

1. Introducción

Desde el marco de ciertas escuelas lingüísticas con una fuerte preocupación por la implementación computacional, la gramática generativa fue frecuentemente criticada por su poca rigurosidad formal y su poca explicitud:

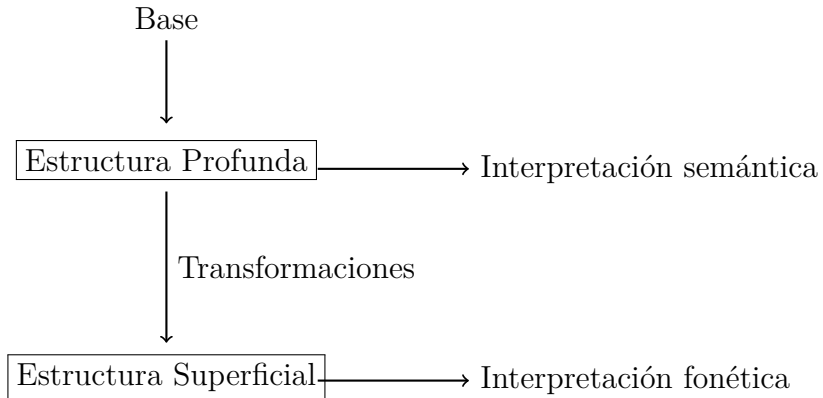
“some linguists also noted problems relating to the lack of clear, formal, and explicit descriptions within transformational grammar. With implicit reference to then current work in transformational grammar, Gazdar et al. (1985, ix) stated that one cannot “evade the entire enterprise of generative grammar by announcing ‘we assume some recursive function that assigns to each grammatical and meaningful sentence of English an appropriate structure and interpretation’. One must set about constructing such a function, or one is not in the business of theoretical linguistics”

Borsley y Börjars 2011: 2

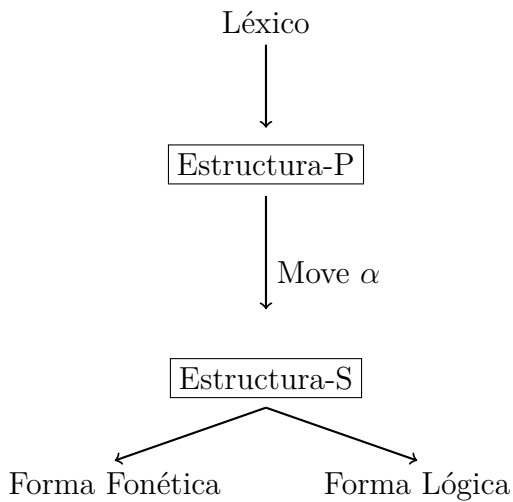
El carácter difuso de algunas definiciones y axiomas vuelve complicado conocer las propiedades matemáticas de las gramáticas minimalistas (e.g. tipos de lenguajes que generan, su complejidad en términos de la teoría de la complejidad). Sin embargo, existen algunos trabajos que intentan paliar ese problema mediante una axiomatización rigurosa de la teoría de modo tal de obtener un sistema capaz de ser implementado en computadora mediante parsers. Uno de estos autores es Stabler (2001, 2003, 2011), que ha desarrollado varios recursos computacionales para implementar la gramática minimalista, adaptando para eso las definiciones teóricas a definiciones operativas computacionalmente. En Stabler y Collins (2016) se hace un intento por hacer eso mismo pero manteniéndose lo más cerca posible de las formulaciones teóricas. En estos trabajos se desarrollan diversas gramáticas minimalistas, algunas de las cuales definen lenguajes diferentes: minimalist grammars, conflated minimalist grammars, relativized minimalist grammars, minimalist grammars with copying, minimalist grammars with head-movement, minimalist grammars for mirror theory.

2. Breve reseña histórica

Teoría Standard (Extendida)



2.1. Rección y Ligamiento



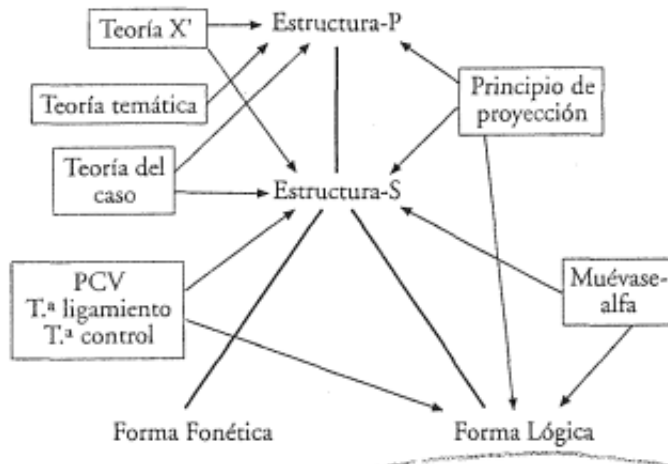
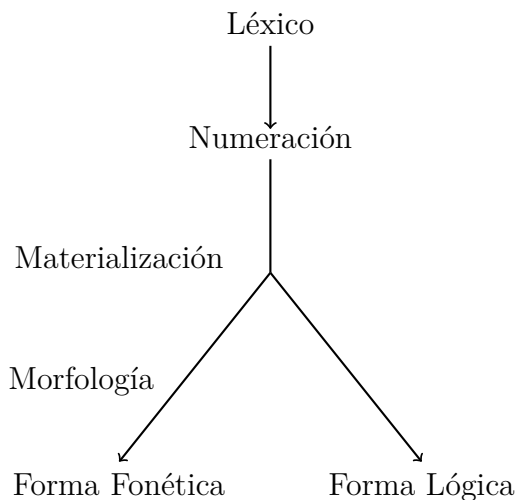


Figura 1: Lagunilla *et al.* (1995)

3. Programa Minimalista



3.1. Algunas definiciones previas

Collins y Stabler (2016) proponen una formalización explícita de la sintaxis minimalista. Para ello, primeramente presentan una serie de definiciones básicas:

- (1) **Gramática universal:** <PHON, SYN, SEM, Select, Mer-

- ge, Transfer>
- (2) **Item léxico:** <Sem, Syn, Phon>
 - (3) **Léxico:** {ÍtemLéxico1, ÍtemLéxico2..., ÍtemLéxico_n}
 - (4) **Lengua-I:** <Léxico, GramáticaUniversal>
 - (5) **Lexical Item Token:** <ÍL, n>
 - (6) **Lexical Array:** {LIT1, LIT2, ..., LIT_n} (un conjunto finito de Lexical Item Tokens)
 - (7) **Syntactic Object:**
 X es un objeto sintáctico ssi
 - I. X es un lexical item token, o
 - II. X es un conjunto de objetos sintácticos.
 - (8) **Estado de una derivación:** <LexicalArray, Workspace>
 - (9) **Raíz:** Para todo objeto sintáctico X y todo estado de derivación S=<LA, W>, si $X \in W$, X es raíz en W.
 - (10) **Selection:** Para todo estado de la derivación S=<LA, W>, si $\acute{I}L \in LA$, entonces $\text{Select}(\acute{I}L, S) = \langle LA - \{LI\}, W \cup \{LI\} \rangle$
 - (11) **Merge:** Para todo objeto sintáctico A y todo objeto sintáctico B, $\text{Merge}(A, B) = \{A, B\}$
 - (12) **Transfer:** Para un objeto SO con el $\text{Triggers}(SO) = \{ \quad \}$, $\text{Transfer}(P, SO) = \langle \text{Transfer}_{PF}(P, SO), \text{Transfer}_{LF}(P, SO) \rangle$.

Stabler (2011) se propone reseñar distintas propuestas de gramáticas minimalistas a partir de manipular distintos elementos:

- rasgos sintácticos constructores de estructura
- localidad
- linealización
- ensamble interno

Encuentra dos resultados

1. Propuestas diversas son matemáticamente equivalentes
2. Todos estos mecanismos generan familias abstractas de lenguaje que son reconocibles eficientemente.

Algunas de las gramáticas que reseña son la *Minimalist Grammar* (MG), la *Conflated Minimalist Grammar* (CMG), *Phase-based Minimalist Grammar* (PMG), entre otras. Aquí solo vamos a trabajar con la MG.

3.2. Minimalist Grammar

A diferencia de lo que ocurre en las gramáticas de estructura de frase, el poder generativo básico en las gramáticas minimalistas no descansa en reglas de reescritura sino en una operación que toma objetos sintácticos y devuelve un objeto sintáctico mayor: la operación de ensamble. Esta operación, asume Stabler, es disparada mediante rasgos de subcategorización y, como en la teoría de la X con barra, da lugar a estructuras endocéntricas. Uno de los aportes importantes de la teoría de la X con barra fue la endocentricidad. La endocentricidad permite distinguir qué propiedades son relevantes y cuáles deben seguir siendo visibles en la derivación. Esto se codifica en las etiquetas:

- (13) **Ensamble en términos de la teoría de conjuntos:**

$$\alpha, \beta \rightarrow \{\alpha, \{\alpha, \beta\}\}$$

- (14) **Ensamble en términos de árbol sintáctico:**

$$\begin{array}{c} \alpha \\ \wedge \\ \alpha \quad \beta \end{array}$$

Stabler utiliza una representación alternativa que llama *bare tree* en la que en lugar de la etiqueta usa un símbolo < o > que señala si el núcleo está abajo a la izquierda o abajo a la derecha respectivamente.

- (15)

$$\begin{array}{c} < \\ \wedge \\ \alpha \quad \beta \end{array}$$

Ver
ejercicio
4.1

Para poder implementar su teoría, Stabler asume que los ítems léxicos constan de rasgos fonéticos y semánticos seguidos de rasgos sintácticos.

- (16) Phon :: feature1, feature2...

En una estructura derivada se reemplaza :: por :
Existen cuatro tipos de rasgos

- Rasgos categoriales: D, N, V...
- Rasgos de selección: =D, =N, =V...
- Rasgos de licenciados o meta: -wh, -focus, -case
- Rasgos de licenciadores o sonda: +wh, +focus, +case

El primer ingrediente para una gramática Minimalista es entonces su léxico:

$$(17) \text{ Lex} \subset \text{Phon} \times \text{Features}^*$$

Estos ítems léxicos son combinados por merge [ensamble]. Revisemos como opera (external) merge [em] ante de introducir su definición formal. Cuando un elemento A tiene el rasgo de selección α -i.e., = α - y un elemento B tiene α como rasgo categorial, A y B se ensamblan y los rasgos de selección se borran. Consideremos la secuencia de (18).¹

$$(18) \text{ Ana saw Mary}$$

Nuestro léxico contiene al menos los siguientes tres items, dos de ellos tienen al menos un rasgo categorial y uno tiene además dos rasgos de selección:

$$(19) \begin{array}{l} \text{a. Ana} :: \text{D} \\ \text{b. Mary} :: \text{D} \\ \text{c. saw} :: =\text{D}, =\text{D}, \text{V} \end{array}$$

La formación de la estructura sigue los siguientes pasos: merge toma los items, [saw :: =D, =D, V] y [Mary :: D] y da como resultado la estructura de (20b):

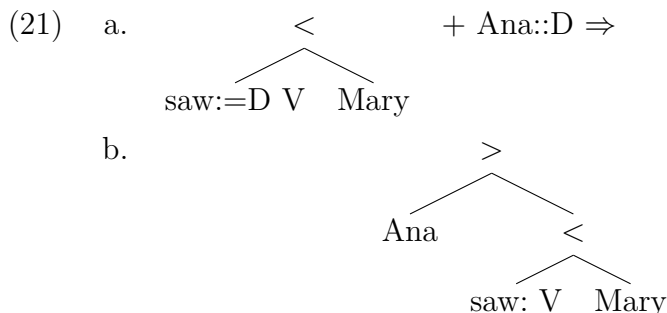
$$(20) \begin{array}{l} \text{a. saw} :: =\text{D}, =\text{D}, \text{V} + \text{Mary} :: \text{D} \Rightarrow \\ \text{b.} \end{array}$$

$$\begin{array}{c} < \\ \swarrow \quad \searrow \\ \text{saw} :=\text{D} \quad \text{V} \quad \text{Mary} \end{array}$$

¹Empleamos un ejemplo del inglés en este caso, para evitar algunas complicaciones con el marcado de los objetos en español

Luego, se introduce el item $[Ana :: D]$ que satisface el rasgo de selección $[=D]$ de V .

Ver
ejercicio
4.2




Stabler define formalmente la función de external merge [ensamble externo] como en (22):

Ver
ejercicio
4.3

$$(22) \quad \text{em}(t_{1[=x]}, t_{2[x]}) = \begin{cases} \begin{array}{c} < \\ \swarrow \quad \searrow \\ t_1 \quad t_2 \end{array} & \text{if } |t_1|=1 \\ \begin{array}{c} > \\ \swarrow \quad \searrow \\ t_2 \quad t_1 \end{array} & \text{otherwise} \end{cases}$$

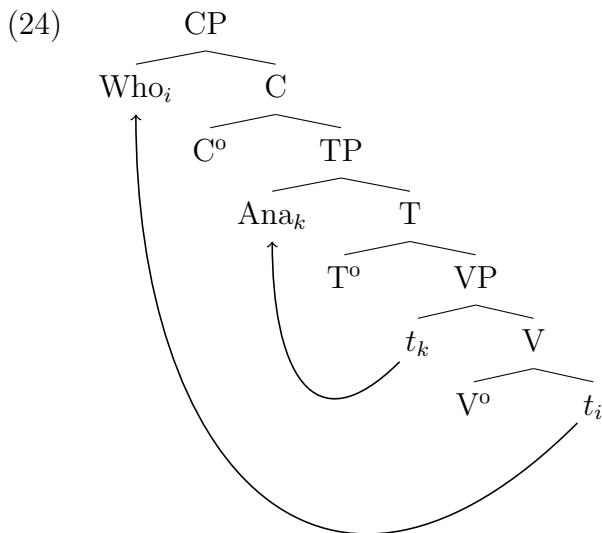
$|t|$ = el número de nodos en t .

(Stabler 2011: 5)

 Para ver una implementación de una gramática con merge y rasgos de selección categorial, revisemos la gramática `g0spSV0.pl` en la implementación de prolog de Stabler.

Como se puede apreciar, esta función construye un nuevo árbol a partir de dos árboles independientes. No obstante, resulta más o menos evidente que no podría derivar una estructura que involucre el movimiento de una frase, como sucede con los movimientos-A –asociados a dependencias argumentales, caso, concordancia– o los movimientos-A’ –interrogación, focalización, entre otros fenómenos. En efecto, la pregunta de (23) requiere, al menos, el movimiento del sujeto a Espec,TP y del objeto a Espec,CP como en (24).

(23) Who did Ana see?



Para poder derivar estas estructuras, es necesario enriquecer los ítems léxicos con rasgos de *licenciamiento*, a saber:

- Rasgos licenciados o meta: $-f$ [$-wh$, $-focus$, $-case$, ...]
- Rasgos licenciadores o sonda: $+f$ [$+wh$, $+focus$, $+case$, ...]

En el marco del Programa Minimalista se propone que las operaciones sintácticas deben obedecer ciertos principios de economía. En lo que respecta al movimiento, la discusión se centra en el punto de si el movimiento tiene que tener la menor cantidad de pasos posible o si debe ser lo más corto posible. Nótese que, aparentemente, ambas condiciones parecen ser igualmente válidas, pero suponen una paradoja: el movimiento más corto implica una mayor cantidad de pasos, mientras que la menor cantidad de pasos supone movimientos más largos. Independientemente de cuál es la solución de Chomsky (1995), la condición de movimiento más corto resulta central. De hecho, la gramática minimalista que aquí estamos estudiando adopta la versión más fuerte de esta restricción, que es la que se expresa en (25)

(25) *Shortest Move Constraint* [SMC]

Exactamente un núcleo en el árbol tiene $-x$ como su primer rasgo.

(Stabler, 2011)

La función que permite derivar una estructura como la de (23) –i.e., *internal merge* [im]– se define como en (26):

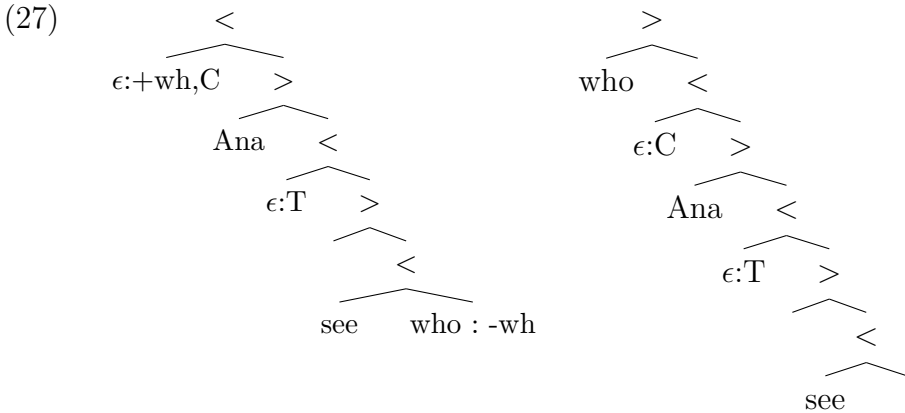
$$(26) \quad \text{im}(t_{1[+x]}) = \begin{array}{c} > \\ \swarrow \quad \searrow \\ t_2^M \quad t_{1\{t_2[-x]^M \rightarrow \varepsilon\}} \end{array} \quad \text{si SMC}$$

t^M = la proyección máxima de un núcleo t .

Para facilitar la comprensión, conviene traducir esta formalización. (26) expresa la idea de que:

1. Si $t_{[+f]}$ es un árbol que tiene un núcleo cuyo primer rasgo es $[+f]$, y
2. si ese árbol tiene un subárbol $t_2[-f]$ cuyo primer rasgo es $[-f]$, (y es el único núcleo que tiene $[-f]$ como su primer rasgo),
3. entonces el resultado de aplicar *internal merge* es el árbol que tiene a la proyección máxima de t_2 –i.e. t_2^M – como su especificador, con un árbol hermano que es el resultado de reemplazar t_2^M por el nodo vacío.

Veamos la derivación simplificada en la forma de *bare tree*:



Para ilustrar formalización de esta función, tomemos la oración de (28), que no es inglés sino *falso inglés*:

(28) Who Ana saw?

Así, los ítems léxicos para (28) se representan de la siguiente manera:

- (29) a. Ana :: D
b. Who :: -wh, D
c. saw :: =D, =D, V
d. ϵ :: =V, +wh, C

Nótese que el orden de los rasgos no es libre. En efecto, luego de ::, los rasgos deben ser ordenados, en general, según el siguiente

Ver criterio:
ejercicio

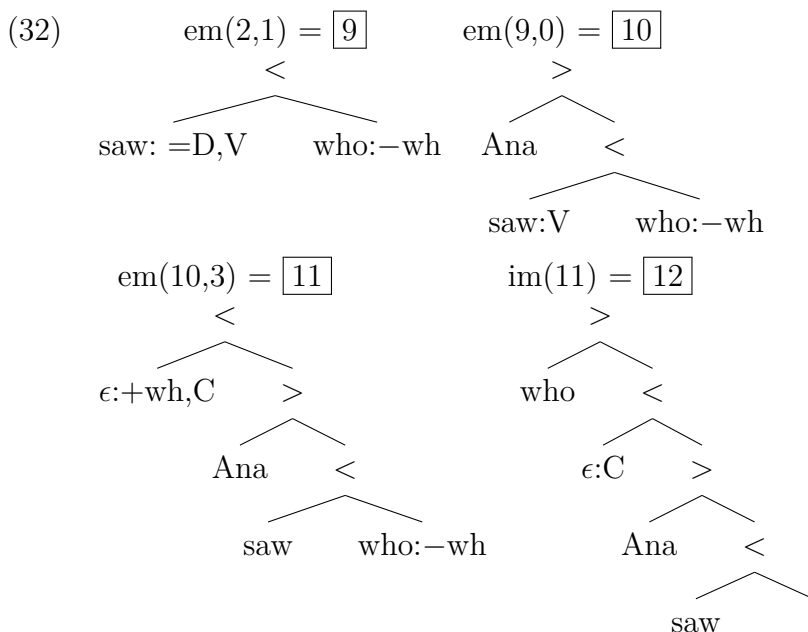
4.5

- (30) =X, \pm x, X
SEL. LIC. CAT.

Veamos, entonces, cómo se deriva la oración del *falso inglés* en (28). Supongamos que nuestro lexicón contiene, al menos, los ítems que aparecen a continuación, a los que le agregamos un número recuadrado \boxed{n} por conveniencia expositiva.


- (31) a. $\boxed{0}$ Ana :: D
b. $\boxed{1}$ Who :: -wh, D
c. $\boxed{2}$ saw :: =D, =D, V
d. $\boxed{3}$ ϵ :: =V, +wh, C
e. $\boxed{4}$ dog :: N
f. $\boxed{5}$ the :: =D, N
g. $\boxed{6}$ which :: =D, -wh, N
h. $\boxed{7}$ Marie :: D
i. $\boxed{8}$ runs :: =D, V

La derivación de esta oración implica la aplicación de tres operaciones de em y una aplicación de im.



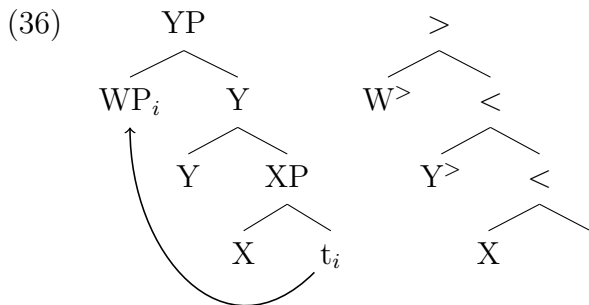
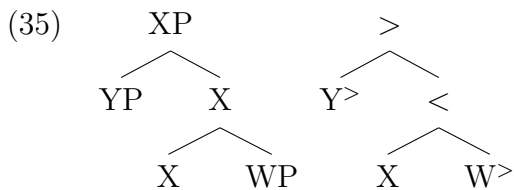
La operación de *im* que aquí se propone involucra la teoría de cotejo (chequeo) de rasgos propuesta en Chomsky (1995). Collins (2001) formula el cotejo como en (33) y (34):

- (33) Move raises α to the checking domain of a head H with a feature F only if the feature F of H enters into a checking relation with a feature F of α .
- (34) F1 and F2 enter a checking relation iff F2 is in the checking domain of F1 and F1 is deleted (F2 may also be deleted).

 Revisemos la gramática `g0.pl` de la implementación [Stabler](#) Ver para ver el funcionamiento de la gramática con rasgos de li- ejercicio cenciamiento $\pm wh$ y $\pm k$ 44

¿Esta gramática presenta alguna restricción en cuanto al orden?

Es interesante notar que las implementaciones que estamos revisando construyen estructuras con núcleo inicial *–i.e., núcleo a la izquierda–*, como (35). Las estructuras de núcleo final *–i.e., núcleo a la derecha–* son extructuras derivadas (39):



Más allá de la discusión en torno de si la sintaxis determina el lineal (véase [Kayne, 1994, 2011](#)), en estas gramáticas los movimientos de licenciamiento colocan necesariamente a los constituyentes desplazados en una posición en que preceden al núcleo. Las estructuras de (39) tienen el orden lineal de (37).

$$(37) \quad W \frown Y \frown X$$

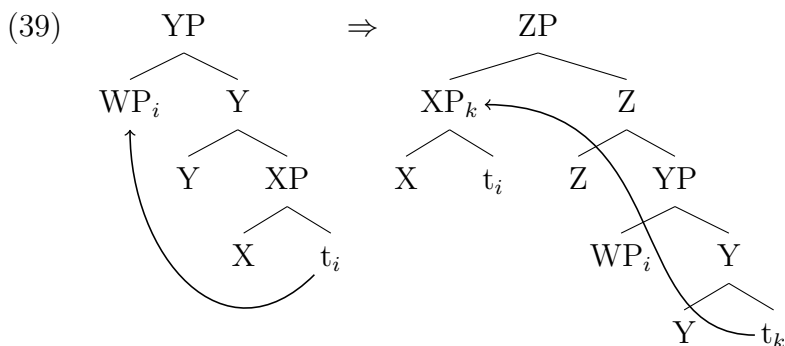
En pocas palabras, si el objeto se mueve para cotejar y eliminar su rasgos de caso *—i.e., —k, —case, etc.—*, entonces siempre se genera la estructura cuya secuencia lineal es la de (38).

$$(38) \quad (\text{Sujeto}) \frown \text{Objeto} \frown \text{Verbo}$$

Si las gramáticas que estamos revisando no emplean movimiento nuclear, la pregunta que surge es cómo se obtienen, entonces, estructuras SVO.

¿Cómo se obtienen estructuras SVO?


Estas gramática adoptan de manera generalizada un tipo de movimiento conocido como movimiento de remanente [*remnant movement*]. En términos sencillos, este tipo de movimiento consiste en el movimiento de una porción de la estructura oracional después de haber extraído un constituyente de su interior.



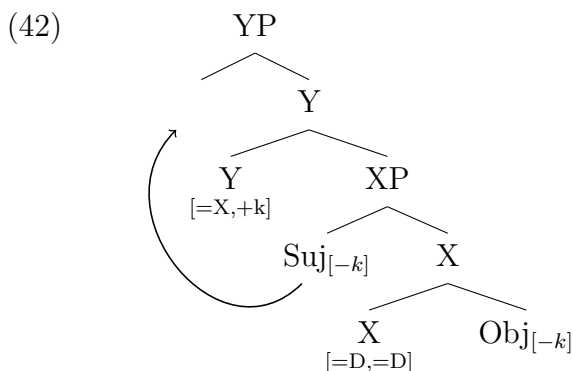
Por ejemplo, este tipo de movimiento propone Ordóñez (2012) para derivar la estructura de (40):

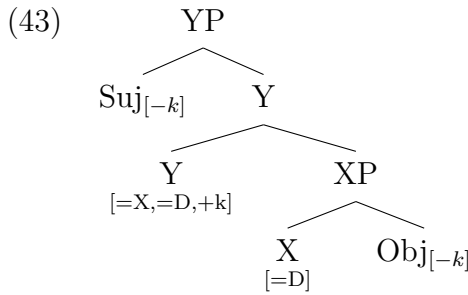
(40) Han hecho a Juan comerla.


- (41) a. $[[_{CP} \text{ han hecho } [_{CP} \text{ Juan } [_{SV} \text{ comerla}]]]]$
 b. $[[_{SV} \text{ comerla}]_j [[_{CP} \text{ han hecho } [_{CP} \text{ Juan } t_j]]]$
 c. $[\text{Juan}_i [[_{SV} \text{ comerla}]_j [[_{CP} \text{ han hecho } [_{CP} t_i t_j]]]]$
 d. $[[_{CP} \text{ han hecho } [_{CP} t_i t_j]]_k [\text{Juan}_i [[_{SV} \text{ comerla}]_j t_k]]$

 Revisemos la gramática **g-ne.pl** de [Stabler](#) para ver el funcionamiento de la gramática con rasgos de licenciamiento $\pm wh$ y $\pm k$

Siguiendo la estructura de **g-ne.pl**, elaboramos la gramática **g-nSP.pl**, que permite derivar algunas oraciones del español. Un punto importante para tener en cuenta es que estas gramáticas contemplan el licenciamiento de caso, es decir que los argumentos tienen que portar el rasgo $[-k]$ o $[-case]$. ¿Cómo se evita en estas gramáticas los siguientes conflictos potenciales?






 Revisemos qué alternativa encuentra la gramática `g0sp-caso.pl` para evitar estos conflictos.

3.3. Lenguaje $a^n b^n c^n$ ($n \geq 1$)


Context-Sensitive Grammar El lenguaje $a^n b^n c^n$ puede ser generado por una gramática sensible al contexto como la siguiente:

- (44)
- I. $S \rightarrow aAbc \mid abc$
 - II. $Ab \rightarrow aAb \mid aB$
 - III. $Bb \rightarrow bB$
 - IV. $Bc \rightarrow bcc$

- (45)
- | | |
|----------------|-----|
| 1. S | |
| 2. $aAbc$ | Ia |
| 3. $aaABbc$ | IIa |
| 4. $aaAbBc$ | III |
| 5. $aaAbbccc$ | IV |
| 6. $aaaBbbccc$ | IIb |
| 7. $aaabBbcc$ | III |
| 8. $aaabbBcc$ | III |
| 9. $aaabbbccc$ | IV |

 Veamos ahora la gramática `anbncn.pl` desarrollada por [Stabler](#).

3.4. Otra implementación:

 Veamos la implementación de python desarrollada por Alex Warstadt para *A Formalization of Minimalist Syntax* (Collins y Stabler, 2016).

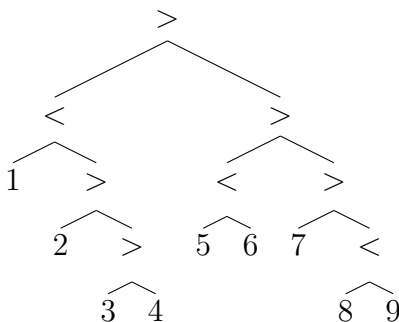
4. Ejercicios

4.1. Ejercicio

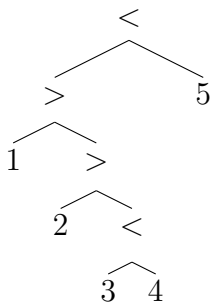
Determine cuál es el núcleo en estos árboles.

Viene de
página 5

(46)



(47)



4.2. Ejercicio

Construya las entradas léxicas que permitan derivar la siguiente oración (puede omitir los núcleos T y C):

Viene de
página 7

(48) La estudiante escribió una gramática.

4.3. Ejercicio

Viene de Derive a partir de la siguiente gramática la oración *Fede lee*
página 7 *Ficciones*.

- (49)
- I. $\text{Fede}::D$
 - II. $\text{Ficciones}::D$
 - III. $\text{lee}::=D\ V$
 - IV. $\epsilon::=V\ =D\ v$
 - V. $\epsilon::=v\ C$

4.4. Ejercicio

Viene de Revisemos la gramática `g0spanish.pl`. Construyamos los ítems
página 11 léxicos relevantes para poder derivar las siguiente oraciones:

- (50)
- a. El ratón el queso encontró.
 - b. Un gato la pared trepó.

4.5. Ejercicio

¿Qué saldría mal si los rasgos no estuvieran ordenados como en Viene de
página 10
(30)?

4.6. Ejercicio

La gramática g0.pl de Stabler genera oraciones con el orden básico de palabras SOV. Revise la gramática y algunas derivaciones posibles. ¿Qué impide generar estructuras con licenciamiento con el orden SVO? Viene de
página 18

4.7. Ejercicio

Viene de Observe el siguiente léxico:
página 18

- (51)
- a. [0] Ana :: D
 - b. [1] Who :: -wh, D
 - c. [2] saw :: =D, =D, V
 - d. [3] ϵ :: =V, +wh, C
 - e. [4] dog :: N
 - f. [5] the :: =D, N
 - g. [6] which :: =D, N
 - h. [7] Marie :: D
 - i. [8] runs :: =D, V

Referencias

- Borsley, R. y Börjars, K. (2011). Introduction. En *Non-Transformational Syntax*, pp. 1–8. Wiley Blackwell, Massachusetts.
- Chomsky, N. (1995). *The minimalist program*. MIT Press, Cambridge, Massachusetts. Manejamos la traducción de Juan Romero Morales, Madrid: Alianza, 1999.
- Collins, C. y Stabler, E. (2016). A formalization of minimalist syntax. *Syntax*, 19(1):43–78.
- Gazdar, G., Klein, E., Pullum, G., y Sag, I. (1985). *Generalized Phrase Structure Grammar*. Harvard University Press, Cambridge, Massachusetts.
- Kayne, R. S. (1994). *The antisymmetry of syntax*, volumen 25. MIT press.
- Kayne, R. S. (2011). Why are there no directionality parameters. En *Proceedings of WCCFL*, volumen 28, pp. 1–23.
- Lagunilla, M. F., Rebollo, A. A., Gonzalez, J. M. I., Carmona, M. B., y Barreto, V. D. (1995). *Sintaxis y cognición: introducción al conocimiento, el procesamiento y los déficits sintácticos*. Síntesis.
- Stabler, E. P. (2001). Recognizing head movement. En de Grote, P., Morril, G., y Retoré, C., editores, *Logical Aspects of Computational Linguistics*, pp. 245–260.
- Stabler, E. P. (2003). Comparing 3 perspectives on head movement. En Mahajan, A., editor, *From Head Movement And Syntactic Theory: UCLA, Postdam Working Papers in Linguistics*, pp. 178–198. UCLA.
- Stabler, E. P. (2011). Computational perspectives on minimalism. En Boeckx, C., editor, *Oxford handbook of linguistic minimalism*, pp. 617–643. Oxford.