

Constraints in Syntax

[1] Phrase Structure and Derivations

SoSe 2005, Universität Leipzig

Gereon Müller

gereon.mueller@uni-leipzig.de

1. Background

Task:

Determine whether a given string of words is a grammatical (well-formed) sentence of language L_i or not.

Approach:

Throughout, a derivational, incremental approach to grammar will be adopted (following Chomsky (1995; 2000; 2001a)).

(1) *The syntactic component of a grammar:*

- a. Lexicon: Set of lexical items (LIs)
- b. Operations: Merge, Move
- c. Constraints: derivational, representational, global, transderivational, translocal

Categories:

LIs can be grouped together if they have a similar syntactic distribution and similar morphological properties: word classes, categories. There are *lexical* (typically open-class) and *functional* (typically closed-class) categories (both are lexical items).

(2) *Lexical categories:*

- a. N (noun): *Mary, man, book, idea, ...*
- b. V (verb): *snore, meet, kiss, give, believe, ...*
- c. A (adjective): *green, nice, fond, ...*
- d. P (preposition): *of, under, for, after, ...*

(3) *Functional categories:*

- a. D (determiner): *the, a, every, some, three, she, it, Ø ...*
- b. T (tense): *did, can, will, Ø, ...*
- c. C (complementizer): *that, whether, if, for, Ø*

(4) *The make-up of LIs:*

- a. Phonological representation
- b. Semantic denotation: This includes the *argument structure* or Θ -*grid* of a predicate (see Heim & Kratzer (1998))
- c. Morpho-syntactic features (incl. categorial features, Case features, ϕ -features)

(number, person, gender), tense features, and selectional features)

Basic assumption:

Syntactic structures arise by iterative application of simple operations to LIs and categories formed by these operations – a building block system that makes syntactic structures grow. This procedure is called a *derivation*. Every intermediate structure thus formed is a *representation*. A sentence is a final representation that results when the derivation terminates. Chomsky (1995; 2000; 2001a) assumes that before a derivation starts, all the LIs that will be used in the derivation are assembled in a *lexical array* (LA) (sometimes, the notion of *numeration* is used – a numeration is a lexical array in which one and the same LI can be selected more than once).

(5) *Lexical array* (LA):

The lexical array is a set of LIs that are selected before the syntactic derivation starts.

(6) *Inclusiveness Condition:*

Material that is not part of the lexical array (other LIs, additional features) is inaccessible throughout a derivation.

2. Merge

The first basic operation that drives derivations is the operation *Merge*, which applies to two categories and yields a complex, structured object (a *phrase marker* or *tree* that is necessarily binary branching). If Merge uses a LI from the LA, the LI has irrevocably left that array.

(7) *Merge:*

Merge(α, β) yields $[\gamma \ \alpha \ \beta]$.

(8) *Head:*

- a. If γ has been created by Merge of α and β , α is the *head* of γ (and γ is a *projection* of α) if α selects β .
- b. A head and its projection share morpho-syntactic features.

Note:

A LI may require one or more other categories with certain properties in its projection in the syntax. Requirements of this type can be encoded in *selectional features* on a LI: [$*F^*$], where [F] is the feature that the LI is looking for. [F] is typically a categorial feature (in which case the selection requirement is often called a *subcategorization* requirement); but it may also be some other morpho-syntactic feature. Selectional features have a peculiar status: They do not seem to be interpretable semantically (at the level of Logical Form).

(9) *Interpretability Condition:*

Features on LIs that are uninterpretable at level R_i must be removed at level R_{i-1} .

(10) *Levels of grammar:*

Lexicon (plus Morphology) → Syntax → Phonological Form, Logical Form

Assumption:

Selectional features must be deleted in the course of a syntactic derivation. Merge deletes a selectional feature under identity with the corresponding feature on a selected item; if uninterpretable, the corresponding feature is also deleted.

(11) *Economy Constraint on Merge:*

Merge can only apply if it deletes a selectional feature of a LI.

(12) *The Scope of Feature Deletion:*

Deletion of selectional features implies deletion of matching uninterpretable features.

(13) *Crucial concepts of phrase structure:*

- a. β is a *complement* of α iff β has been merged with a LI α that selects it.
- b. β is a *specifier* of α iff β has been merged with a non-LI α that selects it.
- c. α and β are *sisters* iff they have been merged.
- d. γ *immediately dominates* α iff Merge has applied to α yield γ . (In that case, α is a *daughter* of γ .)
- e. γ *dominates* α iff (a) or (b) holds:
 - (i) γ immediately dominates α .
 - (ii) γ immediately dominates δ , and δ dominates α .
(α is a *term* or *constituent* of γ iff γ dominates α .)
- f. α *c-commands* β iff (a) or (b) holds:
 - (i) β is a sister of α .
 - (ii) β is dominated by a sister of α .
- g. γ is an XP category iff it is the maximal projection of α .
- h. γ is an X' category iff it is a non-maximal, non-minimal projection of α .
(Note: The status of a projection as an XP, X', or X (= LI) may change during the derivation.)

Note:

As such, the operation Merge does not say anything about linearization. Linear precedence is handled separately, by linear precedence statements (see Gazdar, Klein, Pullum & Sag (1985)).

(14) *Linear precedence statements* (English):

- a. A head precedes its complement: X CompX.
- b. A head follows its specifier(s): SpecX X.

(15) *Linearization Constraint on Merge:*

The output of Merge in language L_i must conform to the linear precedence statements of L_i .

3. Verb Phrases

(16) *Verb types:*

- a. intransitive verbs: *sleep, snore, arrive*
argument structure: $[\Theta_1]$
- b. transitive verbs: *kiss, like, believe*
argument structure: $[\Theta_1 > \Theta_2]$
- c. ditransitive verbs: *give, send, show*
argument structure: $[\Theta_1 > \Theta_2 > \Theta_3]$

Note:

Θ -roles are sometimes given names, such as “Agent”, “Patient”, “Goal”, “Experiencer”.

Terminology:

Linguistic expressions that realize Θ -roles of the argument structures of verbs are called *arguments*. The argument that realizes the highest Θ -role of a (non-trivial) argument structure is sometimes called the *external* argument; other arguments can be referred to as *internal* arguments. Linguistic expressions that have argument structures (i.e., that need to combine with arguments in the syntax) are called *predicates*.

Intended state of affairs:

The highest Θ -role ends up on the highest argument in VP, the lowest Θ -role on the lowest VP-internal argument.

Question:

How can this be ensured by the Merge operation?

(17) *Linking Principle:*

The semantically-based argument structure corresponds to a reverse hierarchy of syntactically accessible selectional features.

Assumption:

The *Economy Constraint on Merge* is revised appropriately:

(18) *Economy Constraint on Merge* (revised):

Merge can only apply if it deletes the highest-ranked selectional feature of a LI.

(19) *Selection requirements and argument structures of verbs:*

- a. arrive: $[*D^*]$ $\leftarrow [\Theta_1]$
- b. like: $[*D^*] > [*D^*]$ $\leftarrow [\Theta_1 > \Theta_2]$
- c. believe: $[*C^*] > [*D^*]$ $\leftarrow [\Theta_1 > \Theta_2]$

Note:

believe (and even *kiss*) can also impose other selection requirements based on the same

argument structure.

Assumption:

Words like the verbs in (19) enter LAs after having passed the morphological component, as fully inflected word forms.

(20) is a derivation of a sentence with a transitive verb. Categorial features are represented as indices on constituents in the derivation (*labelled bracketing*). The LIs of the LA are simplified.

(Gender agreement is not morphologically visible in English, which has an impoverished inflectional system. Gender marking on a verb may become visible in languages like Russian – *On čital/ona čitala knigu*, ‘He/she read a book’.)

(20) *A LA of a VP with an intransitive verb:*

- a. arrived: { [V], [3pers,-pl,+masc], [+fin,+past], [*D*] }
- b. he: { [D], [3pers,-pl,+masc], [nom] }

(21) *A derivation of a VP with an intransitive verb:*

Merge ([D he] , [v arrived]) → [VP [v arrived] [DP he]]

Note 1:

Merge is possible because it deletes the selectional, uninterpretable [*D*] feature of V; it does not delete the categorial [D] feature of D, which is interpretable.

Note 2:

The DP is a complement of V (CompV), not a specifier (SpecV). Hence, given the linear precedence statements that hold in English, the DP must show up to the right of V in VP. Since this is not the surface word order, it can be concluded that further derivational steps are necessary in (21). As it stands, the same prediction is made for verbs like *snore*. [It has been argued that there are two classes of intransitive verbs – *unaccusative* intransitive verbs, and *unergative* intransitive verbs. The former are then assumed to be merged in the complement position of V, as in (21), whereas the latter are merged in the position otherwise reserved for the external argument in transitive constructions. Chomsky (1995, 315) suggests that “unergative verbs are hidden transitives”, with an unexpressed DP in the complement position.]

(22) *A LA of a VP with a transitive verb:*

- a. likes: { [V], [3pers,-pl,+fem], [+fin,-past], [*D*] > [*D*], [*acc*] }
- b. she: { [D], [3pers,-pl,+fem], [nom] }
- c. him: { [D], [3pers,-pl,+masc], [acc] }

(23) *A derivation of a VP with a transitive verb:*

- a. Merge ([D him] , [v likes]) → [VP [v likes] [DP him]]

- b. Merge ([D she], [VP [V likes] [DP him]]) → [VP [DP she] [V' [V likes] [DP him]]]

Problem:

Is there anything that would preclude the reverse application of Merge operations?

(24) *An unwanted derivation for a VP with a transitive verb:*

- a. Merge ([D she], [V likes]) → [VP [V likes] [DP she]]
 b. Merge ([D him], [VP [V likes] [DP she]]) → [VP [DP him] [V' [V likes] [DP she]]]
 (Intended meaning: 'He likes her')

Solution:

Transitive verbs have Case features that play a role similar to selectional features. There are two types of object Cases – *structural* Case and *inherent* Case. ("Case" is written with a capital "C" in order to indicate that the case is *abstract*, not necessarily morphologically visible.) Inherent Case is tied to a certain selectional feature (hence, to a certain Θ -role in the argument structure); structural Case is not. The distinction is particularly relevant in languages like German. The accusative Case of a regular transitive verb is structural. It does not have to go hand in hand with a specific selectional feature (although it usually does); see (25-a) (where *ihn* satisfies a selectional [*D*] feature of V) vs. (25-b) (where *ihn* does not).

(25) *Structural accusative Case in German:*

- a. dass [VP sie ihn mag]
 that she_{nom} him_{acc} likes
 b. dass [VP sie [ihn gehen] laesst]
 that she_{nom} him_{acc} go lets

On the other hand, the dative Case of a simple transitive verb in German is not regular; it is tied to a certain selectional feature:

(26) *Inherent dative Case in German:*

- a. dass [VP sie ihm hilft]
 that she_{nom} him_{dat} helps
 b. *dass [VP sie [ihm gehen] laesst]
 that she_{nom} him_{acc} go lets

Another difference between inherent and structural Cases emerges in *passive* constructions in German: Inherent Case is maintained, structural Case disappears.

(27) *Structural vs. inherent Case in passive constructions:*

- a. dass er/*ihn gemocht wird
 that he_{nom}/him_{acc} liked Aux
 b. dass *er/ihm geholfen wird
 that he_{nom}/him_{dat} helped Aux

(28) *Features of transitive verbs in German:*

- a. mag: { [V], [3pers,-pl,+fem], [+fin,-past], [*D*] > [*D*], [*acc*] }

- b. hilft: { [V], [3pers,-pl,+fem], [+fin,-past], [*D*,*dat*] > [*D*] }

Note:

Like other selectional features, [*Case*] features are uninterpretable; so are [Case] features (in contrast to categorial features). Given the Interpretability Condition, they must be removed from the derivation. Assumption: The removal must proceed as quickly as possible. As soon as V can delete its [*acc*] feature, and DP's [acc] feature, by Merge, it must do so. This rules out the derivation in (24).

(29) *The Timing of Feature Deletion:*

Uninterpretable features are deleted as soon as possible.

4. **Determiner Phrases and Noun Phrases**

(30) *A LA of a simple DP in English:*

- a. a: { [D], [3pers,-pl,+fem], [nom], [*N*] }
 b. woman: { [N], [3pers,-pl,+fem], [nom] }

(31) *A derivation of a simple DP in English:*

Merge ([D a], [N woman]) → [DP [D a] [NP woman]]

Note:

Since D and N both have (uninterpretable) [Case] features, not (uninterpretable) selectional [*Case*] features, there is no deletion of [Case] features involved in the Merge operation; only the [*N*] feature of D is deleted.

(32) *A LA of a more complex DP in German:*

- a. manchen: { [D], [3pers,+pl,-masc,-fem], [+dat], [*N*] } (some)
 b. des: { [D], [3pers,-pl,+masc], [+gen], [*N*] } (the)
 c. Interpretationen: { [N], [3pers,+pl,-masc,-fem], [+dat], [*D*] } (interpretations)
 d. Vorschlags: { [N], [3pers,-pl,+masc], [+gen], } (proposal)

(33) *A derivation of a more complex DP in German:*

- a. Merge ([D des] [N Vorschlags]) → [DP [D des] [NP Vorschlags]]
 b. Merge ([N Interpretationen], [DP [D des] [NP Vorschlags]]) →
 [NP [N Interpretationen] [DP [D des] [NP Vorschlags]]]
 c. Merge ([D manchen], [NP [N Interpretationen] [DP [D des] [NP Vorschlags]]])
 → [DP [D manchen] [NP [N Interpretationen] [DP [D des] [NP Vorschlags]]]]

(34) *Phrase markers that cannot be generated:*

- a. *_{[DP [D manchen] [NP [N Interpretationen] [DP [NP Vorschlags] [D des]]]]}
 (*Linearization Constraint on Merge)
 b. *_{[DP [D manchen] [NP [DP [D des] [NP Vorschlags]]] [N Interpretationen]]}
 (*Linearization Constraint on Merge)

- c. * $[_{DP} [_{NP} [_{N} \text{ Interpretationen }] [_{DP} [_{D} \text{ des }] [_{NP} \text{ Vorschlags }]]] [_{D} \text{ manchen }]]$
 (*Linearization Constraint on Merge)
- d. * $[_{DP} [_{NP} [_{DP} [_{NP} \text{ Vorschlags }] [_{D} \text{ des }]] [_{N} \text{ Interpretationen }]] [_{D} \text{ manchen }]]$
 (*Linearization Constraint on Merge)
- e. * $[_{DP} [_{D} \text{ manchen }] [_{DP} [_{D} \text{ des }] [_{NP} [_{N} \text{ Interpretationen }] [_{NP} \text{ Vorschlags }]]]]$
 (*Economy Constraint on Merge)
- f. * $[_{NP} [_{N} \text{ Vorschlags }] [_{NP} [_{N} \text{ Interpretationen }] [_{DP} [_{D} \text{ manchen }] [_{DP} \text{ des }]]]]$
 (*Economy Constraint on Merge)

Problem:

Is there anything that would preclude the application of Merge operations in (35)?

(35) *An unwanted derivation for a more complex DP in German:*

- a. Merge ($[_{D} \text{ manchen }]$, $[_{N} \text{ Vorschlags }]$) \rightarrow $[_{DP} [_{D} \text{ manchen }] [_{NP} \text{ Vorschlags }]]$
- b. Merge ($[_{N} \text{ Interpretationen }]$, $[_{DP} [_{D} \text{ manchen }] [_{NP} \text{ Vorschlags }]]$) \rightarrow
 $[_{NP} [_{N} \text{ Interpretationen }] [_{DP} [_{D} \text{ manchen }] [_{NP} \text{ Vorschlags }]]]]$
- c. Merge ($[_{D} \text{ des }]$, $[_{NP} [_{N} \text{ Interpretationen }] [_{DP} [_{D} \text{ manchen }] [_{NP} \text{ Vorschlags }]]]$)
 \rightarrow $[_{DP} [_{D} \text{ des }] [_{NP} [_{N} \text{ Interpretationen }] [_{DP} [_{D} \text{ manchen }] [_{NP} \text{ Vorschlags }]]]]]]$

Solution:

This derivation violates an agreement constraint on Merge that holds in German:

(36) *Agreement Constraint on Merge with D:*

D can only be merged with N if the two items have identical Case and ϕ -features.

Note:

So far, the assumption is that personal pronouns like *he* are D elements, hence, DPs when they occur as arguments of predicates. Expressions like *a woman* are also DPs. In line with this, selectional features of verbs so far only included [$*D^*$], not [$*N^*$].

Problem:

What about proper names (like *John*, *Madonna*) or indefinite common nouns in the plural (so-called “bare plurals”, like *books*)?

(37) *Proper names and bare plural common noun phrases:*

- a. She likes John
- b. She likes books about Madonna

Assumption:

[$*N^*$] cannot be a selectional feature on a predicate; but [$*D^*$] can be. The NPs in (37) are selected by empty D elements. If there is no empty D selecting, e.g., *John* in the initial LA, (37-a) cannot be generated by a derivation.

(38) *LA of (37-a):*

- a. likes: { $[V]$, $[3\text{pers}, -\text{pl}, +\text{fem}]$, $[+\text{fin}, -\text{past}]$, $[*D^*] > [*D^*]$, $[*\text{acc}^*]$ }

- b. she: { [D], [3pers,-pl,+fem], [nom] }
- c. John: { [N], [3pers,-pl,+masc], [acc] }
- d. \emptyset : { [D], [3pers,-pl,+masc], [acc], [*N*] }

(39) *Derivation of (37-a):*

- a. Merge ([D \emptyset], [N John]) \rightarrow [DP [D \emptyset] [NP John]]
- b. Merge ([DP [D \emptyset] [NP John]], [V likes]) \rightarrow [VP [V likes] [DP [D \emptyset] [NP John]]]
- c. Merge ([D she], [VP [V likes] [DP [D \emptyset] [NP John]]]) \rightarrow
[VP [DP she] [V' [V likes] [DP [D \emptyset] [NP John]]]]

(40) *LA of (37-b):*

- a. likes: { [V], [3pers,-pl,+fem], [+fin,-past], [*D*] > [*D*], [*acc*] }
- b. she: { [D], [3pers,-pl,+fem], [nom] }
- c. \emptyset : { [D], [3pers,-pl,+fem], [acc], [*N*] }
- d. Madonna: { [N], [3pers,-pl,+fem], [acc] }
- e. books: { [N], [3pers,+pl,-fem,-masc], [acc], [*P:about*] }
- f. about: { [P], [*D*], [*acc*] }

Note:

The selectional feature [*P:about*] on the noun *books* is optional (as is the feature [*D*] on the noun *Interpretationen* in the German example above). This is due to a general difference between V and N: Whereas arguments of V are typically obligatory, arguments of N are optional.

- (41) a. (i) [VP Mary describes John]
(ii) *[VP Mary describes]
(iii) *[VP Describes John]
- b. (i) [DP Mary [D 's] descriptions of John]
(ii) [DP \emptyset descriptions of John]
(iii) [DP Mary [D 's] descriptions]

(42) *Derivation of (37-b):*

- a. Merge ([D \emptyset], [N Madonna]) \rightarrow [DP [D \emptyset] [NP Madonna]]
- b. Merge ([P about], [DP [D \emptyset] [NP Madonna]]) \rightarrow
[PP [P about] [DP [D \emptyset] [NP Madonna]]]
- c. Merge ([N books], [PP [P about] [DP [D \emptyset] [NP Madonna]]]) \rightarrow
[NP [N books] [PP [P about] [DP [D \emptyset] [NP Madonna]]]]
- d. Merge ([V likes], [NP [N books] [PP [P about] [DP [D \emptyset] [NP Madonna]]]]) \rightarrow
[VP [V likes] [NP [N books] [PP [P about] [DP [D \emptyset] [NP Madonna]]]]]
- e. Merge ([D she],
[VP [V likes] [NP [N books] [PP [P about] [DP [D \emptyset] [NP Madonna]]]]]) \rightarrow
[VP [DP she] [V' [V likes] [NP [N books] [PP [P about] [DP [D \emptyset] [NP Madonna]]]]]]

5. Tense Phrases

5.1. T

Problem:

Sentences like those in (43) cannot yet be generated. Auxiliaries and modal verbs do not form a LI together with the main verb (adverbs can intervene).

- (43) a. John will (probably) arrive
b. She has (probably) taken the book
c. Mary can (probably) read it

Assumption:

We can assume that auxiliaries and modals have a selectional feature [$*V^*$]; so they merge with VPs. The categorial feature of *will* and *can* can be assumed to be [T] (for “Tense”), not [V].

(44) LA of (43-a):

- a. arrive: { [V], [-fin,-part], [$*D^*$] }
b. John: { [N], [3pers,-pl,+masc], [nom] }
c. \emptyset : { [D], [3pers,-pl,+masc], [nom], [$*N^*$] }
d. will: { [T], [3pers,-pl,+masc], [+fin,+fut], [$*V^*$, $*-fin^*$, $*-part^*$] }

(45) An unsuccessful derivation of (43-a):

- a. Merge ([D \emptyset], [N John]) \rightarrow [DP [D \emptyset] [NP John]]
b. Merge ([V arrive], [DP [D \emptyset] [NP John]]) \rightarrow [VP [V arrive] [DP [D \emptyset] [NP John]]]
c. Merge ([T will], [VP [V arrive] [DP [D \emptyset] [NP John]]]) \rightarrow
[TP [T will] [VP [V arrive] [DP [D \emptyset] [NP John]]]]

(46) LA of (43-b):

- a. taken: { [V], [-fin,+part], [$*D^*$] > [$*D^*$], [$*acc^*$] }
b. she: { [D], [3pers,-pl,+fem], [nom] }
c. the: { [D], [3pers,-pl,-fem,-masc], [acc], [$*N^*$] }
d. book: { [N], [3pers,-pl,-fem,-masc], [acc] }
e. has: { [T], [3pers,-pl,+masc], [+fin,+past], [$*V^*$, $*-fin^*$, $*+part^*$] }

(47) An unsuccessful derivation of (43-b):

- a. Merge ([D the], [N book]) \rightarrow [DP [D the] [NP book]]
b. Merge ([V taken], [DP [D the] [NP book]]) \rightarrow
[VP [V taken] [DP [D the] [NP book]]]
c. Merge ([D she], [VP [V taken] [DP [D the] [NP book]]]) \rightarrow
[VP [DP she] [V' [V taken] [DP [D the] [NP book]]]]
d. Merge ([T has], [VP [DP she] [V' [V taken] [DP [D the] [NP book]]]]) \rightarrow
[TP [T has] [VP [DP she] [V' [V taken] [DP [D the] [NP book]]]]]

(48) *LA of (43-c):*

- a. read: { [V], [-fin,-part], [*D*] > [*D*], [*acc*] }
- b. Mary: { [N], [3pers,-pl,+fem], [nom] }
- c. \emptyset : { [D], [3pers,-pl,+fem], [nom], [*N*] }
- d. it: { [D], [3pers,-pl,-fem,-masc], [acc] }
- e. can: { [T], [3pers,-pl,+masc], [+fin,-past], [*V*, *-fin*, *-part*] }

(49) *An unsuccessful derivation of (43-c):*

- a. Merge ([D \emptyset], [N Mary]) \rightarrow [DP [D \emptyset] [NP Mary]]
- b. Merge ([V read], [D it]) \rightarrow [VP [V read] [DP it]]
- c. Merge ([DP [D \emptyset] [NP Mary]], [VP [V read] [DP it]]) \rightarrow
[VP [DP [D \emptyset] [NP Mary]] [V' [V read] [DP it]]]
- d. Merge ([T can], [VP [DP [D \emptyset] [NP Mary]] [V' [V read] [DP it]]]) \rightarrow
[TP [T can] [VP [DP [D \emptyset] [NP Mary]] [VP [V read] [DP it]]]]

Note:

The output of the first Merge operation in the derivation in (49) is not immediately an input for the second operation. In fact, these two operations can apply in either order. Trees that have been created and that then wait for further use in a derivation can be said to belong to the *workspace* of a derivation.

5.2. Move

Solution:

There is displacement of the external argument to the specifier of T (SpecT). This position is the *subject* position.

(Strictly speaking, this is the only sense in which one can talk about subjects in this approach: Subjects are XPs in SpecT, period. Informally, however, one often uses the notion of “subject” when referring to external arguments, or to DPs bearing nominative Case.)

Technically, displacement is brought about by the second basic operation that drives derivations, viz., the operation *Move*. *Move* is a composite operation based on *Merge*. Consequently, all constraints on *Merge* (Economy Constraint, Linearization Constraint) also hold for *Move*.

(50) *Move:*

$\text{Move}(\alpha, \beta) = \text{Merge}(\alpha, \beta)$, with the following additional requirements:

- a. The selectional [*F*] feature of the head α can only be deleted under identity with a corresponding feature on the non-head β in specifier position.
- b. The non-head β is taken from the same phrase marker (not from the LA or from the current workspace).
- c. A trace is left in the original position.

Note on (50-a):

[*F*] features triggering Move are sometimes referred to as “EPP” features (because of a constraint called “Extended Projection Principle” from Chomsky (1982), which inter alia required the filling of SpecT). As an alternative to explicitly restricting [*F*]-deletion to identity with a corresponding feature in a specifier position, one can postulate that the [*F*]-features in question are always lowest-ranked in hierarchies of selectional features: ... > [*F*]. In what follows, this latter assumption will be adopted throughout. (The only potential difference might arise in modification contexts; see below.)

Note on (50-b):

Pure Merge is called *external Merge*, whereas Move is called *internal Merge* in recent work by Chomsky (2001b).

Note on (50-c):

There are two possibilities as to what a trace is.

- (i) A trace is an empty category “t” The relation between a moved item and its trace is indicated by co-indexing.
- (ii) A trace is an exact *copy* of the moved item that is unpronounced at the level of Phonological Form (the *copy theory* of movement).

For present purposes, (i) can be assumed (which may require a minimal modification of the Inclusiveness Condition).

(51) *Extended feature sets of T elements:*

- a. will: { [T], [3pers,-pl,+masc], [+fin,+fut], [*V*, *-fin*,*-part*] > [*D*] }
- b. has: { [T], [3pers,-pl,+masc], [+fin,+past], [*V*, *-fin*,*+part*] > [*D*] }
- c. can: { [T], [3pers,-pl,+masc], [+fin,-past], [*V*, *-fin*,*-part*] > [*D*] }

Consequence:

The derivations in (45), (47), and (49) are incomplete.

(52) *A successful derivation of (43-a):*

- ...
- c. Merge ([T will], [VP [V arrive] [DP [D Ø] [NP John]]]) →
[TP [T will] [VP [V arrive] [DP [D Ø] [NP John]]]]
- d. Move ([DP [D Ø] [NP John]], [TP [T will] [VP [V arrive] [DP [D Ø] [NP John]]]])
→ [TP [DP₁ [D Ø] [NP John]] [T' [T will] [VP [V arrive] t₁]]]

(53) *A successful derivation of (43-b):*

- ...
- d. Merge ([T has], [VP [D she] [V' [V taken] [DP [D the] [NP book]]]]) →
[TP [T has] [VP [DP she] [V' [V taken] [DP [D the] [NP book]]]]]
- e. Move ([DP she],
[TP [T has] [VP [DP she] [V' [V taken] [DP [D the] [NP book]]]]])

→ [TP [DP₁ she] [T' [T has] [VP t₁ [V' [V taken] [DP [D the] [NP book]]]]]]

(54) A successful derivation of (43-c):

...

d. Merge ([T can], [VP [DP [D Ø] [NP Mary]] [V' [V read] [DP it]]]) →
[TP [T can] [VP [DP [D Ø] [NP Mary]] [V' [V read] [DP it]]]]

e. Move ([DP [D Ø] [NP Mary]], [TP [T can] [VP [DP [D Ø] [NP Mary]] [V' [V read] [DP it]]]])

→ [TP [DP₁ [D Ø] [NP Mary]] [T' [T can] [VP t₁ [V' [V read] [DP it]]]]]]

Problem:

Why can only the DP *she* move to SpecT, and not the DP *the book*, in a derivation based on the LA in (46) (repeated here as (55), with the revised entry for *has*)? Compare (53) with (56).

(55) LA of (43-b):

- a. taken: { [V], [-fin,+part], [*D*] > [*D*], [*acc*] }
- b. she: { [D], [3pers,-pl,+fem], [nom] }
- c. the: { [D], [3pers,-pl,-fem,-masc], [acc], [*N*] }
- d. book: { [N], [3pers,-pl,-fem,-masc], [acc] }
- e. has: { [T], [3pers,-pl,+masc], [+fin,+past], [*V*], *-fin*, *+part* > [*D*] }

(56) An unwanted derivation of (43-b):

...

d. Merge ([T has], [VP [D she] [V' [V taken] [DP [D the] [NP book]]]]) →
[TP [T has] [VP [DP she] [V' [V taken] [DP [D the] [NP book]]]]]]

e. Move ([DP [D the] [NP book]],
[TP [T has] [VP [DP she] [V' [V taken] [DP [D the] [NP book]]]])
→ [TP [DP [D the] [NP book]] [T' [T has] [VP [DP she] [V' [V taken] t₂]]]]]]

Solution:

There is an agreement constraint on Merge operations applying to T that is similar to the constraint on Merge operations applying to D given above.

(57) Agreement Constraint on Merge with T:

T can only be merged with D if the two items have identical ϕ -features.

Consequence:

T and DP in SpecT must have identical ϕ -features. Given the LA in (55), this is possible if *she* moves to SpecT, but not if *the book* moves to SpecT.

A new problem:

But what if *the* and *book* had ϕ -features matching those of T in the LA to begin with? Could the accusative DP then move to the subject position SpecT?

Solution 1:

The [**D**] feature on T is replaced with a [**nom**] feature on T, as in (58-b) (vs. (58-a), assumed so far). On this view, T selects a nominative DP rather than a DP. Under present assumptions, this is in fact the only way to remove the uninterpretable [nom] feature on a DP; if T does not bear [**nom**], additional assumptions have to be made to ensure [nom] deletion on DP.

(58) *Two theories about the features of T:*

- a. has: { [T], ... [**V**, **-fin**, **+part**] > [**D**] }
- b. has: { [T], ... [**V**, **-fin**, **+part**] > [**nom**] }

Solution 2:

There is a constraint on Move that permits only movement of the higher (c-commanding) item in cases of ambiguity (superiority, minimality).

Conclusion:

In what follows, solution 1 will be adopted for the sake of simplicity.

5.3. Empty T*The situation so far:*

- (i) Movement to SpecT is required for clauses involving auxiliaries and modals to predict the correct word order.
- (ii) Movement to SpecT does not seem to be required for clauses that contain only a transitive main verb to predict the correct word order; see derivation (23), whose output is repeated here as (59-a).
- (iii) Movement to SpecT does seem to be required for clauses that contain only an intransitive main verb to predict the correct word order; see derivation (21), whose output is repeated here as (59-b).

(59) *Word order in the VP:*

- a. [_{VP} [_{DP} she] [_{V'} [_V likes] [_{DP} him]]]
- b. [_{VP} [_V arrived] [_{DP} he]]

Assumption:

All clauses have a T; but T can be empty.

(60) *An extension of the LA of a VP with a transitive verb in (22):*

- a. likes: { [V], [3pers, -pl, +fem], [+fin, -past], [**D**] > [**D**], [**acc**] }
- b. she: { [D], [3pers, -pl, +fem], [nom] }
- c. him: { [D], [3pers, -pl, +masc], [acc] }
- d. Ø: { [T], [3pers, -pl, +fem], [+fin, -past], [**V**, **+fin**, **-past**] > [**nom**] }

- (61) *An extension of the derivation of a VP with a transitive verb in (23):*
- Merge ([D him], [V likes]) → [VP [V likes] [DP him]]
 - Merge ([D she], [VP [V likes] [DP him]]) → [VP [DP she] [V' [V likes] [DP him]]]
 - Merge ([T Ø], [VP [DP she] [V' [V likes] [DP him]]])
→ [TP [T Ø] [VP [DP she] [V' [V likes] [DP him]]]]
 - Move ([DP she], [TP [T Ø] [VP [DP she] [V' [V likes] [DP him]]]])
→ [TP [DP₁ she] [T' [T Ø] [VP t₁ [V' [V likes] [DP him]]]]]]
- (62) *An extension of the LA of a VP with an intransitive verb in (20):*
- arrived: { [V], [3pers,-pl,+masc], [+fin,+past], [*D*] }
 - he: { [D], [3pers,-pl,+masc], [nom] }
 - Ø: { [T], [3pers,-pl,+masc], [+fin,+past], [*V*,*+fin*,*+past*] > [*nom*] }
- (63) *An extension of the derivation of a VP with a transitive verb in (21):*
- Merge ([D he], [V arrived]) → [VP [V arrived] [DP he]]
 - Merge ([T Ø], [VP [V arrived] [DP he]]) → [TP [T Ø] [VP [V arrived] [DP he]]]
 - Move ([DP he], [TP [T Ø] [VP [V arrived] [DP he]]])
→ [TP [DP₁ he] [TP [T Ø] [VP [V arrived] [DP t₁]]]]

Question:

How does it follow that the subject DP agrees with finite V in English with respect to ϕ -features?

Answer:

- T must always have the same ϕ -features as a DP in SpecT, because of the Agreement Constraint on Merge with T.
- This is sufficient for auxiliary and modal contexts, where the finite verb is in T. For contexts in which the main verb is finite, it follows from the following assumption:

(64) *Selection by empty T:*

An empty T selects a V with ϕ -features that correspond to its own ϕ -features.

5.4. Empirical Evidence*Observation:*

There is independent evidence from constituency tests that the external argument DP must leave the VP. (However, none of these tests is absolutely reliable.)

Displacement test:

If a sequence of words may undergo displacement as a group, they must form an XP.

(65) *Displacement:*

Mary wanted to read the book, and [VP₁ read the book] she did t₁

Deletion test:

If a sequence of words may undergo deletion as a group, they must form an XP.

(66) *Deletion:*

Mary never wanted to read the book, but she did [_{VP} –]

Pro-form test:

If a sequence of words may be replaced with a pro-form, it must form an XP.

(67) *Pro-form:*

Mary can read the book, and [_{VP} so] can John

Pseudo-Cleft test:

If a sequence of words may show up to the right of the inflected form of *be* in pseudo-cleft sentences, they must form an XP.

(68) *Pseudo-Cleft:*

What Mary did was [_{VP} read the book]

Note:

From this, one might conclude that external argument DPs are in fact never part of the VP.

But:

Observation:

There is independent evidence that an external argument DP is merged within VP.

Quantifier floating in French (Sportiche (1988)):

In quantifier floating constructions, only a part of an external argument DP moves to SpecT, stranding DP-material in SpecV.

(69) *Quantifier floating in French:*

- a. [_{DP₁} Tous [_{DP₂} les garçons]] ont [_{VP} t₁ lu ce livre]
 all the boys have read the book
- b. [_{DP₂} Les garçons] ont [_{VP} [_{DP₁} tous t₂] lu ce livre]
 the boys have all read the book

5.5. Passive(70) *Active vs. passive sentences:*

- a. John took the book
 b. John has taken the book
 c. The book was taken (by John)

Assumption:

Passivization is an operation in the lexicon affecting V: In English, passivization involves

(i) the deletion of the highest-ranked Θ -role of the argument structure (hence, of the lowest-ranked selectional feature); and (ii) the deletion of an [**acc**] feature (if one is present). Passivization is indicated by [*pas*] on the passivized V. (The optional *by*-phrase is notoriously difficult to handle formally, due to its unclear status as an argument or modifier, and will be ignored here.)

(71) *Active vs. passive forms of V:*

- a. took: { [V], [3pers,-pl,+masc], [+fin,+past], [**D**] > [**D**], [**acc**] }
- b. taken: { [V], [-fin,+part], [**D**] > [**D**], [**acc**] }
- c. taken: { [V], [-fin,+part], [*pas*], [**D**] }

(72) *A LA of (70-c):*

- a. taken: { [V], [-fin,+part], [*pas*], [**D**] }
- b. the: { [D], [3pers,-pl,-fem,-masc], [nom], [**N**] }
- c. book: { [N], [3pers,-pl,-fem,-masc], [nom] }
- d. was: { [T], [3pers,-pl,-fem,-masc], [+fin,+past], [**V**,-fin*,*+part*,*pas*] > [**nom**] }

(73) *Derivation of (70-c):*

- a. Merge ([D the], [N book]) → [DP [D the] [NP book]]
- b. Merge ([V taken], [DP [D the] [NP book]]) →
[VP [V taken] [DP [D the] [NP book]]]
- c. Merge ([T was], [VP [V taken] [DP [D the] [NP book]]])
→ [TP [T was] [VP [V taken] [DP [D the] [NP book]]]]
- d. Move ([DP [D the] [NP book]],
[TP [T was] [VP [V taken] [DP [D the] [NP book]]]])
→ [TP [DP₁ [D the] [NP book]] [T' [T was] [VP [V taken] t₁]]]

Note 1:

The auxiliary *has* does not have the feature [**pas**]; hence, it can never select a passive VP.

Note 2:

There are other instances of *was*; compare (74-abc). In these cases, *was* lacks the feature [**pas**], and either does have the feature [**V**] (but rather [**N**], [**P**] or [**A**]), or it has [**V**] plus [**ing**].

(74) *Other instances of be:*

- a. John was standing in the shadow
- b. Mary was a nice woman
- c. John was in Texas
- d. Mary was proud of John

Question:

Why does the sole DP argument of a passivized transitive construction have to be [nom],

and cannot be [acc] in the LA?

Answer: [Case] on a DP is uninterpretable and must be deleted in syntax. If *the* and *book* bear the feature [acc] in the LA in (72), both [acc] and [*nom(Spec)*] on T will not be deleted, and the Interpretability Condition is violated.

6. Complementizer Phrases

Problem:

How can sentences like those in (75) be generated?

- (75) a. I think that she likes John
 b. I think she likes John
 c. She wonders whether John likes her

Assumption:

LIs like *that* and *whether* in (75-ac) are C (complementizer) elements that select T. (75-b) has an empty C element that also selects T.

(76) *LA of (75-a):*

- a. likes: { [V], [3pers,-pl,+fem], [+fin,-past], [*D*] > [*D*], [*acc*] }
 b. think: { [V], [1pers,-pl,+masc], [+fin,-past], [*C*], [*-wh*] > [*D*] }
 c. John: { [N], [3pers,-pl,+masc], [acc] }
 d. she: { [D], [3pers,-pl,+fem], [nom] }
 e. I: { [D], [1pers,-pl,+masc], [nom] }
 f. Ø: { [D], [3pers,-pl,+masc], [acc], [*N*] }
 g. Ø: { [T], [3pers,-pl,+fem], [+fin,-past], [*V*], [*+fin*], [*-past*] > [*nom*] }
 h. Ø: { [T], [1pers,-pl,+masc], [+fin,-past], [*V*], [*+fin*], [*-past*] > [*nom*] }
 i. that: { [C], [-wh], [+fin], [*T*], [*+fin*] }

(77) *Derivation of (75-a):*

- a. Merge ([D Ø], [N John]) → [DP [D Ø] [NP John]]
 b. Merge ([DP [D Ø] [NP John]], [v likes]) → [VP [v likes] [DP [D Ø] [NP John]]]
 c. Merge ([D she], [VP [v likes] [DP [D Ø] [NP John]]])
 → [VP [DP she] [V' [v likes] [DP [D Ø] [NP John]]]]
 d. Merge ([T Ø], [VP [DP she] [V' [v likes] [DP [D Ø] [NP John]]]])
 → [TP [T Ø] [VP [DP she] [V' [v likes] [DP [D Ø] [NP John]]]]]
 e. Move ([DP she], [TP [T Ø] [VP [DP she] [V' [v likes] [DP [D Ø] [NP John]]]]])
 → [TP [DP₁ she] [T' [T Ø] [VP t₁ [V' [v likes] [DP [D Ø] [NP John]]]]]]
 f. Merge ([C that],
 [TP [DP₁ she] [T' [T Ø] [VP t₁ [V' [v likes] [DP [D Ø] [NP John]]]]]]) →
 [CP [C that] [TP [DP₁ she] [T' [T Ø] [VP t₁ [V' [v likes] [DP [D Ø] [NP John]]]]]]
 g. Merge ([v think], [CP [C that] [TP [DP₁ she] [T' [T Ø] [VP t₁ [V' [v likes] [DP [D Ø] [NP John]]]]]])

- \emptyset] [NP John]]]]]]])
 \rightarrow [VP [V think] [CP [C that] [TP [DP₁ she] [T' [T \emptyset] [VP t₁ [V' [V likes] [DP [D \emptyset] [NP John]]]]]]]]])
 h. Merge ([D I], [VP [V think] [CP [C that] [TP [DP₁ she] [T' [T \emptyset] [VP t₁ [V' [V likes] [DP [D \emptyset] [NP John]]]]]]]]]))
 \rightarrow [VP [DP I] [V' [V think] [CP [C that] [TP [DP₁ she] [T' [T \emptyset] [VP t₁ [V' [V likes] [DP [D \emptyset] [NP John]]]]]]]]])
 i. Merge ([T \emptyset], [VP [DP I] [V' [V think] [CP [C that] [TP [DP₁ she] [T' [T \emptyset] [VP t₁ [V' [V likes] [DP [D \emptyset] [NP John]]]]]]]]]))
 \rightarrow [TP [T \emptyset] [VP [DP I] [V' [V think] [CP [C that] [TP [DP₁ she] [T' [T \emptyset] [VP t₁ [V' [V likes] [DP [D \emptyset] [NP John]]]]]]]]])
 j. Move ([DP I], [TP [T \emptyset] [VP [DP I] [V' [V think] [CP [C that] [TP [DP₁ she] [T' [T \emptyset] [VP t₁ [V' [V likes] [DP [D \emptyset] [NP John]]]]]]]]]))
 \rightarrow [TP [DP₂ I] [T' [T \emptyset] [VP t₂ [V' [V think] [CP [C that] [TP [DP₁ she] [T' [T \emptyset] [VP t₁ [V' [V likes] [DP [D \emptyset] [NP John]]]]]]]]])

(78) LA of (75-b):

- a. likes: { [V], [3pers,-pl,+fem], [+fin,-past], [*D*] > [*D*], [*acc*] }
- b. think: { [V], [1pers,-pl,+masc], [+fin,-past], [*C*], [*-wh*] > [*D*] }
- c. John: { [N], [3pers,-pl,+masc], [acc] }
- d. she: { [D], [3pers,-pl,+fem], [nom] }
- e. I: { [D], [1pers,-pl,+masc], [nom] }
- f. \emptyset : { [D], [3pers,-pl,+masc], [acc], [*N*] }
- g. \emptyset : { [T], [3pers,-pl,+fem], [+fin,-past], [*V*], [*+fin*], [*-past*] > [*nom*] }
- h. \emptyset : { [T], [1pers,-pl,+masc], [+fin,-past], [*V*], [*+fin*], [*-past*] > [*nom*] }
- i. \emptyset : { [C], [-wh], [+fin], [*T*], [*+fin*] }

Observation:

The derivation of (75-b) is identical to that of (75-a).

Note:

Alternatively, one might assume that verbs like *think* can optionally select T instead of C. Under this assumption, postulation of an empty C element is not necessary. Whether such elements are needed or not must be determined on the basis of independent evidence (e.g., are there constraints that refer to C, CP, or SpecC in examples like (75-b)?).

(79) LA of (75-c):

- a. likes: { [V], [3pers,-pl,+masc], [+fin,-past], [*D*] > [*D*], [*acc*] }
- b. wonders: { [V], [3pers,-pl,+fem], [+fin,-past], [*C*], [*+wh*] > [*D*] }
- c. John: { [N], [3pers,-pl,+masc], [nom] }
- d. she: { [D], [3pers,-pl,+fem], [nom] }
- e. her: { [D], [3pers,-pl,+fem], [acc] }

- f. \emptyset : { [D], [3pers,-pl,+masc], [nom], [*N*] }
- g. \emptyset : { [T], [3pers,-pl,+masc], [+fin,-past], [*V*,*+fin*,*-past*] > [*nom*] }
- h. \emptyset : { [T], [3pers,-pl,+fem], [+fin,-past], [*V*,*+fin*,*-past*] > [*nom*] }
- i. whether: { [C], [-wh], [+fin], [*T*,*+fin*] }

(80) Derivation of (75-c):

- a. Merge ([D \emptyset], [N John]) \rightarrow [DP [D \emptyset] [NP John]]
- b. Merge ([V likes], [D her]) \rightarrow [VP [V likes] [DP her]]
- c. Merge ([DP [D \emptyset] [NP John]], [VP [V likes] [DP her]])
 \rightarrow [VP [DP [D \emptyset] [NP John]] [V' [V likes] [DP her]]]
- d. Merge ([T \emptyset], [VP [DP [D \emptyset] [NP John]] [V' [V likes] [DP her]]])
 \rightarrow [TP [T \emptyset] [VP [DP [D \emptyset] [NP John]] [V' [V likes] [DP her]]]]
- e. Move ([DP [D \emptyset] [NP John]],
[TP [T \emptyset] [VP [DP [D \emptyset] [NP John]] [V' [V likes] [DP her]]]])
 \rightarrow [TP [DP₁ [D \emptyset] [NP John]] [T' [T \emptyset] [VP t₁ [V' [V likes] [DP her]]]]]]
- f. Merge ([C whether],
[TP [DP₁ [D \emptyset] [NP John]] [T' [T \emptyset] [VP t₁ [V' [V likes] [DP her]]]]]])
 \rightarrow [CP [C whether] [TP [DP₁ [D \emptyset] [NP John]] [T' [T \emptyset] [VP t₁ [V' [V likes] [DP her]]]]]]]]
- g. Merge ([V wonders],
[CP [C whether] [TP [DP₁ [D \emptyset] [NP John]] [T' [T \emptyset] [VP t₁ [V' [V likes] [DP her]]]]]]])
 \rightarrow [VP [V wonders] [CP [C whether] [TP [DP₁ [D \emptyset] [NP John]] [T' [T \emptyset] [VP t₁ [V' [V likes] [DP her]]]]]]]]
- h. Merge ([D she],
[VP [V wonders] [CP [C whether] [TP [DP₁ [D \emptyset] [NP John]] [T' [T \emptyset] [VP t₁ [V' [V likes] [DP her]]]]]]]])
 \rightarrow [VP [DP she] [V' [V wonders] [CP [C whether] [TP [DP₁ [D \emptyset] [NP John]] [T' [T \emptyset] [VP t₁ [V' [V likes] [DP her]]]]]]]]]]
- i. Merge ([T \emptyset],
[VP [DP she] [V' [V wonders] [CP [C whether] [TP [DP₁ [D \emptyset] [NP John]] [T' [T \emptyset] [VP t₁ [V' [V likes] [DP her]]]]]]]]]])
 \rightarrow [TP [T \emptyset] [VP [DP she] [V' [V wonders] [CP [C whether] [TP [DP₁ [D \emptyset] [NP John]] [T' [T \emptyset] [VP t₁ [V' [V likes] [DP her]]]]]]]]]]]]
- j. Move ([DP she],
[TP [T \emptyset] [VP [DP she] [V' [V wonders] [CP [C whether] [TP [DP₁ [D \emptyset] [NP John]] [T' [T \emptyset] [VP t₁ [V' [V likes] [DP her]]]]]]]]]])
 \rightarrow [TP [DP₂ she] [T' [T \emptyset] [VP t₂ [V' [V wonders] [CP [C whether] [TP [DP₁ [D \emptyset] [NP John]] [T' [T \emptyset] [VP t₁ [V' [V likes] [DP her]]]]]]]]]]]]

Note:

Whereas verbs like *think* can only embed [-wh] CPs, and verbs like *wonder* can only embed

[+wh] CPs, a verb like *know* can select either [-wh] CPs or [+wh] CPs.

(81) *Variable selectional features of 'know':*

- a. Mary knows that John likes her
- b. Mary knows whether John likes her

Note:

Nouns can also embed CPs, i.e., N can have a [*C*] feature.

- (82) a. John heard a rumour that you had read this book
 b. Mary expressed the feeling that the meeting should be held

Problem:

The present system generates categories of variable complexity: NPs, DPs, VPs, TPs, CPs (= clauses). Recall that the main task of syntax is to determine what is a grammatical (well-formed) sentence in a given language, and what is not. But what qualifies as a sentence, and what does not? E.g., an embedded CP like *that she likes John* is not a well-formed sentence; neither is an NP like *rumour that you had read this book*, or indeed a VP like *arrived John*. Essentially, the question is: When does a derivation stop?

Solution:

There is a root C element in every LA that selects finite T, and that cannot be embedded any further. A derivation stops, yielding a sentence (that can then be grammatical or ungrammatical), when root C is merged.

(83) *Root Constraint:*

Every LA has a C marked [root]. No LI can have a feature [*root*].

(84) *Extended LA of (43-b) (compare (55)):*

- a. taken: { [V], [-fin,+part], [*D*] > [*D*], [*acc*] }
- b. she: { [D], [3pers,-pl,+fem], [nom] }
- c. the: { [D], [3pers,-pl,-fem,-masc], [acc], [*N*] }
- d. book: { [N], [3pers,-pl,-fem,-masc], [acc] }
- e. has: { [T], [3pers,-pl,+masc], [+fin,+past], [*V*], [*-fin*], [*+part*] > [*nom*] }
- f. Ø: { [C], [root], [-wh], [+fin], [*T*], [*+fin*].

(85) *Derivation of (43-b) (compare (53)):*

...

- e. Move ([DP she],
 [TP [T has] [VP [DP she] [V' [V taken] [DP [D the] [NP book]]]]])
 → [TP [DP₁ she] [T' [T has] [VP t₁ [V' [V taken] [DP [D the] [NP book]]]]]]
- f. Merge ([C Ø],
 [TP [DP₁ she] [T' [T has] [VP t₁ [V' [V taken] [DP [D the] [NP book]]]]]]) →
 [CP [C Ø] [TP [DP₁ she] [T' [T has] [VP t₁ [V' [V taken] [DP [D the] [NP book]]]]]]]]

7. Adjective Phrases

7.1. Adjectives as Predicates

- (86) a. John is nice
b. She was proud of him

(87) *LA of (86-a):*

- a. John: { [N], [3pers,-pl,+masc], [nom] }
b. \emptyset : { [D], [3pers,-pl,+masc], [nom], [*N*] }
c. is: { [T], [3pers,-pl,+masc], [+fin,-past], [*A*] > [*nom*] }
d. nice: { [A], [*D*] }
e. \emptyset : { [C], [root], [-wh], [+fin], [*T*,*+fin*].

(88) *Derivation of (86-a):*

- a. Merge ([D \emptyset], [N John]) \rightarrow [DP [D \emptyset] [NP John]]
b. Merge ([DP [D \emptyset] [NP John]], [A nice]) \rightarrow [AP [A nice] [DP [D \emptyset] [NP John]]]
c. Merge ([T is], [AP [A nice] [DP [D \emptyset] [NP John]]])
 \rightarrow [TP [T is] [AP [A nice] [DP [D \emptyset] [NP John]]]]
d. Move ([DP [D \emptyset] [NP John]], [TP [T is] [AP [A nice] [DP [D \emptyset] [NP John]]]])
 \rightarrow [TP [DP₁ [D \emptyset] [NP John]] [T' [T is] [AP [A nice] t₁]]]]
e. Merge ([C \emptyset], [TP [DP₁ [D \emptyset] [NP John]] [T' [T is] [AP [A nice] t₁]]])
 \rightarrow [CP [C \emptyset], [TP [DP₁ [D \emptyset] [NP John]] [T' [T is] [AP [A nice] t₁]]]]

(89) *LA of (86-b):*

- a. she: { [D], [3pers,-pl,+fem], [nom] }
b. him: { [D], [3pers,-pl,+masc], [acc] }
c. was: { [T], [3pers,-pl,+fem], [+fin,+past], [*A*] > [*nom*] }
d. proud: { [A], [*P:of*] > [*D*] }
e. of: { [P], [*D*], [*acc*] }
f. \emptyset : { [C], [root], [-wh], [+fin], [*T*,*+fin*].

(90) *Derivation of (86-b):*

- a. Merge ([P of], [D him]) \rightarrow [PP [P of] [DP him]]
b. Merge ([A proud], [PP [P of] [DP him]]) \rightarrow [AP [A proud] [PP [P of] [DP him]]]
c. Merge ([D she], [AP [A proud] [PP [P of] [DP him]]])
 \rightarrow [AP [DP she] [A' [A proud] [PP [P of] [DP him]]]]
d. Merge ([T was], [AP [DP she] [A' [A proud] [PP [P of] [DP him]]]])
 \rightarrow [TP [T was] [AP [DP she] [A' [A proud] [PP [P of] [DP him]]]]]
e. Move ([DP she], [TP [T was] [AP [DP she] [A' [A proud] [PP [P of] [DP him]]]]]])
 \rightarrow [TP [DP₁ she] [TP [T was] [AP t₁ [A' [A proud] [PP [P of] [DP him]]]]]]]
f. Merge ([C \emptyset],
[TP [DP₁ she] [TP [T was] [AP t₁ [A' [A proud] [PP [P of] [DP him]]]]]]]]) \rightarrow

[_{CP} [_C \emptyset], [_{TP} [_{DP₁} she] [_{TP} [_T was] [_{AP} t₁ [_{A'} [_A proud] [_{PP} [_P of] [_{DP} him]]]]]]]]]

Note 1:

Similar analyses can be given for N predicates. As it stands, a noun like *woman* must have two lexical entries, corresponding to its use as a predicate or argument.

(91) *Two uses of N:*

- a. The woman likes John
- b. Mary is a woman

(92) *LIs:*

- a. woman: { [N], [3pers,-pl,+fem] }
- b. woman: { [N], [3pers,-pl,+fem], [nom], [*D*] }

7.2. Adjectives as Modifiers

Note:

As it stands, sentences like (93) cannot yet be accounted for.

- (93) a. He met a nice woman
b. She probably likes him

Problem:

The problem is that there does not seem to be a selectional relation between the adjective *nice* and the noun *man*, or between the adverb *probably* and the verb *likes*. Suppose there were such a relation. First, it seems clear that the selectional feature would have to be on *nice* and *probably*, respectively (compare, e.g., the Linearization Constraint on Merge). (See, e.g., Longobardi (2001) on (93-a), Alexiadou (1997) and Cinque (1999) for cases like (93-b)). We would expect that D and T, respectively, would be the head of the projection after Merge. Consequently, D would have to select either A or N depending on whether *nice* is present or not in (93-a); and T would have to select either A or V, depending on whether or not *probably* is present in (93-b). This does not seem plausible.

Assumption:

The system has to be revised so as to permit modification. In particular, the notion of head and the Economy Constraint on Merge must be revised.

(94) *Head:*

- a. If γ has been created by Merge of α and β , α is the *head* of γ (and γ is a *projection* of α) if (i) or (ii) holds:
 - (i) α selects β .
 - (ii) α is semantically modified by β .
- b. A head and its projection share morpho-syntactic features.

Note:

In principle, there might be a situation in which a projection γ has two heads: α selects β , and α semantically modifies β . In practice, this situation does not arise.

(95) *Economy Constraint on Merge* (revised):

Merge can only apply if (a) or (b) holds:

- a. Merge deletes the selectional feature of a LI.
- b. Merge results in semantic modification.

Note:

In addition to the structural notions of complement and specifier, a third structural notion of (syntactic) modifier can now be introduced. It is worth noting that modifiers are not specifiers – the structural position may be the same, but selectional features are not involved in Merge.

(96) *Complements, specifiers, and modifiers* (see (13)):

- a. β is a *complement* of α iff β has been merged with a LI α that selects it.
- b. β is a *specifier* of α iff β has been merged with a non-LI α that selects it.
- c. β is a *modifier* of α iff β has been merged with α , where α is the head and β neither selects nor is selected by α .

Consequence:

The linear precedence statements must be extended to modifiers. For instance, A modifiers typically precede N, P modifiers follow N in English. In addition, preceding A modifiers can be merged with N only if they share ϕ -features and Case features with N. (As before, such agreement becomes much more obvious in more richly inflecting languages like German or Russian.)

(97) *LA of (93-a):*

- a. met: { [V], [3pers,-pl,+masc], [+fin,+past], [*D*] > [*D*], [*acc*] }
- b. he: { [D], [3pers,-pl,+masc], [nom] }
- c. a: { [D], [3pers,-pl,+fem], [acc], [*N*] }
- d. woman: { [N], [3pers,-pl,+fem], [acc] }
- e. nice: { [A], [3pers,-pl,+fem], [acc] }
- f. \emptyset : { [T], [3pers,-pl,+masc], [+fin,+past], [*V*], [*+fin*], [*+past*] > [*nom*] }
- g. \emptyset : { [C], [root], [-wh], [+fin], [*T*], [*+fin*].

(98) *Derivation of (93-a):*

- a. Merge ([A nice], [N woman]) \rightarrow [NP [AP nice] [N woman]]
- b. Merge ([D a], [NP [AP nice] [N woman]]) \rightarrow [DP [D a] [NP [AP nice] [N woman]]]
- c. Merge ([V met], [DP [D a] [NP [AP nice] [N woman]]])
 \rightarrow [VP [V met] [DP [D a] [NP [AP nice] [N woman]]]]
- d. Merge ([D he], [VP [V met] [DP [D a] [NP [AP nice] [N woman]]]])
 \rightarrow [VP [DP he] [V' [V met] [DP [D a] [NP [AP nice] [N woman]]]]]]
- e. Merge ([T \emptyset], [VP [DP he] [V' [V met] [DP [D a] [NP [AP nice] [N woman]]]]]])

- [TP [T Ø] [VP [DP he] [V' [V met] [DP [D a] [NP [AP nice] [N woman]]]]]]
- f. Move ([DP₁ he],
[TP [T Ø] [VP [DP he] [V' [V met] [DP [D a] [NP [AP nice] [N woman]]]]]]) →
[TP [DP₁ he] [T' [T Ø] [VP t₁ [V' [V met] [DP [D a] [NP [AP nice] [N woman]]]]]]]]
- g. Merge ([C Ø], [TP [DP₁ he] [T' [T Ø] [VP t₁ [V' [V met] [DP [D a] [NP [AP nice] [N woman]]]]]]]]) →
[CP [C Ø] [TP [DP₁ he] [T' [T Ø] [VP t₁ [V' [V met] [DP [D a] [NP [AP nice] [N woman]]]]]]]]]]

Note:

The reasoning is similar for (93-b).

8. Prepositional Phrases

Observation 1:

P can act as a predicate.

(99) *P as a predicate:*

John was in Texas

Observation 2:

P can act as a modifier of N or V.

(100) *P as a modifier:*

- a. The woman in Texas likes John
- b. She met him in Paris

9. Outlook

Infinitives:

The structure of infinitival clauses has not been addressed.

General question:

Do we actually need LAs? It seems that the analyses work just as well if the derivation has access to the whole lexicon/morphology component throughout (see work by Collins, Frampton & Gutmann). As it stands, most initially conceivable LAs will lead to derivational crashes. Chomsky suggests that we need LAs because they reduce complexity. If so, can or should there be meta-constraints on LAs that minimize derivational crashes? (A further argument for LAs arises if transderivational or translocal constraints are adopted. See below.)

10. Exercises*Exercise 1:*

Give LAs and derivations of the following three sentences ((101-c) = (93-b), (101-d) = (99)):

- (101) a. Few riders have won the race
b. I suspect that a lengthy discussion will ensue
c. She probably likes him
d. John was in Texas

Exercise 2:

The present system cannot yet correctly generate sentences with ditransitive verbs like (102-a), and sentences with question words like (102-b). Show this (based on appropriate LAs and derivations). Which (ill-formed) sentences are generated instead?

- (102) a. He has given a book to Mary
b. What can I say?

