Agregue a su vez la entrada léxica para un verbo adicional por cada patrón de subcategorización ejemplificado arriba:

4.5. Metarregla y reglas ID

Viene de Observen la metarregla pasiva en (13), que repetimos acá como
página (36)

11

(36) **Metarregla pasiva:** $VP \rightarrow \alpha, NP_{[a]} \Rightarrow VP[Pas] \rightarrow \alpha, (PP_{[BY]})$

Cuáles de las siguientes reglas ID son compatibles con esta metarregla.

- (37)
- a. $VP \rightarrow H[1]$
 - b. $VP \rightarrow H[2], NP$
 - c. $VP \rightarrow H[3], NP, PP[to]$
 - d. $VP \rightarrow H[4], NP, PP[/br]$
 - e. $VP \rightarrow H[5], NP, NP$
 - f. $VP \rightarrow H[6], NP, PP[+ LOC]$
 - g. $VP[+ AUX] \rightarrow H[7], XP[+ PRD]$
 - h. $VP \rightarrow H[8], NP, S[FIN]$
 - i. $VP \rightarrow H[9], (PP[to]), S[FIN]$
 - j. $VP \rightarrow H[10], S[BSE]$
 - k. $VP \rightarrow H[11], (PP[of]), S[BSE]$
 - l. $VP[INF, + AUX] \rightarrow H[12], VP[BSE]$

- m. $VP \rightarrow H[13], VP[INF]$
- n. $VP \rightarrow H[14], V2[INF, + NORM]$
- o. $VP \rightarrow H[15], VP[INF, + NORM]$

Ahora, tome en cuenta la metarregla de adjuntos:

$$(38) \quad VP \rightarrow W \Rightarrow VP \rightarrow W, AdvP^*$$

Cuáles de las siguientes reglas son compatibles con esa metarregla:

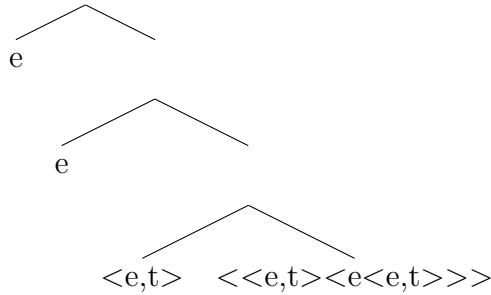
- (39)
- a. $VP \rightarrow H[1]$
 - b. $A^1 \rightarrow H[24], PP[about]$
 - c. $VP \rightarrow H[2], NP$
 - d. $N^1 \rightarrow H[30]$
 - e. $VP \rightarrow H[3], NP, PP[to]$
 - f. $A^1[AGR S] \rightarrow H[25], PP[to]$
 - g. $VP \rightarrow H[4], NP, PP[/br]$
 - h. $N^1 \rightarrow H[31], PP[wi7/»], PV[about]$
 - i. $N^1 \rightarrow H[32], S[COMP that]$
 - j. $VP \rightarrow H[5], NP, NP$
 - k. $A^1 \rightarrow H[26], S[FIN]$
 - l. $VP \rightarrow H[6], NP, PP[+ LOC]$

4.6. Aplicación funcional

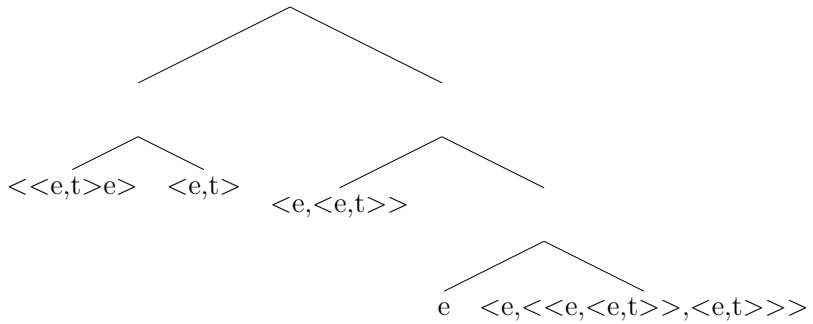
Determine el tipo que se obtiene en los nodos faltantes al utilizar aplicación funcional:

Viene de
la página
14

(40) a.



b.



Referencias

- Bennet, P. (1995). *A Course in Generalized Phrase Structure Grammar*. University College London, London.
- Chomsky, N. (1957). *Syntactic structures*. Mouton, The Hague. Manejamos la traducción de Carlos Peregrín Otero, Madrid: Siglo XXI, 1982.
- Chomsky, N. (1965). *Aspects of the theory of syntax*. M.I.T. Press, Cambridge, Massachusetts. Manejamos la traducción de Carlos Peregrín Otero, Madrid: Aguilar, 1971.
- Earley, J. (1970). An efficient context-free parsing algorithm. *Communication of the ACM*, 13(2).
- Gazdar, G. (1982). Phrase Structure Grammar. En Jacobson, P. y Pullum, G., editores, *The Nature of Syntactic Representation*, pp. 131–186. Reidel, Dordrecht.

- Gazdar, G., Klein, E., Pullum, G., y Sag, I. (1985). *Generalized Phrase Structure Grammar*. Harvard University Press, Cambridge, Massachusetts.
- Graham, S. L. y Harrison, M. A. (1976). Parsing of general Context-Free Languages. *Advances in Computers*, 14:77–185.
- Moreno Sandoval, A. (2001). *Gramáticas de Unificación y Rasgos*. Antonio Machado, Madrid.
- Müller, S. (2016). *Grammatical Theory: From transformational grammar to constraint-based approaches*. Language Science Press, Berlin.
- Peters, P. S. y Ritchie, R. W. (1973). On the generative power of transformational grammars. *Information Sciences*, 6:49–83.
- Ristad, E. S. (1986). Computational complexity of current gpsg theory. En *Proceedings of the 24th annual meeting on Association for Computational Linguistics*, pp. 30–39. Association for Computational Linguistics.
- Ristad, E. S. (1987). Gpsg-recognition is np-hard. *Linguistic inquiry*, 18(3):530–536.
- Ristad, E. S. (1990). Computational structure of gpsg models. *Linguistics and Philosophy*, 13(5):521–587.
- Uszkoreit, H. (1987). *Word order and constituent structure in German*. Center for the Study of Language and Information.
- Valiant, L. G. (1975). General context-free recognition in less than cubic time. *Journal of computer and system sciences*, 10:308–315.