

Phase Theory

Ángel Gallego

Handout/resumen por Pablo Zdrojewski

1. Introducción

La gramática generativa propone que el lenguaje es un componente de la *mente/cerebro* humana. Esto es lo que se conoce como *Facultad del Lenguaje* (FL). La FL está regulada por tres factores:

- (1) *Factores en el diseño de la FL*
 - a. Dotación genética
 - b. Experiencia
 - c. Principios no específicos de la FL

Desde la perspectiva de P&P la atención estaba centrada en los factores (1a) y (1b), es decir entre la adecuación “descriptiva” y la adecuación “explicativa”.

Chomsky observa que dentro del marco de P&P no solo se avanza en la solución de la tensión entre adecuación descriptiva y adecuación explicativa, sino que también se desarrolló el escenario para la investigación del tercer factor: principios de procesamiento de información, de arquitectura estructural y de eficiencia computacional independientes del lenguaje.

El minimalismo no supone indagar en la hipótesis de que la FL es máximamente eficiente desde un punto de vista metodológico, sino desde un punto de vista ontológico.

Minimalismo ontológico: no se pregunta cuán elegante es la teoría de la FL, sino cuán elegante es la FL en sí misma.

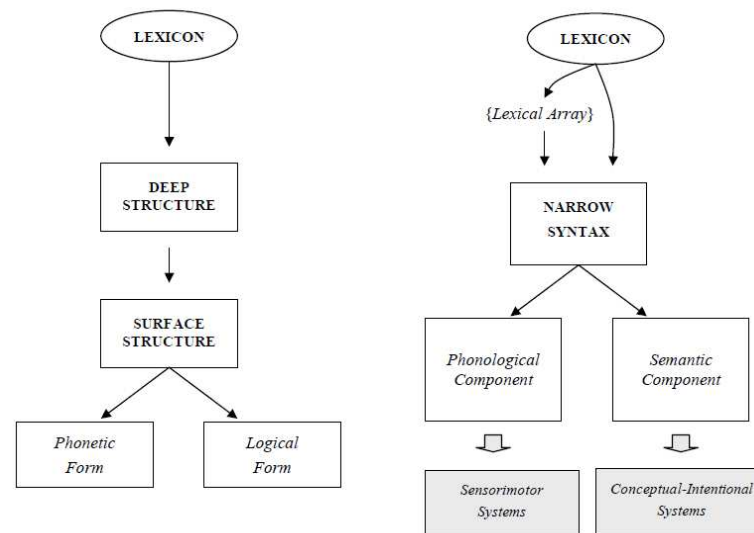
Strongest Minimalist Thesis

- (2) Language is an optimal solution to legibility conditions.

El minimalismo busca mostrar que los principios básicos del lenguaje pueden ser formulados en términos de nociones requeridas por requerimientos de necesidad conceptual.

(3a) *Government & Binding Y-Model*
(Non-Single Cycle Model)

(3b) *Minimalist Program Y-Model*
(Single Cycle Model)



La asimetría central entre la arquitectura de la gramática de GB y del minimalismo es la eliminación de niveles internos de representación DS, SS, LF.

- DS: un nivel de representación en el que se establecen las relaciones temáticas y de X-barras, previo a la aplicación de las operaciones transformacionales.
 - SS: un nivel de representación que constituye el resultado visible de la manipulación de las configuraciones iniciales.
 - En su conjunto DS, SS y LF son eliminados de la gramática como niveles de representación y son redefinidos como componentes. LF deja de existir como un sistema simbólico con propiedades y operaciones formales particulares, y sobrevive simplemente como un componente (una parte sustantiva del sistema que no tiene una clase particular de objetos designados).
- El minimalismo permite tomar seriamente la idea de que los fenómenos lingüísticos están regulados por requerimientos externos de los componentes FON y SEM, es decir, los sistemas de interfaz entre la Sintaxis (Estrecha) (Narrow Syntax) y los sistemas externos.
- Permite determinar si propiedades particulares se siguen de principios generales (no específicos del lenguaje), o de un mecanismo interno que no parece tener una contraparte en el mundo natural (i.e. la recepción, índices, etc).
- La SMT refuerza la idea de que los sistemas externos tienen propiedades por sí mismos, propiedades con las que la sintaxis debe interactuar y en última instancia satisfacer.

➤ Las operaciones computacionales tienen un papel en las interfaces.

(4) *The C-I Hypothesis*
 C-I incorporates a dual semantics, with generalized argument structure as one component, the other one being discourse-related and scopal properties. Language seeks to satisfy the duality in the optimal way [see 2], [external Merge] serving one function and [internal Merge] the other [from Chomsky 2008: 141]

2. Operaciones Computacionales

Una lengua L consiste básicamente en dos componentes principales: el Léxico y la Sintaxis (Narrow Syntax).

➤ La FL especifica un conjunto de rasgos disponibles para cada lengua particular L que toma de un inventario universal de rasgos F_{UG}. El factor de diseño (1b) realiza una selección de F_{UG} para formar F_{PG}, un inventario de rasgos de una lengua particular. Estos rasgos luego son montados en un Léxico particular, una colección de Ítems Léxicos (LI). No obstante, no todos los rasgos pueden ser definidos como ítems léxicos

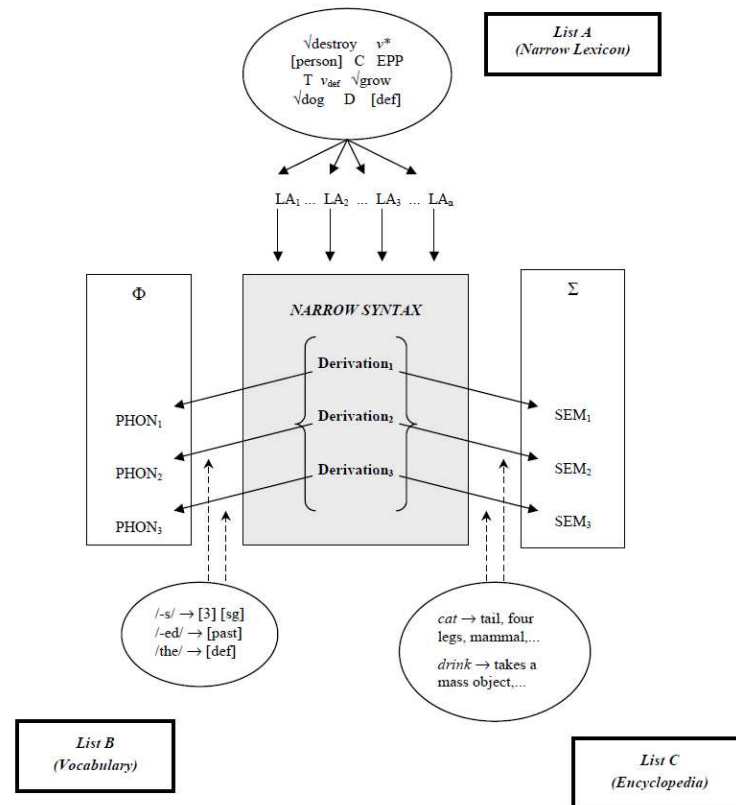
(5) *Lexicon Assumption*
 Some features from F_{UG} become LIs (categorial ones), others are just assembled within LIs, not being manipulated by the basic computational operation of Merge.

Dado el supuesto de (5), un Léxico consiste de LIs y de rasgos, a los que se puede acceder sin restricciones. No obstante, las operaciones que forman estructura no pueden tratar a los rasgos como lo hacen con los LIs. La sintaxis toma los LIs como unidades atómicas formadas por diferentes propiedades/rasgos:

- (6) *Propiedades de los LIs*
- a. Fonológicas
 - b. Semánticas
 - c. Formales

La lista de (6) es consistente con la afirmación de Chomsky (1995) de que los LI contienen información que no puede ser derivada de principios independientes: propiedades idiosincrásicas (excepciones). Esta visión es consistente con un *Lexicalismo Fuerte*, y se separa de una visión como la de la Morfología Distribuida (Halle & Marantz 1993) que abandona la idea tradicional de un Léxico y distribuye la información en (6) en diferentes listas:

(7) *Distributed Lexicon*



De acuerdo con la MD, la sintaxis utiliza elementos de la Lista A, raíces y rasgos morfosintácticos abstractos. Idea que es consistente con un sistema como el de Chomsky (1995; 2000; 2001) en el que los rasgos formales activan la sintaxis y en el que no es obvio que los rasgos fonológicos desempeñen algún papel durante la computación. Gallego adopta el supuesto de la sintaxis solo se preocupa por los rasgos formales.

2.1. Merge

Merge [ensamble] es la operación fundamental dentro del marco minimalista. Esta operación es la encargada de construir la estructura.

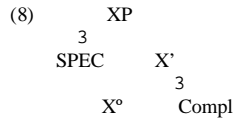
Discusión:

- ¿Cuál es el estatus de las etiquetas [Labels]?
- ¿La aplicación de Merge es parasitaria de que los objetos involucrados compartan rasgos?

2.1.1. De la Teoría de la X con Barra a la Estructura de Frase Desnuda (Bear Phrase Structure BPS)

Chomsky (1995): propone la Estructura de Frase Desnuda (Bear Phrase Structure (BPS) en lugar de la Teoría de la X con Barra.

- Propósito: Reducir las propiedades de la FL a necesidades conceptuales y condiciones de interface.
- BPS: conserva las propiedades empíricas centrales del esquema de X-Barra (endocentricidad, ramificación binaria, mando-c, etc) sin otra herramienta más que el Léxico y los mecanismos se siguen de la necesidad conceptual.



- (9)
- a. $\text{XP} \rightarrow \text{SPEC X}'$
 - b. $\text{X}' \rightarrow \text{X}^\circ \text{ Compl}$

- Conceptualmente, lo único que es necesario es unir/concatenar LIs para crear unidades más grandes (Frases). Esta operación es denominada *Merge*.

(10) *Merge Assumption*
Merge operates with LIs, not features

- *Merge* es potencialmente no-delimitada/irrestricida, de modo que es capaz de producir una infinitud discreta de expresiones estructuradas.
- Chomsky (2008) propone que los LIs tienen una propiedad especial que induce *Merge*, a la que se refiere como *Edge Features* (EF) [Rasgos de Filo]:

For an LI to be able to enter into computation, merging with some SO, it must have some property permitting this operation. A property of an LI is called a feature, so an LI has a feature that permits it to be merged. Call this the *edge feature* (EF) of the LI. [Chomsky 2008: 139]

1. Los LIs dotados con estos rasgos son capaces de participar en dependencias sintácticas que manifiestan cierta fuerza atractora, es decir, la habilidad de agruparse con otros LIs. Las unidades que no poseen EFs constituirían expresiones por sí mismas: interjecciones y unidades fosilizadas como las frases idiomáticas.

- (11)
- a. $\text{LI}_{\{EF\}}$ (mergeable LI)
 - b. LI (unmergeable LI)

¿Cuál es el estatuto de los EFs?

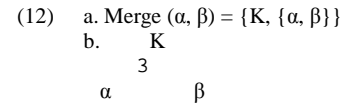
1. Los EFs no presentan diferentes valores para el rasgo, de modo que son diferentes de los rasgos de tiempo, persona o caso.
2. Los EFs se diferencian de ciertos rasgos, como los rasgo- ϕ , en que no participan en procesos de Compatibilización (Match) y Valuación.
3. Los EF no se borran (Chomsky 2007) y pueden desencadenar *Merge* con cualquier LI, sin ninguna restricción sobre los rasgos. De modo que los EF son "Match proof", es decir, no requieren de compatibilización. (Chomsky 2008)
4. Si *Merge* es definida de esta manera, la generación de estructuras sería irrestricida. No se puede impedir la generación masiva de estructuras. (Boeckx 2008, 2009).

Dos formulaciones de BPS

Chomsky (1995):

Merge es una operación que construye objetos sintácticos en la forma de conjuntos binarios.

K indica la categoría sintáctica (o el tipo semántico) del objeto sintáctico resultante: su etiqueta (*label*).



(La etiqueta en (12a) es un intento de introducir la notación de X-barra en términos de la teoría de conjuntos.)

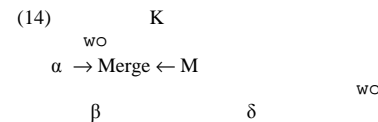
Chomsky (2004: 108; 2008: 8): la aplicación de *Merge* provee dos relaciones fundamentales:

1. *Membrecía* (*set-membership*) (contención o dominancia) que de manera derivada provee la relación *termino-de* (*term-of*)

- (13) *Term* (formerly, constituent)
- a. K is a term of K. [i.e., the entire set is a term]
 - b. If K is a term of K, then the members of the members of L are terms of K [Chomsky 1995]

2. La segunda relación fundamental que provee *Merge* es *Probe-Goal* (Chomsky 2008: 8). Típicamente, cuando dos LIs son ensamblados, uno de ellos tiene el estatuto de ser el núcleo (*Probe*) de la estructura, en términos de Collins (2002), el *locus*.

3. *Mando-c*: aparte de las relaciones de *set-membership* y *Probe-Goal*, *Merge* provee la relación *mando-c*. *Mando-c* es definido como parasitario de *Merge*: Cuando α se ensambla con β , α manda-c a todos los miembros de β .



- (15) *Derivational c-command* (Epstein's 1999)
X c-commands all and only the terms of the category Y with which X was paired by *Merge* or by *Move* in the course of the derivation [from Epstein 1999: 329]

Observación:

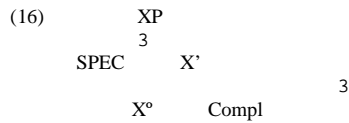
La definición estándar de *mando-c* requiere del supuesto de etiquetas/proyecciones. Esta visión es incompatible con las propuestas recientes de Chomsky de BPS en las que las etiquetas son eliminadas.

Informally, the new unit $\{\alpha, \beta\}$ is regarded as a projection of some head α or β . In phrase structure grammars, including X-bar theories, the projection is identified by a new element (N, N-bar, NP, etc.), violating the inclusiveness condition. We therefore assume that $\{\alpha, \beta\}$ is identified either by α or β (its label); a label, then, is always a head. In the worst case, the label is determined by an explicit rule for each choice of α, β . A preferable result is that label is predictable by a general rule. A still more attractive outcome is that L requires no labels at all. [from Chomsky 2004: 108-109]

- Chomsky no discute la existencia de etiquetas, sino la creación de etiquetas. El punto en cuestión es en qué medida es necesaria una operación que cree $\{K, \{\alpha, \beta\}\}$ a partir de $\{\alpha, \beta\}$, o si este paso puede ser calculado por otros medios (i.e. identificación).

- Chomsky (2008) asume que las etiquetas son identificadas mediante algoritmo que busca los objetos sintácticos mínimos (núcleos) y los toma como los *Locus* de la derivación.

- Chomsky asume tácitamente que si localizamos el núcleo de un objeto sintáctico, localizamos su etiqueta.



- (17) *Labeling Algorithm*
- In $\{H, \alpha\}$, H an LI, H is the label
 - If α is internally merged to β , forming $\{\alpha, \beta\}$ then the label of β is the label of $\{\alpha, \beta\}$
- [from Chomsky 2008: 145]

A partir de (17) es necesario reconsiderar tanto la noción de etiqueta como la de mando-c:

- Chomsky (2008) propone que las relaciones como mando-c sean reducidas a instancias de dependencias Probe-Goal, según la cual si α manda-c a β , entonces α es el Probe and β su Goal.
- El sistema de (17) considera que el estatuto de los LIs está determinado a medida que la derivación se desarrolla: al ser tomados del Lexico, los LIs son núcleos, pero luego, una vez que Merge ha sido aplicado, pueden formar frases.

Observación:

Ninguna de las dos concepciones de BPS recurre a índices, niveles de barra o alguna otra marca. Todo esto es consistente con la *Condición de Inclusividad*.

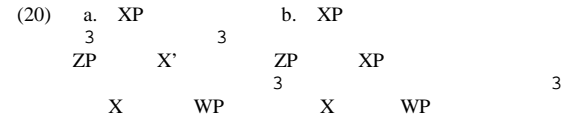
- (18) *Inclusiveness Condition*
Any structure formed by the computation is constituted of elements already present in the lexical items selected for N[umeration]; no new objects are added in the course of computation, apart from rearrangements of lexical properties (in particular, no indices, bar levels in the sense of X-bar theory, etc.) [from Chomsky 1995b: 228]

Junto con (18), Chomsky (2005: 11-13; 2007: 5; 2008: 5) propone la condición de (19)

- (19) *No Tampering Condition*
Merge of X and Y leaves the two SOs unchanged [...] Merge cannot break up X or Y, or add new features to them. Merge is invariably “to the edge” [from Chomsky 2008: 138]

¿Cuál es el estatuto de las proyecciones intermedias?

- Kayne (1994) elimina las proyecciones intermedias del sistema. Este punto no es trivial si se busca impedir que las proyecciones intermedias manden-c a sus especificadores. El modo en que se eliminan las proyecciones intermedias varía de autor en autor: Kayne estipula que X' sería una proyección máxima desde una perspectiva de la estructura de frase basada en segmentos: (20a) debería ser representada como (20b), como consecuencia se elimina la distinción especificador/adjunto.

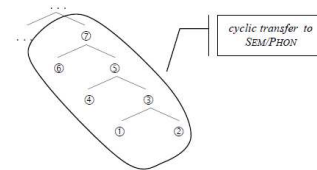


- Chomsky (1995b: 243-244), por su parte, simplemente estipula que XP y X son las únicas entidades relevantes para la estructura desnuda. X' se vuelve invisible en las interfaces.

Propuesta alternativa - Uriagereka (1999):

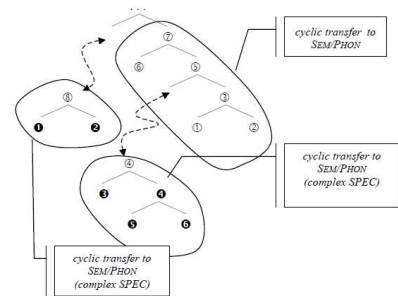
- Uriagereka (1999) en su propuesta de *Multiple Spell-Out* (MSO) aborda este problema mediante la propuesta de que hay una transferencia dinámica (*dynamic transfer*) cada vez que la derivación alcanza un punto en que *Merge* no puede seguir aplicándose. Esto da lugar a lo que Uriagereka (1999a) denomina *command unit* (CU).
- ¿Una *Command Unit* se obtiene por la aplicación continua de Merge. Si se ensamblan elementos a un marcador de frase que ya ha sido ensamblado se obtiene una *Command Unit*. (Uriagereka 1999)ç

(21) “Command Unit”



- Los especificadores complejos hacen que el sistema abandone una CU para construir una nueva, de modo que se fuerza una múltiple transferencia o múltiple Spell-Out en los que diferentes pedazos de estructura (CUs) están pegados:

(22) Not a “Command Unit”



- (22) no es una *Command Unit* porque implica la aplicación de *Merge* a diferentes objetos (véase Uriagereka 1999)

Observación:

El problema que afecta a las proyecciones intermedias descansa crucialmente en la existencia de etiquetas (Labels).

2.1.2. Estructura de Frase Desnuda sin Etiquetas

Propiedad fundamental de BPS según la formulación de Chomsky (2007, 2008): *las etiquetas no son creadas sino que son identificadas.*

¿Por qué, de acuerdo con Chomsky (1995), Merge da como resultado (23b) y no (23a)?

- (23) a. Merge (α, β) = { α, β }
 b. Merge (α, β) = { $\gamma, \{\alpha, \beta\}$ }

1. La representación de (23b) es preferible a la de (23a), pero solo en la interfaz (semántica). Cuando Merge toma V y D el output se comporta como V y no como D o como V y D juntos.
2. El etiquetamiento (Labeling) permite capturar esta asimetría. No obstante, este efecto sobre la salida puede ser interpretado idealmente como una consecuencia y no como una causa. Si (23b) fuera el resultado correcto, debería seguirse de principios que sean ciegos de los requerimientos de las interfaces.
3. La asimetría entre α y β en (23b) es requerida en ambas interfaces:
 - o Para SEM: o α o β , digamos que α es interpretado como un predicado, mientras que β es interpretado como un argumento.
 - o Para PHON: α y β difieren puesto que o bien α precede linealmente a β , o viceversa.

Las asimetrías en cuanto al etiquetamiento emergen solamente en las interfaces

- (24) *The Radical Interface Thesis*
 PHON and SEM properties of syntactic objects (i.e., linear order and semantics) are not computationally relevant, but side-effect (emergent) consequences

Considérese (24) dentro de la Teoría Temática:

Dos LI se ensamblan (First Merge):

- (25) Merge (love, Mary) = {love, Mary}

- El sistema no ve ninguna diferencia entre *love* y *Mary*. Para crear una diferencia (formalizar que *love* selecciona a *Mary*, y no al revés). Un mecanismo adicional es necesario. El etiquetamiento es un mecanismo posible que no viola la condición de visibilidad:

- (26) Merge (love, Mary) = {love, {love, Mary}}

1. En GB: *Mary* recibe un papel temático /Tema/ del verbo *love*. El proceso de asignación es problemático porque supone la introducción de formativos semánticos en la computación sintáctica. Sin embargo, no hay evidencia morfológica que apoye esta idea. No hay morfemas que puedan relacionarse directamente con "ser-Tema" o "ser-Meta".

2. La asignación de roles temáticos sería redundante si el sistema pueden utilizar la configuración (no ambigua) creada por la estructura de frase. Si la configuración puede ser utilizada por SEM de modo tal que la noción relacional de /Tema/ se obtenga.
3. Esta idea es compatible con una perspectiva neo-Davidsoniana, por ejemplo en un caso como *Mary arrive*, el ensamble da algo como (27b) y no como (27a).

- (27) a. $\exists e$ [arrive (e) & Mary (e)]
 There is an event e such that it was an arriving and it was Mary
 b. $\exists e$ [arrive (e) & Theme (Mary, e)]
 There is an event e such that it was an arriving and Mary is a Theme of e

La semántica neo-Davidsoniana que requerimos es cercana a (27b) y el etiquetamiento puede utilizarse para obtener ese resultado. Queremos que el ensamble de *arrive* y *Mary* ofrezca una relación que no existía desde un primer momento.

Una extensión de esta misma lógica se puede aplicar a casos de focalización y topicalización:

- (28) a. Charlie Mingus, John particularly liked.
 b. CHARLIE MINGUS John particularly liked!

- Las expresiones de (28) suelen ser analizadas como el resultado del chequeo de algún rasgo. El DP *Charlie Mingus* comporta un rasgo semántico que tienen que ser chequeado con el especificador de una proyección particular en el dominio del CP.
- En (28a), un caso de topicalización, el DP movido está dotado con un rasgo de [TÓPICO] que tiene que ubicarse en una relación Especificador-Núcleo particular. Esta visión carece de una motivación conceptual dentro de un sistema como el de Chomsky (2000, 20001) en los que solamente los rasgos formales están sujetos a procesos de chequeo/valuación. No hay necesidad de agregar ninguna marca específica, si lo que se mueve es identificado como un tópico una vez que alcanza su posición final.
- Chomsky sugiere que tanto las interpretaciones temáticas y las periféricas (Tópico, foco) parecen corresponderse, intralingüísticamente, con las operaciones Merge y Move.

- (4) *The C-I Hypothesis*
 C-I incorporates a dual semantics, with generalized argument structure as one component, the other one being discourse-related and scopal properties. Language seeks to satisfy the duality in the optimal way [see 2], [external Merge] serving one function and [internal Merge] the other [from Chomsky 2008: 141]

Move es simplemente otro tipo de Merge: Internal Merge

Chomsky (2004) distingue dos variedades de Merge sobre la base de una dualidad semántica en la interface SEM:

- (29) a. External Merge: Merge (α, β) $\alpha \neq \beta$
 b. Internal Merge: Merge (α, β) $\neg (\alpha \neq \beta)$
- **External Merge:** junta dos objetos sintácticos independientes (α and β) tomados del Lexicon.
 - **Internal Merge:** uno de los objetos sintácticos es tomado desde el interior de otro: α ya está presente dentro de β antes de su ensamble.

Copy Theory of Movement: Internal Merge no deja una huella, sino una copia del elemento movido. Una cadena no trivial es entonces definida como un conjunto de *ocurrencias*, es decir, como diferentes instancias de un elemento (token) léxico definido por un contexto.

Pregunta: Existe evidencia de que el etiquetamiento es necesario para las interfaces, pero ¿es necesario también durante la computación (sintáctica)?

Respuesta: El etiquetamiento no es necesario para la computación sintáctica

La formulación original de BPS recurre a las relaciones de conjunto más básicas para derivar las etiquetas:

- (30) *The label of $\{\alpha, \beta\}$ is...*
- The intersection of α and β (i.e., $\alpha \cap \beta$)
 - The union of α and β (i.e., $\alpha \cup \beta$)
 - One or the other of α, β
- Si bien en cualquiera de las tres opciones no hay una pérdida de información, la cuestión es cuál es la información relevante que debe transmitirse a la etiqueta.
- Bajo el supuesto de una descomposición de las categorías sintácticas en términos de los rasgos $[\pm N]/[\pm V]$, Chomsky (1995) rechaza tanto (30a) como (30b) y sostiene que (30c) es el resultado correcto.
- Nótese que si no se considera que las etiquetas son identificadas y no creadas, cualquiera de las opciones de (30) requiere de la postulación de una nueva operación para capturar el etiquetamiento.

Retomemos (23):

- (31) a. Merge $(\alpha, \beta) = \{\alpha, \beta\}$ label identification (see Chomsky 2004; 2005; 2007; 2008)
b. Merge $(\alpha, \beta) = \{\gamma, \{\alpha, \beta\}\}$ label creation (see Chomsky 1995b; 1995c; 2000; 2001)

Problemas aparentes: A. FIRST MERGE, B. MERGE de dos frases.

A. FIRST MERGE

Si (17) es aplicado sobre una instancia de first merge, el algoritmo fallará en identificar su etiqueta, puesto que α y β son núcleos, no puede identificarse la etiqueta mediante la cláusula a de (17).

Respuesta Chomsky al problema:

“either [LI] may project [...] [which] seems unproblematic, though one of the choices may yield some sort of deviance”.

Propuesta alternativa (Boeckx 2008c):

La derivación debe esperar hasta un segundo paso de *Merge* para que la etiqueta sea identificada.

- Propuesta alternativa paso a paso:

Supuesto: IA counts as a head, after cyclic transfer of its complement. If IA is a DP, then transfer will only leave D (or *n*, see Boeckx 2009, Chomsky 2007)

- External Merge toma un verbo y su primer argumento (el argumento interno, IA).
- (32) Merge $(V, IA) = \{V, IA\}$ no label identified
- De acuerdo con (17), la primera aplicación de Merge no dará como resultado ninguna etiqueta, por lo que debe esperar a que un tercer elemento sea introducido. EL IA cuenta como un núcleo.

- El próximo núcleo que se introduce v^* , un núcleo de fase.

(33) Merge $(v^*, \{V, IA\}) = \{v^*, \{V, IA\}\}$ v^* is the label

4. La cláusula a de (17) identifica a v^* como la etiqueta puesto que $\{V, IA\}$ no es un núcleo sino un frase. Se puede expresar la idea de Boeckx's (2008c) como en (34) y formalizar la afirmación de Chomsky, según la cual todas las operaciones excepto external Merge tienen lugar en el nivel de la fase.

(34) *Phase level labeling*
The label of K is determined at the phase level

Consecuencia de (34):

Los nodos VP o TP no existirían, ya que corresponderían al problema recién presentado (external Merge de dos LIs). Asimismo, puesto que los nodos VP and TP no pueden ser identificados por el algoritmo de (17), ni el VP ni el TP podrían moverse.

B. EXTERNAL MERGE de dos frase

- **Uriagereka (1999):** los especificadores complejos están sujetos rápidamente a Spell-Out, lo que los convierte rápidamente en unidades congeladas (frozen), de modo que no pueden proyectar su etiqueta cuando se fusionan con el v^*P .
- Chomsky (2007, 2008), en cambio, deja que el algoritmo de (17) se aplique libremente y sugiere que cualquiera de las dos etiquetas proyecta.

(35) Merge $(EA, v^*P) = \{EA, v^*P\}$

Problema: Si se considera la aplicación estricta del algoritmo, (17) no permitiría que ninguno de los dos proyecte: por (17a) no se podría la etiqueta porque el EA y el v^*P son frases y no núcleos, no son LIs, excepto que se considere que el especificador, en este caso el EA, está congelado como en el sistema de Uriagereka. Por otra parte, (17b) no se aplica porque no sería un caso de Internal Merge, el EA no es un miembro del v^*P .

- Gallego explora una propuesta alternativa según la cual el EA se ensambla debajo de v^* , como un adjunto del VP.

(36) $\{v^*, \{EA, \{V, IA\}\}\}$

De acuerdo con (17), v^* es el elemento que el sistema identifica núcleo/etiqueta/Probe.

Ahora bien, en (36), el que sea el EA el elemento que se mueve está determinado por el hecho de que es un elemento que está activo. Un elemento está activo si su rasgo de Caso no está valuado.

- De acuerdo con Boeckx (2003) y López (2007), Gallego asume que solo los elementos activos están sujetos a movimiento. En (37), el IA estaría inactivo y el EA estaría forzado a moverse para ser sondeado por un núcleo más alto que valúe Caso.

(37) a. $\{v^*, \{EA, \{V, IA_{[-i]}\}\}\}$
b. $\{EA_{[+i]}, \{v^*, \{t_{EA}, \{V, IA_{[-i]}\}\}\}\}$

El EA en (37b) es inmune a la transferencia cíclica, puesto que en el nivel de la fase se encuentra en el filo (Edge). Asimismo, todas las frases que están sujetas a External Merge con $\{V, IA\}$, $\{EA, \{V, IA\}\}$, o $\{v^*, \{EA, \{V, IA\}\}\}$ tienen que ascender.

- (46) a. External Merge: argument structure (formerly, deep interpretation)
 b. Internal Merge: discourse-oriented properties (formerly, surface interpretation)
 c. Pair Merge: predicate composition

La adjunción en BPS:

La adjunción es formalizada mediante pares ordenados, ya que es el único modo de imponer una asimetría dentro de un conjunto.

- (47) a. $\{\alpha, \beta\} = \{\beta, \alpha\}$ set (no asymmetry between α and β)
 b. $\langle \alpha, \beta \rangle \neq \langle \beta, \alpha \rangle$ pair (α goes before or after β)

Chomsky's (2004): el análisis de *Pair Merge* supone que los adjuntos involucran una estructura más compleja que la de *set merge*: $\langle \alpha, \beta \rangle$ no es $\{\alpha, \beta\}$, sino $\{\{\alpha\}, \{\alpha, \beta\}\}$.

Discusión: La adjunción presenta dependencias más simples que la selección de argumentos (solo los argumentos tienen que recurrir a la noción de papel temático para cambiar el estatuto del DP, etc). Los adjuntos son simples predicados de eventos:

- (48) a. John read *War and Peace* on Saturday.
 b. $\exists e$ [read (e) & Agent (John, e) & Theme (*War and Peace*, e) & on-Saturday (e)]
- Si se considera que la semántica se sirve de la sintaxis, la postulación de Pair Merge presenta un segundo problema, ya que es una operación que no es lógicamente necesaria.
 - Lo único que se necesita para capturar la diferencia semántica de la composición de predicados es que su sintaxis sea diferente de aquella de la selección de argumentos.
 - La adjunción se obtiene mediante set Merge y la asimetría con la selección de argumentos es una consecuencia de la falta de etiqueta.
 - La estructura resultante de la adjunción de un XP a un VP sería [VP, XP]. Toda la estructura se comporta como un VP porque la categoría que recibe la adjunción retiene todas sus propiedades.
 - El agregado de un adjunto no modifica la categoría sintáctica del objeto que es modificado. En (49) el *have* perfectivo subcategoriza un V perfectivo marcado con *-en*, el agregado del adjunto *quickly* o *in the yard* no cambia sus requerimientos seleccionales.

- (49) a. [{Has/*is} [VP eaten a bagel]]
 b. [{Has/*is} [VP [VP eaten a bagel] quickly] in the yard]]

Reformulación del análisis de Chomsky (2004): No hay una variante adicional de Merge

- (50) *Merger of adjuncts*
 In the case of adjuncts (vis-à-vis argument-taking), no label can be identified
- De acuerdo con (50) los adjuntos y los EAs son dependientes similares.
 - La única diferencia entre adjuntos y EAs es que los EAs están “activos”, de modo que son capaces de moverse para recibir caso.
 - Los EA deben ser los primeros especificadores para que la asimetría con respecto al IA sea creada. → Con excepción del primer especificador, ninguno de los otros puede ser distinguido formalmente.
 - **Consecuencia:** no puede recibir una interpretación especial en el componente SEM.

- (51) {XP, {YP, {EA, {v*, {V, IA}}}}}

- EA y IA son diferenciados por el hecho de que v^* -V los separa. Como consecuencia el algoritmo de (17) puede crear una diferencia formal entre ambos.

Síntesis parcial:

1. El sistema de BPS sin etiquetas de Chomsky (2007, 2008) no puede distinguir entre especificadores complejos y adjuntos.
2. Los adjuntos son básicamente especificadores que están inactivos, como consecuencia no pueden moverse para volverse visibles para la asignación de caso.
3. Este sistema permite capturar las siguientes propiedades de los adjuntos:
 - (i) su estatuto de islas que se sigue de que son inactivos.
 - (ii) su condición de no tener etiqueta, que es un resultado natural dentro del algoritmo de (17).

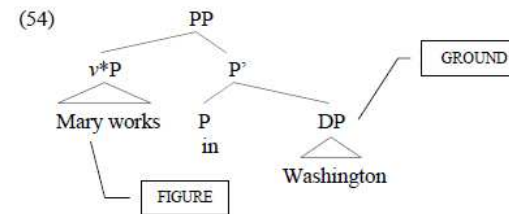
Sobre la interpretación de los adjuntos por los sistemas C-I

Gallego (en prensa) propone un análisis de los adjuntos según el cual son tratados como PP que establecen una relación de Figura-Fondo (Talmy 2000) con el VP.

- Los modificadores del VP en (52) exhibirían la misma sintaxis que los modificadores en (53)
- (52) Mary works in Washington.

- (53) Mary is in Washington.

- En (53), *in* establece una relación Figura-Fondo entre *Mary* y *Washington*. El mismo tipo de dependencia se establecería entre *Mary works* y *Washington* en (53). *Mary works* es la Figura y *en Washington* es el fondo.
- Interpretación “The event of Mary working is (placed) in Washington”.



- El v^*P *Mary works* sería introducido por p , una versión contextual de v^* en el dominio de P.
- p sería un núcleo *aplicativo alto* en el sentido de Pylkkanen (2008).
- La sintaxis de (54) sería básicamente la de una cláusula mínima.
- La representación de (54), no obstante, no es inmediatamente compatible con la teoría BPS sin etiquetas. ¿Cómo se obtienen estas relaciones de Figura-Fondo en un análisis sin etiquetas?
 - El DP dentro del PP es interpretado como el fondo por la búsqueda mínima realizada por p , un núcleo análogo a v^* .
 - El v^*P es interpretado como la Figura debido a la distinción Especificador-Complemento que establece p .

- (55) a. Merge (in, Washington) = {in, Washington} no dependency established
 b. Merge (p, {in, Washington}) = {p, {in, Washington}} {in, Was.} is the complement of p
 c. Merge ({Mary, {v*, works}}, {p, {in, Washington}}) = {{Mary, {v*, works}}, {p, {in, Washington}}} {Mary, {v*, works}} is the specifier of p

Síntesis: No es necesario postular una operación independiente de adjunción. Las propiedades fundamentales de los adjuntos pueden ser capturadas si dan lugar a estructuras sin etiquetas (como los especificadores complejos), si están “inactivos” y si su semántica se ajusta a la sintaxis de Figura-Fondo de los PP.

2.2. Agree

En P&P se proponen varios mecanismos para capturar la idea de que los rasgos abstractos (especialmente el Case) debe ser chequeado. En GB y las primeras formulaciones minimalistas, el chequeo de rasgo complementa la teoría temática. Una cadena argumental tiene una posición temática y una posición de caso.

(56) Chain Condition

An A-chain must be headed by a Case position and must terminate in a θ-position
 [from Chomsky & Lasnik 1995: 46]

La idea general consiste en que las operaciones de chequeo (temáticas y no temáticas) son creadas por internal merge, y se realizan en una configuración Especificador-Núcleo.

- (57) *Jean a vu(*-e) la fille.* (French)
 Jean have-3.SG seen-FEM.SG the-FEM.SG girl-FEM.SG
 ‘Jean has seen the girl’

[from Boeckx 2004: 23]

- (58) a. *Jean l’a vu-e.* (French)
 Jean CL-her-have-3.SG seen-FEM.SG
 ‘Jean saw her’

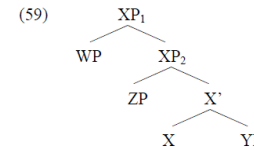
- b. *Quelle fille Jean a(-t-il) vu-e.* (French)
 which-FEM.SG girl-FEM.SG Jean have-3.SG-he seen-FEM.SG
 ‘Which girl did Jean see?’

- c. *Cette fille a été vu-e.* (French)
 this-FEM.SG girl-FEM.SG have-3.SG been seen-FEM.SG
 ‘This girl was seen’

[from Boeckx 2004: 23]

(58) muestra que el participio pasado muestra concordancia con el objeto siempre que el de DP se ha movido – de manera cíclica sucesiva- de un especificador a otro especificador. in a successively cyclic fashion– from specifier to specifier. Se ha observado, no obstante, que algunas variantes de chequeo no parecen requerir de desplazamientos, que se explicaba mediante movimientos encubiertos.

En el minimalismo temprano las operaciones de chequeo requerían de dominios particulares denominados “checking domains”.



Complement domain = YP
 Checking domain = WP, ZP, and X

[adapted from Chomsky 1993a: 177]

Para refutar la idea de que el chequeo de rasgos se realice solamente en una relación especificador núcleo, es necesario encontrar casos en los que la concordancia se visible y no haya una configuración Especificador-Núcleo disponible.

- (60) There {*seems/seem} to be two men in the boat.
 [from Boeckx 2004: 24]

- (61) *Mér {*virðist/virðast} Þeir vera skemmtilegir.* (Icelandic)
 me-DAT seem-{3.SG/3.PL} they-NOM be-INF interesting
 ‘It seems to me that they are interesting’

[from Boeckx 2004: 24]

- (62) *Vivek-ne [kitaab parh-nii] chaah-ii.* (Hindi)
 Vivek-ERG book-FEM read-INF-FEM want-PFV-FEM
 ‘Vivek wants to read the book’

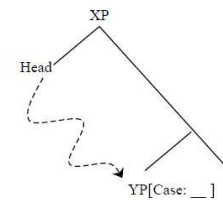
[from Boeckx 2004: 25]

Los casos de (60)-(62) no pueden ser explicados mediante el chequeo en una relación Especificador-núcleo.

Boeckx (2004) propone que estos casos pueden explicarse por un mecanismo de chequeo a larga distancia bajo mando-c como el de (63).

çç

- (63) Case Assignment (at a distance)



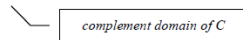
D’Alessandro & Roberts (2008) proponen un abordaje en términos de fases que disocia el movimiento de la concordancia. La idea central es que la concordancia no solo se puede establecer entre elementos que ocupan la misma fase, sino también entre elementos que ocupan el mismo dominio de complemento de una fase.

- (64) *Complement Condition on Agreement (CCA)*
 Given an Agree relation A between probe P and goal G, morphological agreement between P and G is realized iff P and G are contained in the complement of the minimal phase head H.
 [from D'Alessandro & Roberts 2008: 482]

Esto permite explicar la concordancia con el participio pasado de los ejemplos anteriores sin recurrir a un requerimiento de una relación especificador-núcleo. Esto se debe a que el núcleo relevante ocupa el mismo dominio de complemento del núcleo de fase, después de que el movimiento del participio haya tenido lugar.

- (65) *Le ragazze sono arrivate.* (Italian)
 the girls-FEM.PL be-3.PL arrived-FEM.PL
 'The girls have arrived'
 [from D'Alessandro & Roberts 2008: 483]

(66) $[_{CP} C [_{TP} Le ragazze_z T [_{VP} v_{Aux} [_{VP} arrivate_i v_{Prt} [_{VP} t_i t_z]]]]]]$



- (67) *Juan ha {llamado /*llamada} a María.*

(68) $[_{CP} C [_{TP} Juan T [_{VP} ha v_{Aux} [_{VP} llamado_z v_{Prt} [_{VP} t_z a María]]]]]]$



Problema:

- (69) *Conosciuta, Marco non ebbe più paura.* (Italian)
 known-FEM.SG-CL-her.FEM.SG Marco not had more fear
 'Having met her, Marco was not afraid anymore'
 [from D'Alessandro & Roberts 2008: 485]

Solución: Phase Sliding (Gallego 2005, 2009)

Merge provee dos relaciones sintácticas: set-membership and Probe-Goal.

Chomsky (2000; 2001) introduce la operación Agree que se aplica en una relación Probe-Goal. Básicamente, asume que las categorías funcionales centrales (CFC: C, T, y v) son introducidas en la sintaxis con un conjunto de rasgos-φ no-interpretables ([gender], [number], and [person]), que deben ser eliminados.

Pesetsky & Torrego (2007): la (no)interpretabilidad no depende de los rasgos como tales, sino de los LIs que los portan. Si un rasgo F hace una contribución semántica a un LI dado, entonces F es interpretable en ese LI

- (70) *Feature Interpretability*
 A feature F is interpretable at SEM if it makes a semantic contribution in the LI in which it appears, otherwise F is uninterpretable and must be valued and deleted

- (70) puede ser interpretado como un modo de codificar la observación de que los rasgo-φ son interpretables en los nombres, pero no en los verbos.

Si la sintaxis es ciega a los requerimientos semánticas, ¿en qué medida la interpretabilidad es relevante durante la computación?

Chomsky (2001: 5; 2004: 116) formula la idea de que los rasgos no interpretables entran en la derivación no valuados. De lo único que se interesa la sintaxis, entonces, es de la valuación, y no de su interpretabilidad.

Los LIs, de este modo, vienen desde el lexicon no como en (71a), sino como en (71b)

- (71) a. LI [iF] (interpretable) vs. LI [uF] (uninterpretable)
 b. LI [3F] (valued) vs. LI [_ F] (unvalued)

Más precisamente, los rasgos adoptan la forma de (72), que presenta dos componentes: un atributo y un valor:

- (72) [attribute: value]

(73) *Attribute – Values*

ATTRIBUTE	VALUE	ATTRIBUTE	VALUE
[person]	{first, second, third}	[wh]	{± at most}
[gender]	{masculine, feminine, neuter}	[topic]	{± at most}
[number]	{singular, plural, dual}	[focus]	{± at most}
[case]	{structural, inherent, oblique}	[neg]	{± at most}
[tense]	{past, present, future}	[specific]	{± at most}
[mood]	{indicative, subjunctive, (conditional)}	[affective]	{± at most}

Chomsky (2000; 2001) sostiene que T y v* son tomados del lexicon con sus rasgos-φ no valuados, lo que los hace actuar como Probes que buscan un Goal (i.e., el elemento más cercano que esté mandado-c y que compatibilice todos sus rasgos)

- (74) *Conditions on Agree*
 a. Probe and Goal must be active for Agree to apply
 b. Agree divides into Match and Valuation
 c. Probe must contain a full set of features (it must be complete) to delete the uninterpretable FF of matched Goal
 [adapted from Chomsky 2001: 6]

Actividad: los rasgos flexivos desencadenan operaciones de chequeo, de modo que la morfología no interpretable (uFF) hace que los objetos sintácticos estén activos.

- (75) *Activity Condition*
 Uninterpretable (unvalued) morphology renders syntactic objects 'active'

Los rasgo-φ activan a las CFCs mientras que el Caso activa a los elementos nominales.

Los uFF deben ser eliminados antes de que la derivación sea mapeada a SEM, puesto que el rasgo de número, por ejemplo, no puede estar presente en T en la semántica. Cuando un rasgo recibe un valor se vuelve inactivo.

For reasons that will become clear when the notion of *phase* is introduced, activity operates within computational boundaries; features eventually get a value, and become 'inactive'. What effects does inactivity have? In the case of nominals, *Hyperraising*

(76) * $[_{CP} C [_{TP} John_i T \text{ seems } [_{CP} C [_{TP} t_i T \text{ likes } Mary_j]]]]$

(77) *Freezing Effect* (non-final version)
 DPs whose Case have been checked are computationally inert

Match

Chomsky (2000; 2001) sostiene que Agree es parasitario de la relación Match, una dependencia que es sensible al tipo de rasgos que dos Lis comparten, no su valor.

(78) *Match*
 F and F match if they belong to the same attribute class (e.g., [number], [Case], etc.), independently of value (e.g., singular vs. plural, nominative vs. accusative, etc.)

(79) LI [person:] . . . LI [person: 2nd]

(80) $[_{CP} C [_{TP} T_{[\phi]} [_{v^*P} John_{[3.SG]} v^*_{[\phi]} [_{VP} \text{ loves his daughter}_{[3.PL]}]]]]]$

En (80) hay una estructura transitiva nucleada por el verbo liviano v^* . En (81), el conjunto- ϕ de v^* sondea *his daughters*, y T sondea a *John*.

(81) $[_{CP} C [_{TP} T_{[\phi]} \underbrace{[_{v^*P} John_{[3.SG]}]}_{\uparrow} v^*_{[\phi]} \underbrace{[_{VP} \text{ loves his daughters}_{[3.PL]}]}_{\uparrow}]]]]$

(82) a. Probe $[\phi] \leftrightarrow [3.PL] \text{ Goal}$ step 1: Match
 b. Probe $[3.PL] \leftarrow [3.PL] \text{ Goal}$ step 2: Valuation

La valuación requiere de Match, pero Match no requiere de valuación.

Hay dos casos en los que Match no es seguido por la valuación del rasgo. El primero se da cuando uno de los dos elementos que entran en una relación de Match es defectivo.

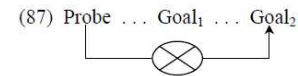
(83) *Defectiveness*
 An LI is defective if it lacks some attribute(s) of a given class

(84) *Defective CFC*
 a. T_{def} : raising and ECM structures
 b. v_{def} : unaccusative and passive structures

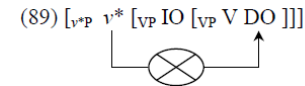
(85) $[_{CP} C [_{TP} T_{[\phi]} [_{v^*P} v_{[\phi]} \text{ arrived } John_{[3.SG]}]]]]$

El Segundo ocurre cuando el un LI contiene tanto rasgos interpretables como no interpretables.

(86) *Freezing Effect* (non-final version)
 DPs whose Case has been checked are rendered computationally inert, but their iFF remain visible for Match, triggering *defective intervention effects*



(88) a. **Pedro le me envía.* (Spanish)
 b. *Pedro me lo envía.* (Spanish)



(90) *Juan_i (*me) parece* $[_{TP} t_i T_{def} \text{ tener problemas}]$ (Spanish)

(91) *Merge Assumption*
 Merge operates with LIs, not features

(92) *Agree Assumption*
 Agree operates with features, not LIs

3. Locality and the concept of cycle

El sistema obedece restricciones de localidad que determinan que las operaciones ocurran en dominios específicos: las fases.

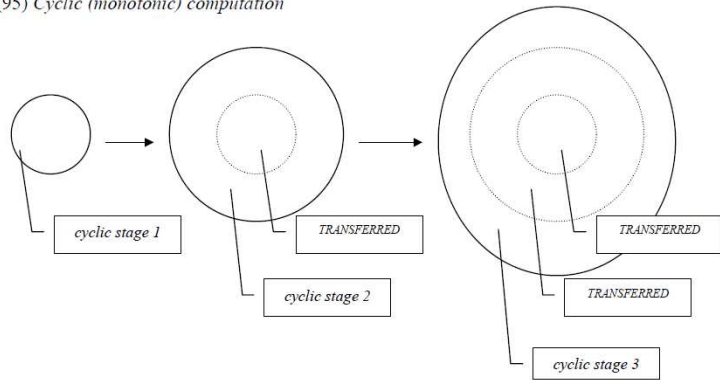
La dinámica computacional está rigida por una periodicidad sistémica, que típicamente se corresponde con fenómenos que incluye tanto cuestiones vinculadas con el fraseo fonológico como propiedades interpretativas intrincadas como las de ligamiento y se extiende hasta el establecimiento de dependencias de Caso/concordancia.

La idea de que las derivaciones (sintácticas) obedecen condiciones de ciclicidad impide que una regla R de cierto tipo puede aplicarse en una derivación dada si una regla diferente R' no ha podido aplicarse antes. En este sentido, las reglas derivacionales se aplican partiendo desde el constituyente más pequeño y luego se aplica al siguiente constituyente más grande, que lo incluye.

En este sentido, al considerar que los algoritmos que construyen la estructura deben preservar el input de una operación dada y que ese input debe poder ser identificado en el resultado de la operación, ciertos dominios sintácticos tienen un estatuto especial por cuanto requieren que las operaciones se apliquen dentro de sus límites, sin que sea necesario volver a estadios previos de la derivación o que deba anticiparse a estadios subsecuentes.

(94) *Strict Cycle Condition*
 No rule can apply to a domain dominated by a cyclic node A in such a way as to affect solely a proper subdomain of A dominated by a node B which is also a cyclic node
 [from Chomsky 1973: 97]

(95) Cyclic (monotonic) computation



Existen ciertos fenómenos empíricos que sugieren que la computación procede más o menos como está ilustrada en (95).

Dependencias anafóricas

Las dependencias anafóricas deben ser licenciadas dentro de ciertos dominios, típicamente la cláusula:

- (96) a. John_i shaved himself_i
 b. *John_i said [CP that Peter shaved himself_i]

(96b) → *himself* no puede estar ligado por *John*, puesto que parecen no estar lo suficientemente cerca para que se pueda establecer la dependencia en cuestión. Si *John* y *himself* estuvieran más cerca como en (97), la dependencia de (96) podría aplicarse nuevamente.

- (97) Peter said [CP that John_i shaved himself_i]

Esta situación parece explicarse por restricciones estructurales: el sistema no puede ir más allá de la cláusula para establecer la dependencia entre *himself* y su antecedente. El sistema parece relacionar *himself* con el primer DP disponible en ese dominio. Esta misma idea explica la agramaticalidad de (98)

- (98) *You saw [DP my_i pictures of yourself_i]

La anaphora *yourself* no puede esperar a que en la derivación se introduzca el pronombre *you*, en consecuencia, dado que *my* el antecedente potencial más cercano, *my* y *yourself* quedan coindizados.

En (99), *herself* puede tomar como antecedente tanto a *Susan* como a *Mary*. Este hecho puede ser misterioso si la anáfora solo puede estar coindizada por un DP en la cláusula en la que se origina. La coindización con Susan es posible porque antes del movimiento-wh, la anáfora está en una posición mandada-c por Susan.

- (99) [CP [Which pictures of herself_j] C did Mary_i say [CP that Susan_i likes t_j]]?

- (100) [CP Mary_i said [CP that Susan_i likes [DP which pictures of herself_i]]]?

No obstante, si el ligamiento requiere que haya una relación de mando-c entre el elemento que liga y el elemento ligado, la coindización entre *Mary* y *herself* en (99) debe quedar descartada. Para explicar esto, se considera que la frase-wh pasa por una posición lo suficientemente cercana a *Mary* en la que la condición de mando-c pueda ser satisfecha (101). Y lo mismo puede obtenerse en cierto tipos de dependencias a larga distancia (102)

- (101) Mary said [CP [which pictures of herself_i] C Susan likes t_i]

- (102) [CP Which pictures of herself_i did Mary say [CP that John thinks [CP that Susan likes t_i]]]?

¿Cómo se crea la dependencia entre la posición de superficie y las posición de base de la frase-wh? La frase-wh se mueve desde su posición de base a al Spec-C matriz aterrizando en cada especificador que encuentre en el camino o solo se mueve a especificadores particulares que se encuentran entre la posición de base y la posición de superficie?

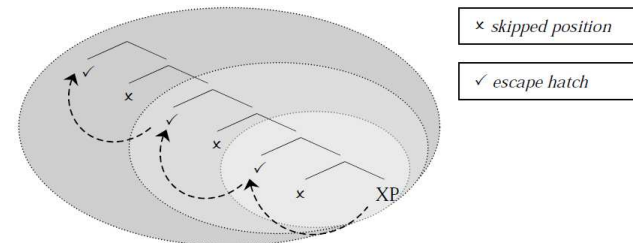
(103) Filler-Gap Relations

- a. Uniform: movement proceeds through every node along the path⁴⁷
 b. Punctuated: movement targets dedicated positions (escape hatches)

Hay evidencia para defender que el movimiento debe tener pasos intermedios y de modo que no puede proceder de un solo salto desde su posición de base del frase movida a su posición de superficie. No obstante, hay una cuestión que debe determinarse que tan grande es este movimiento paso-a-paso.

La visión más tradicional es que el movimiento procede a través de posiciones especializadas, la hipótesis (103b), que se puede ver en (105)

(105) Successive cyclicity

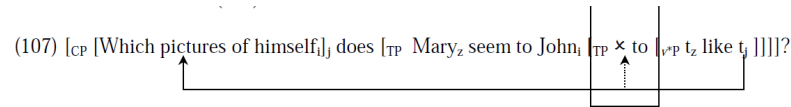


La evidencia a favor de esta hipótesis surge de datos como los de (106)

- (106) a. *_{CP} John_i said [_{CP} that Sue likes pictures of himself_i]]
 - b. [_{CP} [Which pictures of himself_i]] did John_i say [_{CP} that Sue likes t_j]]?
 - c. *_{CP} Mary_j seems to John_i [_{TP} to t_j like pictures of himself_i]]
 - d. *_{CP} [Which pictures of himself_i] does Mary seem [_{TP} to John_i [_{TP} to like t_j]]]?
- [from Abels 2003: 30]

- (106a) es agramatical por *John* no puede ligar a *himself*.
- (106b) es gramatical porque la frase-wh puede ocupar la posición de Spec-C intermedia.
- (106c) es agramatical porque Mary deja una copia entre *John* y la anáfora.

El caso crucial es (106d), según Abels (2003) la agramaticalidad de (106d) se explica porque el movimiento de la frase-wh no deja ninguna copia en el especificador del TP no finito.



Siguiendo esta lógica según la cual si hay una copia, debe (o puede) haber reconstrucción, pero si no la hay la reconstrucción debe fallar. La predicción que surge es que (108) debería ser gramatical, si la frase-wh debe ocupar el Spec-CP más incrustado.

- (108) *_{CP} [Which pictures of himself_i]] does [_{TP} Mary seem to Susan [_{TP} to have told . . .
- . . . John_i [_{CP} that she likes t_j]]]]?

Asimismo, esta lógica es incompatible con el movimiento-A local, que en general no admite reconstrucción.

- (109) No one is certain to solve the problem.
- [from Boeckx 2007: 59]

(109) no puede ser parafraseado por “It is certain that no one will solve the problem”. Esto sugiere que el sujeto no puede reconstruirse por debajo de la negación. Por último, (110) muestra que la frase-wh movida no puede reconstruirse en una posición dentro de la cláusula subordinada, ya que el pronombre está ligado por el sujeto *everybody*:

- (110) [_{CP} [Which of his_(z/1) pictures]] did Bill_z ask me [why nobody/everybody₁ hated t_j]]?

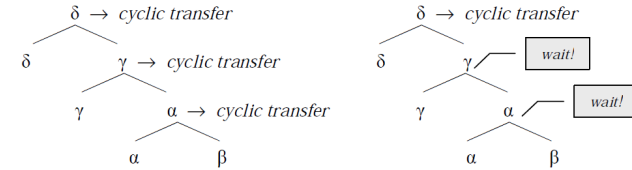
Estos datos muestran que la hipótesis (103b) no es correcta. Es decir *la ausencia de efectos de reconstrucción no nos conduce directamente a la conclusión de que no hay copias*. De todas maneras, esto no explica por qué la reconstrucción debe realizarse en ciertas posiciones especiales, no obstante permite descartar una opción teórica.

La noción de ciclo:

La idea de que la computación se ajusta a consideraciones de economía del tipo de las que fueron presentadas previamente son las que se indagan desde Chomsky (1991). Esta es la lógica que subyace a las estrategias de último recurso, a la comparación de derivaciones que tienen un acceso secuencial al Lexicon, y a la eliminación de la teoría de símbolos *ad hoc*, etc.

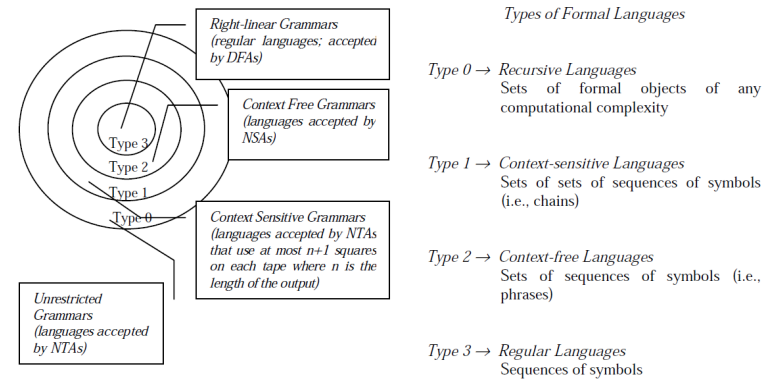
En la actualidad, la discusión teórica gira en torno de cuánta estructura hay involucrada en cada ciclo. Por un lado, propuesta como las de Epstein et al (1998) sugieren que cada aplicación de una regla transformacional constituye un ciclo. Por el otro, Chomsky (2000:2001: 2008) permite que los ciclos “crezcan” un poco antes de ser enviados a las interfaces:

- (111) a. *Cyclic Syntax* (see Epstein et al. 1998) b. *Cyclic Syntax* (see Chomsky 2000 and sub.)



La concepción de Chomsky supone una mayor carga de memoria. Es posible reinterpretar esta observación diciendo que los ciclos de Chomsky involucran memoria de corto plazo. Esto es consistente con el hecho de que las lenguas naturales son sensibles-al-contexto, en el sentido de la Jerarquía de Chomsky.

- (112) *The Chomsky Hierarchy* (see Chomsky 1956)



- (113) a. Recursively enumerable languages (objects of any complexity) – *unrestricted grammars*
 - b. Context-sensitive languages (sets of symbol string sets) – *context-sensitive grammars*
 - c. Context-free languages (sets of symbol string) – *context-free grammars*
 - d. Regular languages (strings of symbols) – *regular grammars*
- [from Uriagereka 2008: 226]

La importancia de la carga de memoria en este punto no tiene que ver con la performance, sino con los recursos del sistema que requiere el procedimiento generativo por sí mismo.

- Tipo 0 – Máquinas de Turing
- Tipo 1 - Internal Merge

Grupo Minimalista

Tipo 2 – External Merge

Tipo 3 – Máquina marcovianas.

Existen, entonces, buenas razones para tomar un abordaje de los procesos lingüísticos que considera que la sintaxis opera con conjuntos léxicos pequeños que son llevados a los componentes interpretativos de una manera cíclica. Esto no solo se apoya sobre bases empíricas sino que se ajusta al tipo de abordaje orientado a principios de economía que persigue el minimalismo: dentro de una computación cíclica, el sistema puede concentrarse en espacios de trabajos particulares, permitiéndole olvidar los estadios previos, lo que reduce la carga computacional.