‘Problems of Projection’: A Note on Chomsky’s (2013) *Lingua* paper.

<page 44, between examples (17) and (20)>

Joseph Galasso (California State University, Northridge~ Linguistics Dept)

[1]. *The most general case of lack of label is successive-cyclic movement.* [2]. *The intermediate steps are of the form* $\{_{\alpha}$ XP, YP$\}$*, where XP can be for example a wh-phrase with YP a CP.* [3]. *The syntactic object α cannot be labeled, but it must be interpreted, if only for theta-marking.* [4]. *If XP raises, then α will be labeled Y, as required.* [5]. *Therefore XP must raise, and successive cyclic movement is forced*].

[1]. [*The most general case of lack of label is successive-cyclic movement*].

Labeling

Starting out with the theory-internal assumption that labeling comes out of a movement operation, which as a result forces **Dynamic antisymmetry** (DA) (cf. Moro 2000, Ott 2011), then, for example, two lexical items within a **set** {…} necessarily come without order. Unordered members (X, Y) make-up the set {X, Y}. So the set {X, Y} (where both are heads) retain their c-command sisterhood status and thus don’t derive order. Or where {XP, YP] (where both are phrases) behave similarly without hierarchy, and don’t derive order[1]. This has been sharpened by Chomsky’s analysis that the theory internal and *ad hoc* configuration of Spec-Head no longer holds, and that the basic configuration is reduced to a {Head-Head} merge relation, whereby one of the two terms {H} must be complete (i.e., where its phi-features* are already drawn from the lexicon, such as with nouns or verbs) (Chomsky BEA 2001 ms. p. 12).[2]

The interest here lies in the fact that one of the two Hs must be replete with features so that it may serve as a Probe in a **probe-goal relation**. In a sense, the traditional Spec-Head has been reduced to a probe-goal relation whereby a symmetrically and unordered merged set {H, H} then

[1] Adjunction processes of {XP, YP} which seem to show asymmetry (thought which still must force an interpretation) have been discussed in the literature (e.g., see BEA p.18).

[2] BEA ‘Beyond Explanatory Adequacy’ ms. *MIT working papers*. May 2001. *Interpretable n, v-roots act as ‘completed’ Heads while uninterpretable Case and Phi features act as probes.

becomes an asymmetrical (ordered) pair <H, XP>[3] and where the modified H (Head) seeks out a goal within XP. The question is how does this asymmetry establish itself?

[1a] Turning to the recent incarnation of the minimalist program (MP) (Chomsky 1995), perhaps the most essential property which has come out is the notion that all *syntactic objects* (SO) are the result of a (re)combinatory operation called **Merge**, whereby SO = {X, Y} is the mere result of an unordered binary merge of two items: SO = the unordered members of (X, Y). In this 'first-instance' merge, SO renders two lexical items (X, Y) as an unordered set {X, Y} (where {X, Y} = {Y, X}). Call this **External merge** (EM) since X remains independent of Y (they are not part of one another), and as defined by set {…} as being unordered, there is no intrinsic order to (X, Y).

But this is not 'Language' as there is yet no hierarchy/order within the set—it still being a primitive membership composed of nothing other than 'flat' symmetric sister relations (in logic, what we call 'logical-and' Λ, as in the Λ-expression: 'I need to buy a and b and c and d' where set{a, b, c, d} are unordered). Hence, this 'first-instance' EM of the set {X, Y} doesn't necessarily give us syntax (but merely a string of items, a lexicon)—viz., there is no hierarchy of the kind required of syntax.

[1b] At the very minimum, what we need as a prerequisite for syntax is an ordered pair <X, Y> (where X precedes Y). This renders the unordered {set} now as an ordered <pair> (see fn. 40). Specifically, the problem with first-instance merge (EM) is that SO is yet to be labeled: viz., *there is a labeling problem*. 'Labeling of {H, XP} requires that the **Head** (H) not be of the form <X, Y>' (Chomsky, *Lingua* paper p. 46). In other words, SO cannot emerge as a simpleton instance of an arrangement of two heads <X, Y>. Some form of **Labeling** must ensure. In order for SO to be labeled, a 'second-instance' merge operation must take place creating **Displacement** (hence, labeling comes via displacement). While displacement yields a hierarchical expression, it does so in a unique way as specified by the theory: 'Labeling differs from other notions in that it is not virtually detectable by direct inspection of expressions' (Chomsky, *Lingua* 37). In other words, labeling via a movement operation is theory-internal.

[3] Using MP terminology, a **membership set** {X, Y} comes freely unordered, while a **pair** <X, Y> is necessarily ordered. When movement/raising is shown of an element within a set (DA), we can speak of an ordered **recursive set** as {X {X, Y}}. (See Galasso 2016 for full account of Merge as related to the development of child syntax).

So, a theory-internal operation is needed that takes us beyond a mere grouping of items of a (first) **membership set** {X, Y} (say, of two equal heads) to a (second) **recursive set** {X, {~~X~~, Y}} (or pair <X, Y>) which establishes an ordered syntax—noting that {X} of {X, Y} raises and gets displaced from its original membership set, this creating hierarchy. Call this displacement operation **Internal merge** (IM) of **H** since one of the two Heads (H) displaced must leave a **Copy** of itself behind. This forms a recursive structure [SO = $\{X_j, \{X_j, Y\}\}$][4]. At this point, the SO can be labeled by the raising of the H {X} of the **Phrase** (P) [XP {X {X, Y}}]. Ordering as seen within sets {X, Y} has now emerged as an ordered pair <X, Y> (x comes first, then y).

[2] [*The intermediate steps are of the form* $\{_{\alpha}XP, YP\}$*, where XP can be for example a wh-phrase with YP a CP*].

In this phase of derivation, $\{_{\alpha}XP, YP\}$ again are sisters, and properly make-up the equivalent membership set of (XP, YP) with no order. Both are potential (independent) projections which must then get later defined based on the actual (spell-out) projections of their contained lexical features. So, to suppose that XP is a 'wh-phrase' and that YP is a CP (which houses the wh-phrase) is tantamount to saying the 'projection' is one thing, and 'that which projects' is another—though the two are intertwined as specified by what type of P can project what kind of X (where X = H(ead), and XP is P(hrase) headed by X. Phrases are projections of Heads as Heads are bundle of features). For example, Chomsky addresses the fact wh-expressions can also remain *in-situ* without raising (and therefore may in fact not constitute CP), as in the example: *They thought JFK was assassinated in which Texas city?* (so-called 'quiz-show' structures) (p 44).

But the problem here is about projection of both [wh-expression & CP] when they form part of a raised constituency, as is found within an intermediate step along successive cycling: viz., if both {XP, YP} are sisters of a set at some intermediate step in the derivation, then how does labeling generate necessary projection? How does the SO get labeled? In order for a P to project, it must be defined by a set of features specific to its H. So, a potential symmetry of {[[wh-Q]…],

[4] Chomsky tends to distinguish the site of the labeling of a syntactic object (SO) as either being placed at some point after movement (IM) (as with successive-cycling movement), or at the place of a merge-based/base-generated point of the derivation (EM)—with the former (IM) XP bearing the subscript {β}, and the latter (EM) {α}. E.g., where raising of XP is involved and where a copy of itself is found in the lower structure, the use of β is employed $\{_{\beta}XP, YP\}$ = [XP [copula [$\{_{\beta}XP, YP\}$]]], as opposed to $[_{\alpha}N\ TP]$ where NP is the product of simple merge. Take {α…β} to form a chain of [SO {α…β}] (or {γ.. {α.. {β}}} that spans three domains). For example, $\{_{\alpha}\ NP_i, \{T, \{_{\beta}\ NP_i, \{v, VP\}\}\}\}$, the highest copy/instance of the NP is in the domain of the entire SO that is labeled α, since every instance of the NP is within this domain. The lower NP instance, however, is not in the domain of β since not every instance of this NP is within this domain.

& [[CP]…]} = {XP, YP} cannot stand. One of the two Ps must raise in order to form DA (as discussed citing Moro, Ott). One question here is why shouldn't YP raise if both {XP, YP} are equal-distant sisters. The raising conditions behind DA should allow either of the two terms to raise. But it seems only XP (the wh—phrase) raises. (Chomsky expresses this concern in his footnote 36: 'One may ask why YP doesn't raise'). Perhaps the notion of 'computation atom' expressed within the lexical item (LI) has a determining factor behind why the XP (=wh-phrase) of the set {XP, YP} raises leaving {YP} (=CP) without recourse of IM. The CP may be a vacuous phrase (without a head) unlike the wh-expression which is headed—creating an inherent DA within the set even before potential IM takes place for labeling of SO.

Chomsky cites Moro here regarding copular/Small Clause (SC) constructions—e.g., [be [lightning, the cause of fire]] where one of the two terms of the SC must raise so that the SO can be labeled. Hence, DA comes out of a need to label the SO projection.

A very nice example of this is what we find in otherwise symmetric {X, Y} configurations. For example, consider the SO to be of the two lexical items, the membership set: {boat, house}. As it stands, the SO cannot be labeled. What is needed is **raising** thus providing a syntactic hierarchy via a DA-set. (Raising thus renders an unordered membership set into a recursive set). As Moro claims, one of the two terms must raise. If {boat} raises from {X, Y}, we get {boat, {boat, house}} = {$_{\beta}$X, {X, Y}} where the lower X copy is invisible to pronunciation, it being a discontinuous element. Hence, the SO gets labeled and projects as [boat [~~boat~~ house]] (and where hierarchical syntax is generated that allows us to interpret 'boat-house' as a kind of 'house' and not a kind of 'boat'). Such N-compound structures are likened to Adjectival Phrases where the H of the P labels the projection—e.g., [AdjP [Adj black] [N bird]] whereby 'black' is the Head of the phrase, so that the interpretation is that a 'black-bird' is a kind of 'bird' and not a kind of 'black'.

[3] *The syntactic object α cannot be labeled, but it must be interpreted, if only for theta-marking.*

[4] *If XP raises, then α will be labeled Y, as required.*

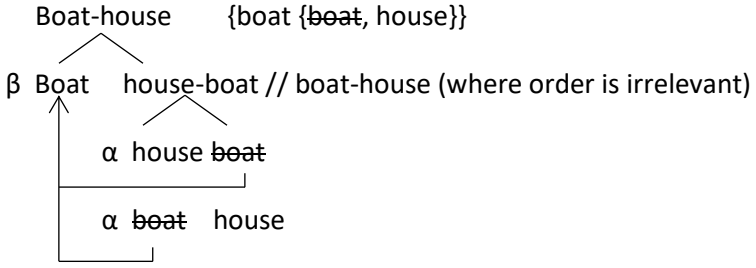
One of the questions regarding 'performed operations on syntax' is essentially 'how does the algorithm operation work? (Chomsky refers to the operation as a labeling algorithm (LA). The problem of labeling is inherent in language, bottom- up vs. top-down solutions have been jousted from the very conception of the problem: viz., (i) in order to label a word, it must be embedded in a top-down structure (language is structure-dependent), (ii) in order to generate a structure, there must be a well-defined status of word. (The catch-22 is similar to what we find in our approximate understanding of the way genetics works: proteins create amino acids and amino acids create proteins, etc.).

Here is how the problem is stated. A labeling algorithm (LA) must scan a syntactic object (SO) and provide a label for one of the two (within a binary selection). So, suppose SO = {H, XP} (H = head, XP= non-head) is rendered via the **external merge** (EM) of two items. Then LA will select H as the label. In other words, the feature specificity for labeling is already encoded in the pair, and projection is straightforward since the H is visible to the LA—e.g., H = verb, and XP = noun complement, [VP [V, NP]]. But what if LA scans two sisters {X, Y}, or {XP, YP}, where both terms are non-heads? Here, the search is ambiguous. It is in this case, when the label {H} is not made available in the search, that the MOVE-property of raising (**internal merge** (IM)) must establish an hierarchical configuration, breaking symmetry. So, consider {XP, YP} as a mere membership set. In order to break with symmetric sister-relations, (since both reside within a flat structure), the raising of one of the two terms must proceed, yielding {XP, {XP, YP}} => [XP [~~XP~~, YP]]. A pair is created with derived syntax (order) <XP, YP>.

Antisymmetry in Compounds

Consider {X, Y} as two lexical items: e.g., the twin-noun formation of [N, N] leading to the compound {boat, house} [N boat-house]. In order for LA to label one of the two SOs as the (adjective-like) **modifier** (M) (in marking the compound 'boat-house' vs. 'house-boat') one of the two SOs must raise in order to break the symmetric reading. In other words, the labeling {N, N} would force one of the two Ns to raise thus allowing the M to project, and as a result, allow the H to be labeled accordingly: {N {~~N~~, N}}, where the moved item acts in an adjectival M capacity and where the residual unmoved item becomes H by default. As we see, what is different about lexical items/*compounds* as compared to English *Phrase* types is that Heads of an **English Phrase** are **Head initia**l e.g., [VP V [~~V~~, N]], whereas Heads of English lexical **compounds** are **Head-final** (as opposed to Spanish compounds where they are **Head-initial**—e.g., *comi-ratas* (eater-rat) (= rat-eater). In the examples below regarding 'boat-house' what needs to be selected is not H, but rather the modifier (M) of H. The item which stays in place, not a result of MOVE, will thus be labeled as Head. The item which raises will become the Modifier of H.

[5] boat-house (a kind of house)



Within the flat structure [5α], order is symmetric and irrelevant. MOVE (IM) (raising) is a unique displacement property which allows for recursive structures found in [5β] above, thus providing a mechanism for labeling the head of a phrase. A 'boat-house' is a kind of 'house' (and not a kind of 'boat'), where order becomes crucial to the compound's reading.

Where X modifies the head—This same formation is exactly what happens in Adjective Phrase (AdjP) structures: So, compound antisymmetry looks something like the following {X {~~X~~, Y}} where {X} gets labeled as M, and as a result, where in-situ Y takes on a head status. In the N-compound 'house-boat', the H of the compound is the N 'boat' (= a kind of boat).

[6] red car.

Where the two lexical items {X, Y} [X, Y] = [red, car]. In this formation, no word order is realized since both items are sisters and are thus ambiguous in their reading. It is not until some raising takes place via IM that order can be assumed: [red, car] > [red [~~red~~, car]] = [AdjP red [~~red~~, car]][5].

***Compounds: Root vs. Synthetic* (cf. §1 [15]).**

[7] a. coffee-maker = (maker of coffee) => [coffee-[maker of ~~coffee~~]]

cigarette smoker = (smoker of cigarettes)

=> [cigarette-[smoker of ~~cigarettes~~]]

b. chain-smoker (not *smoker of chains)=> *[chain-[smoker of ~~chains~~]]

Where examples in [7a] (referred to as **Synthetic compounds**) require *double merge* via displacement: [coffee-[maker of ~~coffee~~]].

And where example [7b] (referred to as **Root compounds**) is the result of a *single merge*: [chain-smoker]

In [7b], the adjectival root is not derived via displacement (notwithstanding the fact that some first-order displacement had to have been initially carried out in order to derive the order {chain, smoker}, since the two items must derive an ordered pair: we don't equally get *{smoker,

[5] We may freely substitute {}-brace brackets with []-square brackets here as the latter is typically used in syntactic tree diagramming e.g., [VP [V, N]].

chain}. Hence, root compounds begin their journey with at least a first-instance merge of a recursive set, but they do not go beyond that in forming further displacement up the tree.[6]

In sum, it does seem that at least some amount if movement is involved with the formation of all types of compounds. For one, the placement of the Head of the compound follows similar parameterization constraints which are found in syntax (e.g., the Head Parameter [+/- Head initial]). As well, the fact as shown above that *synthetic compounds* are the result of MOVE (which contrasts with *root compounds* which are not derived via MOVE) suggests that displacement in language is of a morphosyntactic nature.

(See web-link no. 27. for movement in compounds).

(See https://www.academia.edu/34403441/Working_Papers_4).

Dynamic Antisymmetry

Following a version of Andrea Moro's dynamic antisymmetry (DA) (as presented in Chomsky's 2013 *Lingua* paper), let's consider that we have reached a point in the derivation where [XP, YP] have already been formulated, leavening us with only two Heads {if, how} to contend with, both having identical labeling. What DA stipulates is that in order for one of the terms [XP, YP] to be labeled and associated with the right head, one of the two [XP, YP] terms must raise.

So, if Syntactic Object (SO) = {XP, YP} where both Ps are identical in terms of labeling (both are equal phrases), in order for labeling>projection to be properly headed, one of the two terms (X, Y) would have to raise, thus breaking a symmetrical flat-sister relation.

Consider the [copular-small clause] structure below, where the small clause is of the form [XP, YP] (cited in Chomsky 2013 p. 43 (taken from Andrea Moro's work on *dynamic antisymmetry*)):

[copular (H) [small clause {XP, YP}]] , SO = [XP, YP]]

[6] But this is based upon a theory-internal stipulation which states that 'everything must move at least once in order to become visible as part of labeling'. Accepting this, there still remains the basic assumption that root-compounds are indeed base-generated and are not derived via movement. In Galasso (2016, p. 79), drawing on the '*everything must move at least once*' stipulation, this first-instance of move—as shown in root compounds, as well as all phrases – is referred to as **first-merge (Move-1)**. A reference to **second-merge (Move-2)** then is made for subsequent second-level raisings which yields syntactic compounds.

[8] [be [lightning, the cause of the fire]]

It seems the structure above would yield an ambiguous result if search were applied as is. In other words, in order to label SO (small clause), one of the two terms must raise, breaking its symmetric/sister relation. In this projection, the copular 'Be' triggers one of the two terms of the small clause to raise from out of the identical [XP, YP] labeling:

[8'] a. [XP lightning *is* [XP ~~lightning~~, YP the cause of the fire]]

a. [YP the cause of the fire *is* [XP lightning, ~~YP the cause of the fire~~]]

Another salient example to this structure would be the even more stark case of how one of the two terms in an otherwise identical [XP, YP] has to raise.

Consider the structure below:

[9] a. **If** *you keep eating ice-cream*, **how** *can you lose weight*?

b. How can you lose weight if you keep eating ice-cream?

The two heads (H) involved in the labeling of [XP, YP] are [H {how, if}]. Where XP, YP are given:

[XP you keep eating, YP can you lose weight]]

Noting here how both clauses {XP, YP} can be inverted as long as they remain headed by their proper **polarity-item** head (H)—in this case, the H 'if' triggers the [XP you keep eating ice-cream], while the H 'how' triggers the [YP can you lose weight]. Here's how antisymmetric raising might work in such examples (which show an inverted [YP, XP] for illustrative purposes only, since the order of XP, YP is irrelevant within its flat sister-relation [X, Y]:

[10] [[XP you keep eating] [YP can you lose weight]] (where both terms [XP, YP] are identical in label).

Notice their *symmetric* quality: (how can you lose weight if you keep eating? > if you keep eating, how can you lose weight?)

Now, what must happen to break this symmetry is that the head must select one of the two terms [XP, YP] to raise/MOVE, thus allowing for the proper labeling of the phrase—where the two Hs are {H {**if**}} and {H {**how**}}.

(a) [**if, how** [XP you keep eating, YP can you lose weight]]

(a') [**if** [XP you keep eating **how** ~~[XP you keep eating~~, YP can you lose weight]]]

(b) [**how, if** [XP you keep eating, YP can you lose weight]]

(b') [**how** [can you lose weight] **if** [XP you keep eating, ~~YP can you lose weight~~]]].

Of course, we quickly notice that the order of the Heads {how, if} too is irrelevant, with the same projections that follow:

(c) [**how, if**] [XP you keep eating, YP can you lose weight]]

(c') [**how** [YP can you lose weight] **if** [XP you keep eating, ~~YP can you lose weight~~]]

So, it appears based on our discussions of dynamic antisymmetry DA so far, that it is the head (H) which defines and projects all syntactic objects (SO). All relevant information about an SO, a lexical expression, will be provided by a single designate element within it—call it a 'computational atom', which defines the H. So, what is the triggering mechanism involved? It seems the H 'if' has encoded in its lexical head a collection of *computational atoms* (i.e., features) that requires 'if' as a polarity item to trigger raising of only the declarative [XP you keep eating]=> [if [XP you keep eating]], while preventing the projection of *[if [can you lose?]] (where only the H interrogative item 'how' can trigger an auxiliary inversion derived from the lower matrix phrase [<u>how</u> [YP <u>can</u> you…?]] ([_ _ [you can how?]].

Connected to this, consider how one might parse an already formulated expression such as the now infamous Pink Floyd saying (song 'Another brick in the wall' from the album 'The Wall'). Imagine there are two ways to say the saying, of course, depending on which H you select first:

<If you don't eat your meat, you can't have any pudding! How can you have any pudding if you don't eat your meat?>

Again, notice how each polarity expression Head (if, how) selects its own complement phrase via antisymmetric raising. Consider how DA might go about determining the sequence of the Pink Floyd saying. Recall, at this point, the two Hs in question are symmetric (due to their sister relations) and either one or the other could be selected to start the complex sentence: {H, H}, or {if, how // how, if}.

So, the question we turn to now is to ask: What is the mechanism involved whereby the H selects its appropriate XP?

Head as Computational Atoms

What Chomsky suggests here is that each specific H carries a 'collection of atoms', minimal elements of the H which enter into a computation—'relevant information about SO, a lexical item, will be provided by a single designated element within it, call it a 'computation atom' which defines H. Hence, a lexicon is a class of atoms to be computed (p. 41). Let's consider that the two heads {H, H} [if, how] carry their own special properties of atoms which engage in a computation. Suppose the H {how} carries an anaphoric copy of itself which must be found somewhere within the search of [YP], as in [How…copy]. So, one such atomic feature of {how} can be expressed as follows:

[11] {how$_j$….copy$_j$} => [how……[if…..how]]

(along with other traditional features which go into the projection of the wh-element {how}, such as {+interrogative}, {+ aux inversion}, {+polarity express}, {+adverbial/manner}, {CP-projection}, etc.).

Consider below the search mechanism for {how$_j$….copy$_j$} => [how……[if…..how]]:

[12] **{how}**

[if, **how**$_j$] search => [XP [you ~~can~~ have any pudding ~~how~~ $_j$]]

How => [XP can [you __ have any pudding___]]

How => *[YP you don't eat you meat]

***how** you don't eat your meat **if** you can have any pudding.

The H {how} must select appropriate XP {how$_j$….copy$_j$}.

Consider now {if}

[13] **{if}**

[**if**, how] search=> [YP you don't eat your meat]

If => [YP you don't eat your meat]

*If => *[XP can you ~~can~~ have any pudding ~~how~~ $_j$]]

In this case, the H {if} can't select an XP which contains the anaphoric copy {how} found within the scope of search, therefore, the XP becomes what is referred to as *discontinuous* and is not visible to labeling, discontinuous due to it already being either selected by H 'how', or by the fact that the most prominent feature [copy] can't enter into an Agreement (AGR) checking relation of {how$_j$….copy$_j$}.

So, what we are supposing is that since the H {if} carries no such anaphoric copy, but rather only features associated with H of C (e.g, the polarity condition of (if> then), as well as CP-projection), then part of the LA search becomes modified in order to secure SO labeling. What we can say is that while part of the formal checking requirements of {how} must search for a copy of itself, the computational atoms for H {if} require no such copy. It is in this sense that DA might be applied:

{if, how} [XP, YP] (where [XP, YP] are sister relations in the same sense that Moro talks about his small clause formations).

Notice again how the order doesn't apply to sequence of {if, how}. Since the XP, YP is already formulated (as part of knowing the Pink Floyd phrase), the only selection that needs to be made is which H you select in order to trigger the appropriate XP/YP raising:

[14] a. [*If, _ how* [*XP can you have any pudding [YP you don't eat your meat,*]]

b. [*If* [*YP you don't eat your meat,*] *how* [*XP can you have any pudding [YP* ~~*you don't eat your meat*~~]]

In sum: H {How} searches for a copy of itself, finds it in XP, then proceeds to raise XP. {If} doesn't raise XP because the copy of {how} would go unchecked. Hence {if} is rather forced to raise the other sister relation. It is in this case that {if} is less restricted than {how}, and where {how} must restrict itself to search out an anaphoric

copy within the matrix clause. In other words, {how} requires less search ambiguity over {if}. Heads {if, then} (cited below) however would seem to be more on a par with each other (since neither contain a copy of itself in XP). What then drives a particular labeling perhaps has to do with logic.

[15] What of the Hs {if, then}?

The [if…then] checking sequence would work as follows:

e.g., **If** you eat your meat, **then** you can have pudding.

But notice how the two heads can't seem to be reordered:

*then you can have pudding, if you eat your meat.

This would mean that the feature attributed to the H {if} requires it to be an antecedent (in first position) to any anaphoric expression [if…then] => [if..[then]], where the two Heads are not symmetrical/identical in nature (they are not sisters). This is different from what we saw with the Hs {if, how} where both were symmetric sister relations [if, how] (though where {how} contained an anaphoric feature {+copy}.

e.g., [if, [then]] [XP you eat your meat, YP you can have pudding]

It may be understood that the H {then} of the {if, {then}} sequence searches for the prominent element {can} within XP, where {can} contains a conditional element related to the lexical item {then}.

If X, then you can Y: {if X, then Y}. => logical structure

It was this same kind of underlying thematic/logical structure which motivates Move to take place on a Semantic/Thematic level. Recalling that Move/displacement up until now has had two motivations:

(i) syntactic (S), and (ii) phonological (PF).

Well, it may be the case that Move also caters to two sub-types of S structure: (i) syntactic (surface-level, PF) and (ii) thematic/semantic (underlying logical form, LF).

[16] ***Reasons for Move/displacement:***

(i) Phonological (PF)

(ii) Syntax (S)

a. Syntactic (CP)

b. Semantic/Thematic (vP)*

---

*Note. See the end section of this note 'Family of Merge' for a brief discussion of the **Duality of Semantics**. Following Miyagawa (2000), we consider the two Phases of **CP**, and the light verb **vP** to be the phases which serve this duality of semantics—viz., with the highest *functional* projection of CP dealing within the probe-goal relation of AGReement, and the lower *lexical* projection of vP dealing within the probe-goal relation Case. We'll come to consider Case as being somewhat lexical/semantic in nature, presumably loosely associated with theta marking, while we'll consider AGR as being quintessential formal in nature. (Recall that in a phase-base theory (Chomsky 2008), TP is not considered a phase. But we may entertain the notion that a T-feature [T] may either serve as a featural {F} or Affixal {Af} adjunct which can adjoin to either TP or CP (along with affix lowering onto the V), and that as T(ense)/TP is not phasal in nature, the T-feature itself may be free to *percolate* up or down the syntactic tree). We'll propose, loosely following e.g., Miyagawa (2010) and Radford (2009, 2016) in a number of respects, that the duality of semantics, as pegged to Chomsky's notion of Phase-theory (where only CP & vP are considered phases), that the light verb **vP assigns Case** (Case being lexical semantic in nature) and **CP assigns AGR**.

Let's consider below this duality with respect to 'reasons for MOVE': semantic or syntactic?

Semantic Move.

Consider Chomsky's (2013) remarks below:

[17] a. Which books did John read => S/PF

b. 'For which books X, John reads books X' => semantic/thematic.

One of the reasons for Move/displacement here seems to be to get the semantic roles correctly assigned. For example, here the XP {which books} has two semantic roles: it receives its role as object of {read} (as in the expression 'read books'), and it also serves as a distinct interrogative operator, binding the variable in the object position, so that the interpretation is something like 'for which books X, John reads books X'. Showing base-generated structure, then movement, the structure looks like this:

b. ['which books did [John read ~~which books~~]]?

In addition to two types of Move: *Internal* & *External* Merge, Chomsky identifies two types of merge: *Copy* & *Repetition.* Consider the **repetition-structure** of the phrase 'What hit what'?

[18] What hit what?

It seems that the lexical item 'what' takes on two independent arguments {X, Y}, they are repetitions of the PF spell-out, but both are independent: consider the argument structure:

a. [HIT [x, y}]] = X hit Y…e.g., 'Boy hit ball'

Compare (a) above to the **copy-structure** below:

[19] What was hit (what)? (John hit what?)

In [19], 'what' is a copy derived by [__ was hit what] (=~~What~~ was hit what?) with Internal Merge (IM) raising of 'what' to the surface subject position. Consider the underlying structure:

a. The ball was hit
   i. __ hit the ball. (John hit the ball).

In terms of IM, the *raising of a copy* seems to be a **syntactic effect,** whereas *raising of repetition* seems to be a **semantic effect.** This same semantic/thematic vs. syntax split regarding merge can be seen in:

b. Which books did John read?

Here in (b), 'which' takes on two independent semantic roles (via repetition):

(i) Object of 'read'
(ii) Interrogative operator

For 'Which books X', 'John read books X'.

A further complication might actually be suggested that Copy (IM) serves a two-fold operation:

[20] (i) can serve **semantics** when dealing with **Case** (argument/thematic marking), as shown in (b) above, and/or when dealing with a light verb (vP) projection…, or

(i) It can also serve **syntax** when dealing with **Agreement,** when dealing with a CP projection.

Hence, what we get out of Internal Merge (IM) is a potential **double**-**merge** operation of (i) semantics, followed by (ii) syntax.

Copy as IM:
(i) Semantic (vP): Case
(ii) Syntax (CP): AGReement

In sum, what the above dual treatment of merge (IM) details is that while Repetition is triggered by a single-merge (exclusively for semantics), Copy may trigger double-merge (both for semantics (vP-Case), and for syntax (CP-AGR).

(See Miyagawa (2010) for such an account of a Case/AGR double-merge. In fact Miyagawa goes even further and suggests that the entirety of Move is based solely on AGR).

Syntactic Move.

Let's follow-up on Move as having a syntactic effect. The quintessential notion of syntactic move has to do with Agreement (AGR) (following Miyagawa here)—which is a movement-based operation triggered for no other apparent reason other than to check off a formal AGR feature. This is what we find in most instances where the higher non-argument position of CP is involved.

AGR-based Move.

T(ense) only carries the set of (person, number) AGR features in a clause (XP) where T is selected by C of a (C(omplementizer Phrase) (Radford 2009, p. 397).

What this means is that the Head C (as part of its computational atoms) carries and 'hands-over' AGR features from H of C to H of T.

[21] (a) I am hoping [CP [for] [TP him to win]] [-Tense/-Case]

(b) I am hoping [CP [C that] [TP he wins]] [+Tense/+Case]

Let's flesh such a treatment out in terms of DA and H computational atoms as discussed above.

Suppose the two Heads are {H, {for, that}}, and the {XP, YP} are [TP {him to win}] and [YP {he wins}].

What the search reveals in labeling the SO is that the H {for}, carries no Tense/AGR features (from C) and so only the default non-finite verbs along with the default accusative case get selected. In (b), it is the well-formed H of CP {that} which selects and labels for AGR (Tense/Nominative Case). What we have regarding the labeling of XP in terms of the H is precisely the same type of selection as discussed in our treatment of dynamic asymmetry:

[22] {for, that} [XP him to win, YP he wins]

Both {XP, YP} are identical. So in order to label the SO, one of the two terms must be modified by raising.

[for, that} [~~him to win~~, he wins]

(i) [for [him to win]] => search [<u>for him to win</u> [XP ~~him to win~~, (YP he wins)]]

(ii) [that [he wins]] => search [<u>that he wins</u> [XP (him to win),YP ~~he wins~~]]

H {for} carries computational/atomic features [-T, -Case] => [-AGR]

H {that} carries [+T, +Case] => [+ AGR].

Dynamic Antisymmetry & Recursiveness in Possessives {'s}, {of}

John's book vs. Book of John's

Consider the structure {John {Poss, book}}, [John [Poss, book]]. Here John [J] can raise either of the two items [X, Y] (X Poss, Y book]. If J selects (via search) {Poss}, leaving {book} in place, we get the underlying structure [John ['s book]] => [John's [_ book]] where clitic {'s} raises and attaches to the stem (a result of DA movement due to PF considerations, (as presented above in §1 [2], but see §4 [10] where PF clitics don't apply).

If, however, J selects to raise {book}, leaving {Poss} in place (so that it remains frozen in-situ and therefore can't become a clitic), we then get [book of [John ['s, ~~book~~]]. The double possessive elements found in expressions such as *I am a friend of John's* can be traced back to how the H John selects either *Poss* or *Book* to raise, following similar steps as laid out in our discussion of Dynamic Antisymmetry which shows DA merge as [H [X, Y]].

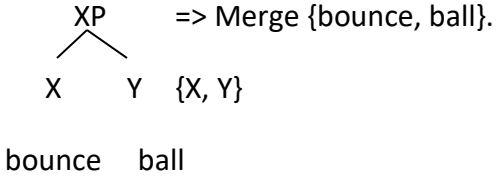
Dynamic Antisymmetry of Internal Merge: Closing Remarks

Various approaches have pinned DA to models which stipulate that 'every element must move at least once' in order to be visible to a labeling algorithm (LA). This reduces to meaning that every right-branching structure must end in a trace (See Kayne (1994). Hence, complements (object DPs) must vacant their base-generated positions. Added to this movement stipulation is that subjects (DP) too must move out of VP, coupled with other language-specific notions determining whether or not a verb must raise to H of T, (or if T affix lowering down to H of V is required, as in English). All in all, the notion of movement, as it entails all the aforementioned operations, is without doubt the crucial mechanism at play in determining fully-fledged human language capacity.

A Note: Family of Merge

As part of our closing remarks, let's recap how the step-sequences of Merge (both internal and external) come together to form a fully articulated syntactic tree, particularly focusing on **Case** and **Agreement**, (or semantics & syntactic, which make-up the so-called **duality of semantics**):

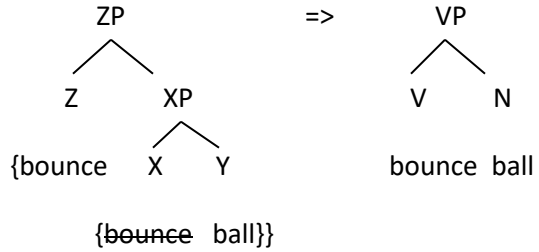
(1) Step-1: External Merge (EM)



At step-1, there is no hierarchical word order. XP could yield either 'ball bounce//bounce ball'. Step-1 is exclusively semantic, whereby two items have simply been pulled from out of the lexicon.

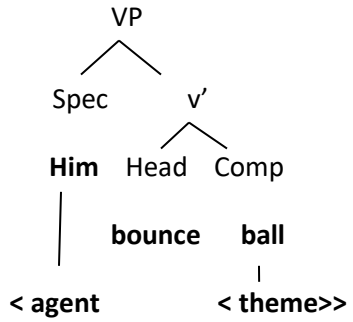
In order to create a hierarchical syntactic phrase, Move is required (our step-2):

(2) Step-2: Internal Merge (IM):



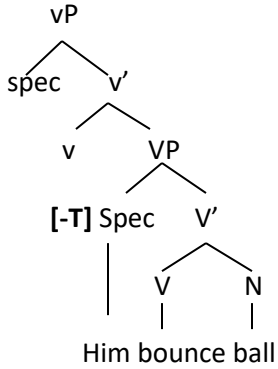
At step-2, we get Internal Merge (IM) (with a non-pronounced copied movement of 'bounce') which now allows order via dynamic asymmetry (DA). It is here that we arrive at a properly formed Spec-Head-Comp. This is a prosaic lexical/semantic projection still without Case, Tense or Agreement.

(2') Spec-Head-Comp configuration (Theta-marking):



Theta marking is a semantic assignment.

(3) Step-3: Light Verb (vP)



The [-T] tense feature may assign the sort of tense we find with [–Finite] imperatives, subjunctives.

Note no Case, AGR at this step.

At step-3 above, what this next merge projection allows is an added available Spec position for the subject of the lower VP to raise into. Subject/Spec of VP, once inserted into Spec vP, gets assigned Case [+Nominative]. Light verbs can assign Case, since vP is semantic in nature and Case is a residual effect of semantics. (vP is said to straddle both lexical and functional projections. Unlike VP which can only map a theta-grid, the light verb vP projection takes the derivation one step further in securing Case. While light verbs such as *make*, *do*, project alongside the main verb e.g., 'John *makes* roll the ball' (= John ø rolls the ball), such light verbs, when stranded within vP, can't project Tense). Light verb formations arguable give us the antiquated **subjunctive** mood (which particularly contrasts with the **indicative**) whereby *nominative Case is assigned without finite Tense*—e.g., the subjunctive sentence: '[I'd suggest that [**he study** for the exams]]') and not (*'He studies…').

(We'd suggest that subjunctives have a fully functioning vP with a somewhat defective TP, whereby case gets assigned by the above [featural T-feature] within T, but where the [affixal-T] feature of the Head is unspecified.)[7]

But before Case can be assigned, it must be handed-over by a [+T] feature (Case is marked via Tense—[+Finite] tense marks [+Nom] case, [-Fin] tense marks [-Nom] case (see below).

Case marking via Tense:

(i) Finite tense assigns [+Nominative] case:

e.g., I think [<u>he walks</u> often] : He => walks

(ii) Non-Finite tense assigns [–Nom]/Accusative case:

e.g., I saw [<u>him walk</u> often]: Him => walk

Since vP can't mark for [+Tense], in order for vP to assign case, a TP must project whose Head projects finite tense. Consider the next step-4 below:

[7] One distinction that can be made between featural vs. affixal is that featural may encode tense as incorporated in the verb stem itself, perhaps as in French, (or English auxiliary and irregular verbs), whereas the affixal-feature is a decomposition of stem+affix.

(4) Step-4: Tense Phrase

(i) TP — Featural-T 'hands-over' case marking feature onto spec vP.

Spec T' — [+T] is an adjunctive in nature and may attach to any phrase.

[+T] vP — Pronoun 'he' enters into a Case-marking configuration with T.

spec v' — 'him' receives [+Nom] case 'He' via T marking.

he v VP

Spec V'

V N

[-T]

~~Him~~ bounce ball

Tense treelet structure: (past tense).

When Tense projects [+Past] {ed}, along with Case marking, we now find affixal-T lowering, as [Stem + [affix]].

(ii) T' (with light verb vP not shown)

[+T/Aff] VP

[{ed}] V — Affixal Tense{ed} lowering.

[bounce]ed]

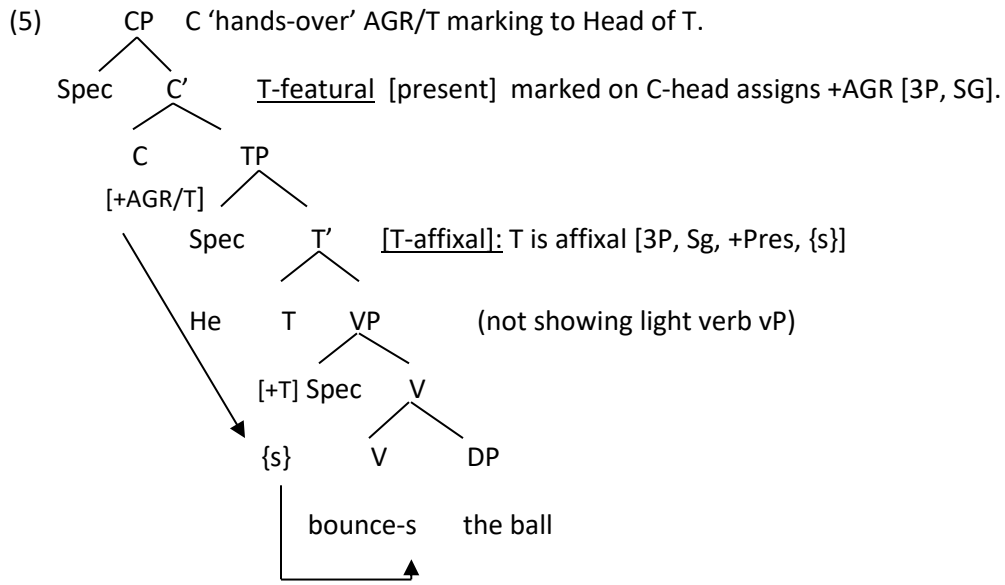
Once Case and Tense are projected via vP, TP respectively, the question now turns to the more formal projection of AGReement (since the two lexical/semantic assignments have both been satisfied). Following some of the remarks made in Radford (2009) about AGR being assigned via C-head of CP, as well as his more recent remarks that C-head may in fact house Tense (with the T-feature assigning AGR down to TP), we similarly suggest the extension, that, like what we find of **wh-subject constructs** (e.g., 'Who found it'?), all declarative TP projections are in fact fully extended CP projections. By proposing that CP assigns AGR, coupled with vP which assigns Case, we arrive at the **duality of semantics** with the added feature that a **phase**-**base** theory (MP) also

incorporates these two phases: CP & vP—with (i) CP being formal/syntactic in nature, and (ii) vP being theta/semantic in nature.

The duality this way is captured by the lexical vP vs. functional CP split, with Case being a residual lexical phenomena and AGR being quintessential formal. In fact, Miyagawa (2000) goes even further by speculating that it is AGR which is solely responsible for MOVE.

Let's consider the last step showing a full CP-articulated structure:

'He bounces the ball' (showing 3P, singular {s} (AGR)):



Subject Agreement (Radford 2016, p. 338):

When T agrees with its Subject/Spec:

(i) The person/number features of the subject are copied onto the Aux/affix T. (We are adding that the AGR projection is instigated by C).
(ii) The tense feature [T] on T is copied onto the subject.

With this treatment, CP 'Wh-subject' constructs of the 'Who bounces it?' type (where the wh-word 'who' must raise to spec of CP) mimic what we find with declarative TPs ('He bounces it'): both are full CP projections since, in this analysis, the AGR of the verb e.g., 'bounces' (of 'He bounces it') can only be assigned via 'hand-over' of C-head [+AGR/T] to T.

Featural vs. Affixal T.

The assumptions made above is that T(ense) comes in two forms: (i) featural and (ii) affixal. When featural, since T is adjunctive in nature, theoretical considerations could be made that T can be found adjoined to any appropriate phrase (not just TP). (Radford (2016, p. 338) goes further to suggest that even DP/Nominals (Nouns) can theoretically take-on a tense feature). Tense has been problematic in the MP literature for quite some time: for instance, it seems to be a [+Interpretable] feature, thus semantic in nature, while, at the same time it seems to be implicated both in Case and Agreement. And Chomsky considers Tense not to be a phase.

What this may call for is the notion that T, when functioning as [+featural], a featural-T can percolate up and down the tree without impunity (when inserted at the **edge** of any phrase) and enter into **probe**-**goal** relations of various kinds (as determined by the nature/feature specificity of the host head). For instance:

(i) when T-featural is adjoined to C, it seems to be implicated in securing AGReement of the subject-verb accord,

(ii) when T-feaural is adjoined to T, it delivers two functions:
   a. it can deliver Case (as found with light verb vP),
   b. it can deliver a non-affixal Tense (such as with bare verb stems, auxiliary do>did, have>had, is>was, as well as irregular verbs go>went, etc).

(iii) when T-affixal: {ed}, {s} [+/-past] affix lowers onto the verb stem.