Foundations of ('old' and 'new') Minimalist syntax

Diego Gabriel Krivochen
University of Oxford
diegokrivochen@hotmail.com

Initial warning

We will divide Minimalism into two rough periods:

1. Chomsky (1995, 2000, 2001, 2004, 2008 -maybe?-)

   And mostly everyone else actually using Minimalism for purposes of doing grammar. What you'll also find in textbooks (Radford, 2009; Hornstein et al., 2005; Adger, 2003…)

2. Chomsky (2013, 2015, 2019, 2020a, 2021)

   And Collins (2017), Epstein et al. (2015, 2022), Fong et al. (2019), and other associates… (what I'll call 'philosophical Minimalism'). Possibly also MCB/MBC?

But first…some basic notions of argument structure

Predicates are classified based on how many arguments they take

0 participants: 'zero-adic' verbs (meteorological Vs)
e.g.: rain, snow, hail…

1 participant: 'monadic' verbs (intransitive Vs)
Only take a *subject*. Two kinds (actually three, but we'll simplify things to what makes sense configurationally):
Unaccusatives: the subject is a *theme*.
e.g., arrive, leave, appear, stand, $grow_1$, $break_1$, locative be, explode, die, blossom…
Unergatives: the subject is an *agent*. Verbs of emission (of light, substance, sound, etc.), manner of motion, intake, etc.
e.g., walk (and all forms of walking: stroll, hop, etc.), shine (and other emission of light verbs), dream (and other 'cognate object' verbs: dream a dream, run a race…), drink (and other 'hyponymic object' verbs: drink a beer, eat a banana…), shower (= have a shower), lunch (= have lunch), dance (= do a dance), shiver…

2 participants: 'dyadic' verbs (monotransitive Vs)
Take a *subject* and an *object* or a *complement clause*
e.g.: hit, $break_2$, $show_1$, like, have, buy, destroy, cherish, rely/depend (on), $order_1$, paint (and other substance smearing verbs), $grow_2$…

3 participants: 'triadic' verbs (ditransitive Vs)
Take a *subject*, an *object*, and an *oblique* (usually a location) or a *complement clause*
e.g.: give, send, lend, $show_2$, $order_2$, tell, mail, pass, load, put…

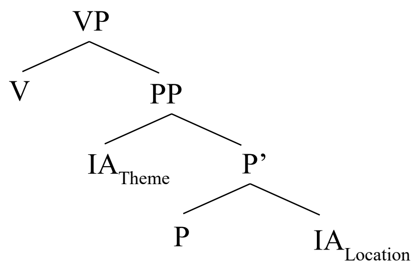
- Other than the subject of unaccusatives (which is underlyingly inside the VP, an 'initial 2' in Relational Grammar terms), we'll refer to subjects as *external arguments*. Everything else will be an *internal argument*.

- After Marantz (1984), Kratzer (1996) and related work, *external arguments* (for concreteness: agentive subjects) are not arguments of the lexical verb, but of a functional category above VP: Kratzer calls it Voice, most other people call it *v* ('little v'). Harley (1995) calls it EventP.
    - Of course, some go farther and say that Voice and *v* are distinct, and also that there's a whole bunch of other stuff there. We won't.
    - *v* is responsible for assigning a thematic role to the external argument, and also for accusative case assignment
        - Basically, Burzio's generalisation in a functional category

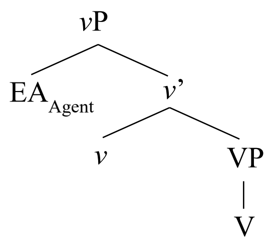
- Importantly, whenever we speak of *subjects*, *objects*, and the like, we're being informal. Grammatical functions play no role whatsoever in Minimalism (and haven't played a role in generative grammar arguably since early GB)
    - In stark contrast, LFG, Relational Grammar, Arc Pair Grammar are all fundamentally based on the idea that grammatical functions are primitives

The rough templates of the lexical verb area we'll work with look like this:

1)

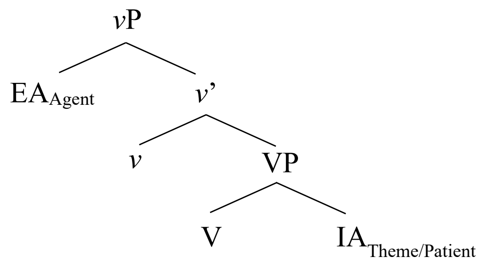


<u>Unaccusative</u> (Hale & Keyser, 2002: 196; Mateu Fontanals, 2002)

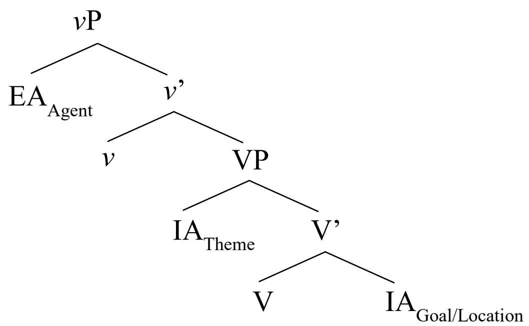


<u>Unergative</u>

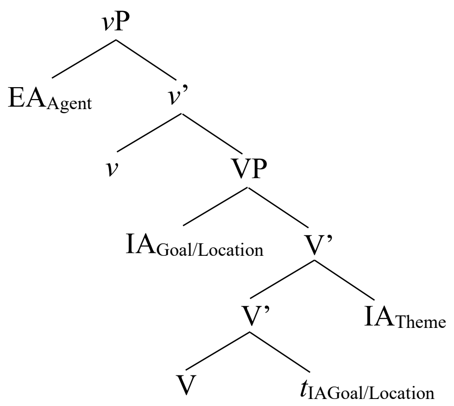
Note: I am simplifying the analysis of unergatives, which being denominal require conflation of an N onto an affixal V head



<u>Monotransitive</u>



<u>PIOC ditransitive</u> (Larson, 1988 style)



<u>DOC ditransitive</u> (Larson, 1988 style)

- There are also lexical analyses of the PIOC/DOC alternation within Minimalism, which are pretty much equivalent to LFG's
    - See, e.g., Harley (1995), Hale & Keyser (2002: 184-185) vs. Lam (2008: 122, ff.), Butt et al. (1999: 60).

Minimalist assumptions

- Language is an optimal solution to the requirements imposed by the systems of sound and meaning
    - Sometimes 'perfect' is used, particularly in the early days (e.g., Chomsky, 2000: 93, 96)
- The idea is not that the *theory* of the faculty of language is 'perfect' or 'optimal', but that FL itself is
    - This 'perfection' is what grounds the idea of *economy*
    - In the early days, this was a twofold dictum: *eliminate superfluous elements in representations and superfluous steps in derivations* (Chomsky, 1995). Don't do more than what is strictly necessary
- Structure building and structure mapping were reduced to a single operation: Merge
    - Perhaps strongest exponent of Chomsky's 'congenital dendrophobia' (Jim McCawley dixit): instead of changing strings for graphs, now we have some version of set theory (not quite ZF, as Gärtner, 2022 among others have shown)
    - This is important: Minimalism has *consistently* and *explicitly* rejected graphs
        - Chomsky (1995: 226; 2020a: 38-39; Collins & Groat, 2018: 2; Collins & Stabler, 2016: 48, ff. -although they use a somewhat mixed metaphor-, etc.)
    - Only difference between structure building and structure mapping is where we get things from:
        - From lexicon to syntax: External Merge (EM)
        - From syntax to syntax: Internal Merge (IM)
- In early Minimalism derivations start with the selection of a collection of lexical items from the lexicon
    - This collection is known as a Lexical Array
    - If each lexical item is indexed with an integer indicating how many times it's used in a derivation (so the collection is a multiset of sorts), we speak of a Numeration

- Derivations must exhaust Numerations…
- …and, not use anything that is not specified in the Numeration (*Inclusiveness Condition*)

- Lexical items are bundles of features
  - Some of these are *interpretable*, some are *uninterpretable*
  - Whether a feature is interpretable or not depends on the category it appears in
    - For example: person and number features are interpretable on N, but not on T
  - Uninterpretable features cannot reach the interfaces: if they do, the derivation *crashes*
    - In the early days (1995-ish), IM was motivated by the need to create configurations where we could get rid of uninterpretable features: Spec-Head relations
    - Things changed *very* quickly (by 1998 already). Chomsky (2000: 126) says '*we should not expect Spec-head relations to have any special status*'.

The derivational machine

Minimalism Part 1:

Merge(X, Y) = {X, {X, Y}}

- Sort of Wiener-Kuratowski-like (but note: {X, {X, Y}}, **not** {{X}, {X, Y}} for some reason)
- Operates over syntactic objects: a SO is a lexical item or a set of LIs
- Merge is accompanied by labelling (Chomsky, 1995: 224; also 2000: 133):

> [Given {γ, {α, β}} created by Merge] *The label γ must be constructed from the two constituents α and β. Suppose these are lexical items, each a set of features. Then the simplest assumption would be that γ is either*
> *a. the intersection of α and β*
> *b. the union of α and β*
> *c. one or the other of α , β*
> *The options (a) and (b) are immediately excluded: the intersection of α, β will generally be irrelevant to output conditions, often null; and the union will be not only irrelevant but "contradictory" if α, β differ in value for some feature, the normal case. We are left with (c): the label γ is either α or β; one or the other projects and is the head of K. If α projects, then K = {α, {α, β}}.*

In this context, phrasal levels are defined contextually:

> *a category that does not project any further is a maximal projection; a category that is not a projection at all is a minimal projection (a lexical item); any other projection is an intermediate projection* (Nunes, 1998: 160)

See also Chomsky (1995: 222-223)

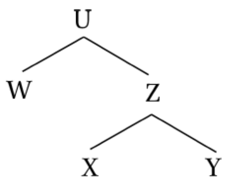
Merge delivers sequential structure building, which operates *bottom-up* (if we visualise things as a tree):

a. {a, {a, b}} (e.g.: {kill, {kill, John}})
b. {c, {c, {a, {a, b}}}} (e.g., {v, {v, {kill, {kill, John}}}})

c. {c, {d, {c, {a, {a, b}}}}}} (e.g., {v, {Bill, {v, {die, {die, John}}}}})

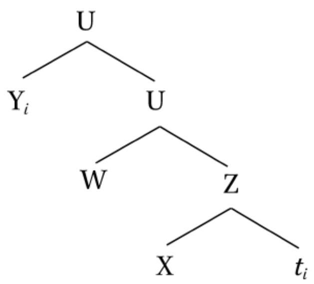
And so on. We don't *expand* an axiom until getting to a terminal string, we *combine* atomic elements (here, lexical items) recursively.

Trees and sets are used interchangeably (almost no one uses sets when doing grammar)



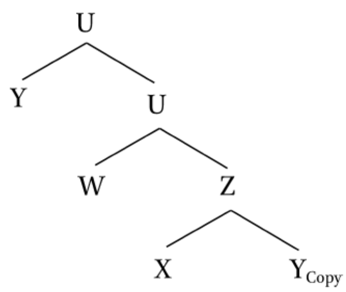
{U, {W, {Z, {X, Y}}}}

a.



{U, {Y_i, {U, {W, {Z, {X, t_i}}}}}}

b.

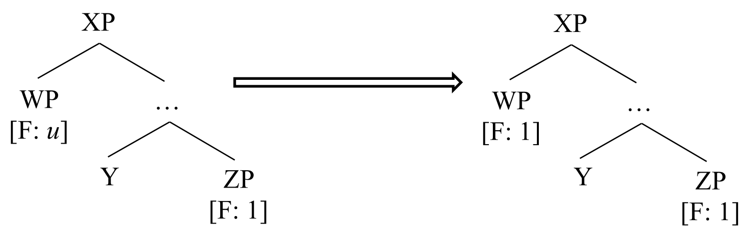


{U, {Y, {U, {W, {Z, {X, Y_{Copy}}}}}}}

<u>Agree</u> (not mentioned in MCB, MBC, Chomsky, 2020a, 2021, Kitahara, 2022, Seely's lectures in 2023, Fong et al., 2019, or the formalisation in Collins & Stabler, 2016)

- Quite alarmingly, since Agree is fundamental in actual grammatical analysis (Chomsky et al., 2019 refer back to Chomsky, 2000, 2001 for Agree, and offer no step by step derivation using Agree with current free Merge. Chomsky, 2020b; Epstein et al., 2022: Chapter 7 assume that Agree works just as in Chomsky, 2001, 2008)
- Merge operates over *terms*. Agree operates over pairs of features: valued in a *probe* – unvalued in a *goal*
- The value of the relevant attribute in the *goal* gets copied onto the corresponding attribute of the *probe*, where *probes always c-command goals*: Agree is fundamentally counter-cyclic (but see Zeijlstra, 2012)

2)



- Classically, there is only one goal per probe, but *multiple Agree* is also a thing (e.g., Hiraiwa, 2001; Zeijlstra, 2012): a single probe, multiple goals.

- We're kinda simplifying things: recently, Agree has been conceptualised as a two-part operation (e.g., Smith et al., 2020: 10):

  Agreement between a controller and target [read: Probe and Goal] *proceeds in two steps:*

a. AGREE-LINK: in the syntax, a target has unvalued phi-features that triggers agree with controller. The result is a link between controller and target.
b. AGREE-COPY: the values of the phi-features of controller are copied onto target linked to it by agree-link

From Deal (2022): conditions over Agree

Structural description: Agree holds between a probe and a goal iff all of the following conditions hold.
a. The probe bears uF: features that are uninterpretable and unvalued. [furthermore, the probe must have a *full* set of features. Otherwise, it's *defective*]
(Probe specification)
b. The probe c-commands the goal.
(Structural condition)
c. The uF of the probe matches with iF of the goal.
(Match condition)
d. The goal is active: it also has uninterpretable features (uF').
(*Activity condition*)
e. The goal is the closest element to the probe meeting the conditions above.
(*Minimality condition*)

Agree applies to:

- Case assignment: NPs have [*u*-Case], which gets valued depending on the functional head they Agree with
  - Nominative if they agree with T
  - Accusative if they agree with *v*
  - All other cases are either Lexical or Inherent (Woodford, 2006), and thus not covered by Agree (but possibly are a side-product of Merge)

- Subject-Verb agreement: NPs have lexically valued [Person] and [Number] features (these are usually grouped under what's called φ-features, including also [Gender]). T has unvalued φ-features, which probe for an NP to Agree with.

These two things are intimately related: Case is valued on an NP depending on the functional head it phi-agrees with.

- Wh-movement: [u-wh] in C probes for a [wh] phrase
  - In Epstein et al. (2022: 113), however, the wh-phrase has u-Q and C has Q. Probe-Goal is replaced by MS, although why C would trigger MS is unclear
  - Epstein et al. also assume that MS *simultaneously* finds the probe and the goal, which, well…

Chomsky (2000, 2001): Agree values the probing features of the probe, and has as a by-product that the uninterpretable features of the goal are also valued.

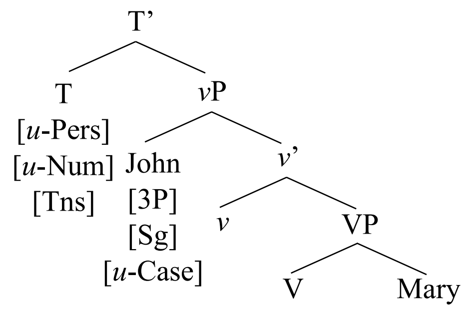
Case itself is not matched, but deletes under matching of φ-features (Chomsky, 2001: 6)

- Clear example: u-φ valuation in T and Case valuation on NP. The same happens with *v* and ACC: structural case is always assigned as a by-product of phi-agreement.

The MI/DbP framework does not view structural case as the uninterpretable counterpart of an otherwise interpretable feature. Instead, it is a sui generis feature with a special relation to the *φ -features: it gets valued only as a by-product of φ-feature agreement. Thus, when the unvalued φ-features of finite Tns probe, on this approach, and find a suitable goal — for example, a DP with a full set of φ-features — the unvalued case feature of that DP gets valued as a kind of "bonus"* (Pesetsky & Torrego, 2007)

For example, take an intermediate representation in the derivation of a transitive structure:

3)



This structure has been built from the bottom-up via Merge (we first Merge (V, Mary), then the output of that with *v*, etc.). We'll focus only on the *T-John* interaction for now.

At this stage, T needs to value its phi-features. It searches its c-command domain and finds the closest NP, [John], with valued phi-features. Two things happen:

- The values for P and N in [John] get copied to T
- The value of NP's [*u*-Case] gets valued Nominative as a by-product of phi-agreement with T

How does search take place?

<u>Minimal Search</u> (see MCB: §2.4.2. I have rambled about MS in Krivochen, 2023)

Essential in the analysis of at least:

- Labelling (Chomsky, 2013: 43; 2015: 6; Bauke & Blümel, 2017: 4; Ke, 2022; Epstein et al., 2015, 2022; van Gelderen, 2022)
- Linearisation (Collins, 2017)
- Agree (Branan & Erlewine, 2021; Ke, 2019, 2022; Preminger, 2019; Milway, 2023)
- Long-distance dependencies / chain formation (Chomsky, 2020a, 2021)

> *Minimal Search* (MS): *Σ searches as far as the first element it reaches and no further. In searching WS, MS selects a member X of WS, but no term of X. In the optimal case of selection of a single SO* [Syntactic Object] *Z, Σ selects the first relevant term Y of Z, and stops.* (Chomsky, 2021: 18)

- MCB / Marcolli (2023) appeal to MS when arguing against 'extensions of Merge' (sidewards Merge -Nunes, 2004-, parallel Merge -Citko, 2005-)
  - It is more economical to search within your own tree, even if you have to go very deep, than to search in a disjoint tree

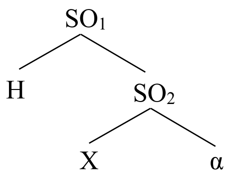
How is 'the first relevant term' identified? Some proposals require counting and set comparison (Kitahara, 2021; Hayashi, 2021):

> [(31)] *{$_\beta$ X {$_\alpha$ Y {Z, W}}} (α=Y, β=X)*
> […] *In (31), MS is required to refer to set β to locate the LI X: let us call it PATH (X) = β. Locating Y, in turn, requires MS to refer to sets β and α: PATH (Y) = (β, α). Here, PATH (X) is a proper subset of PATH (Y), where we can conclude that MS locates X prior to Y and that label β becomes X. In determining label α, X is irrelevant since it is not a member of set α. Since PATH (Y) = α and PATHs of the other competitors (Z and W, which are contained in the merge-mate set of Y) will include sets other than α, Y serves as label α.*

{$_\alpha$ X, Y} (both heads): Path(X) = Path(Y) = α
{$_\gamma$ {$_\beta$ X, …} {$_\alpha$ Y, …}}: Path(X) = (γ, β), Path(Y) = (γ, α)

(Hayashi, 2021: 22)

Same deal in Kitahara (2020). Consider the following tree:



Kitahara (2020: 210) says:

> *MS selects H over X because the path of H (={SO_1}) is a proper subset of the path of X (={SO_1, SO_2}); hence, only H counts as an accessible head for labelling*

MCB assign a weight to accessible terms in a tree depending on their distance to the root, and the weight of embedded terms becomes negligible in the structure of the coproduct

- Marcolli (2023 lecture slides): the coproduct is '*a 'decomposition operation" (one input two outputs) listing all possible ways of decomposing an objects into parts*'

Ke (2019, 2022): set-theoretic Merge, breadth-first search

> *MS = <SA, SD, ST>, where MS = minimal search, SA = search algorithm, $SD_{\in set}$ = search domain (the domain that SA operates on), ST = search target (the features that SA looks for).*
> *Search Algorithm (SA):*
> *a. Given SD and ST, matching against every head member of SD to find ST.*
> *b. If ST is found, return the heads bearing ST and go to Step (c); Otherwise, get the set members of SD and store them as a list L.*
> *i. If L is empty, search fails and go to Step (c); otherwise*
> *ii. assign each of the sets in L as a new SD and go to Step (a) for all these new SDs in parallel.*
> *c. Terminate search.* (Ke, 2019: 44)

- The choice of BF search is due to the fact that it mimics Chomsky's (2013, 2015) argument about the problems of {XP, YP}.

Branan & Erlewine (2021) recognise a problem: search algorithms are formulated for structured data (e.g., a phonebook, an array of some sort). How to go from unordered sets to something that MS applies to is nontrivial.

- For example: Chow's (2022) MS assumes set-theoretic Merge but depends crucially on being able to define left and right daughters of a node, such that in MS, starting from a branching node, the left daughter is checked first, followed by the right daughter. Then, the search goes back to the left daughter, and applies recursively.
    - The *left* daughter. Of an *unordered* set.

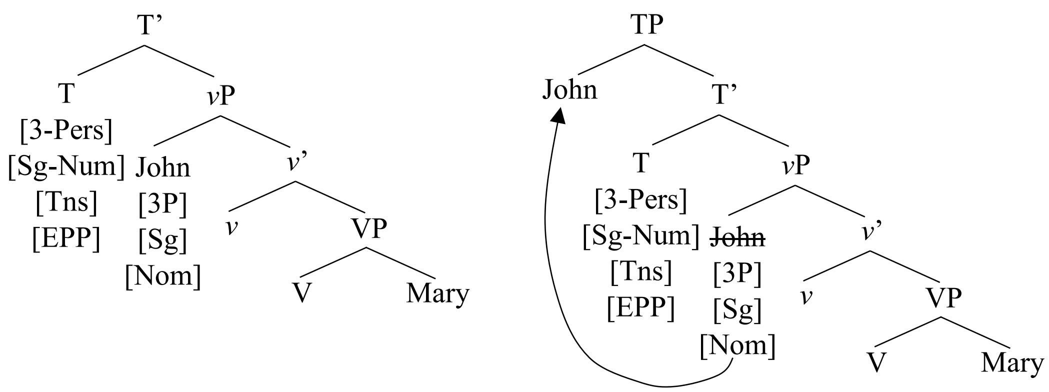
But let's assume that MS works as intended. T probes down, finds [John], Agree happens.

We still have the problem of word order (if computed somehow from the tree). [John] needs to move above T so as to precede it

- Enter EPP: Spec-TP must be filled. This used to be an S-structure filter in GB, now demoted to simple feature.

- [EPP] in a head H means 'please fill my Specifier position'
    - Any head that has a filled Spec must have an EPP feature
    - EPP is uninterpretable: it receives no interpretation at CI or SM. Thus, it needs to be satisfied and deleted before the structure is sent to the interfaces

- Given EPP in T, we Internally Merge [John] to Spec-TP
    - T is already agreeing with it, so why not

4)


- Internal Merge (a.k.a. 'Move') is, at this stage, a composite operation:

Move of β, targeting α, has three components:
A probe P in the label L of α locates the closest matching G in its domain
A feature G' of the label containing G selects a phrase β as a candidate for 'pied piping' [i.e., when something moves alongside its phonological features]
β is merged to a category K (Chomsky, 2000: 135)

- Because we're always looking for the simplest (!) thing, External Merge is always preferred to Internal Merge
    - Merge-over-Move principle: apply Move only as a Last Resort

Move is more complex than its subcomponents Merge and Agree, or even the combination of the two (Chomsky, 2000: 101)

Then,

> *Merge or Agree (or their combination) preempts Move, which is a 'last resort', chosen when nothing else is possible* (Op. Cit.)

For example:

LA_1 = {there, T, be, someone, outside}

LA_2 = {T, be, someone, outside}

Derivational stage: [T [$_{VP}$ be someone outside]]

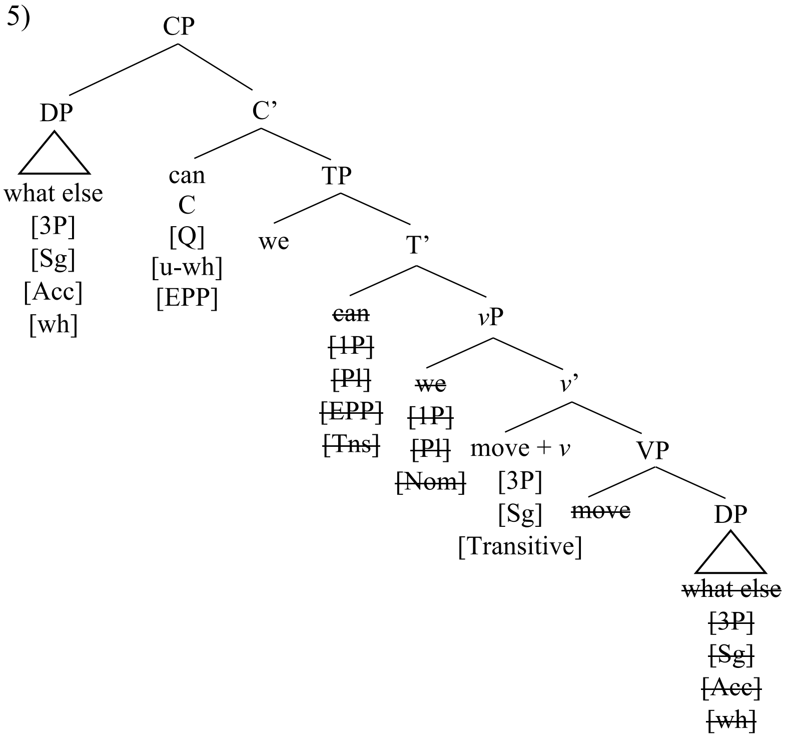
Given LA_1, Expl insertion blocks NP movement ('there is someone outside')

Given LA_2, there's no choice but to move NP ('someone is outside')

- Movement leaves behind a *copy*
    - GB used *traces*, but given Inclusiveness and the fact that we don't have traces in NUM, we need to resort to copies
    - These are problematic:
        - What exactly is a 'copy'?
        - How are copies identified?
        - What condition of 'identity' allows us to link copies (cf. Chomsky's 2021 'certain identical inscriptions', on which more below)?
        - Long-standing problem with 'copies' vs. 'repetitions'
        - Are copies specified in NUM? Do we really have multisets?
        - …
- Copies are (almost always) deleted at PF, but are fully available at LF
    - No more reconstruction to 'undo' movement
    - Copies may be pronounced

Let's look at a *wh*-interrogative (a.k.a. 'constituent interrogative' / 'partial interrogative'): apart from subjects, *what else can we move?*

5)



- The assignment of Accusative case takes place exactly under the same conditions as Nominative, but with *v* as a probe instead of T
    - *Case-assignment here is a reflection of the semantic property Transitivity (TR) of the verb* (Chomsky, 2021: 23; also Chomsky, 2001: 6) (the discussion is about ECM - Chomsky still insists on ECM at times-, but the argument applies to garden-variety monotransitivity)

- The DP [what else] has a lexically valued [wh] feature
    - Roughly, we can identify it with it being an operator / triggering lambda abstraction at LF

- Rizzi (2006): *Criterial positions*

  XP_F and X_F must be in a Spec-head configuration, for F = Q, Top, Foc, R, …

  So, the DP [what else] must be in a Spec-head relation with a C head that also bears a [wh] feature
  C's [u-wh] feature gets valued via Agree, the DP raises to satisfy C's EPP feature
  Note: Agree and Move are in principle distinct. If C did not have an EPP feature, there's no reason for DP to move (at this stage in the theory)

  - EPP satisfaction and phi-agreement can be divorced

        - This is important in the analysis of existential sentences with expletive *there*

Labelling is still a thing: when merging X and Y, we need to know if the output is more X-like or more Y-like

- For some reason, Minimalists hate labels and want to get rid of them (Collins, 2002, 2017; Seely, 2006; Chomsky, 2013…)

- When Merge was triggered by selectional requirements (Chomsky, 1995, 2000), the selector always projected

When α, β merge, it is to satisfy (selectional) requirements of one (the selector) but not both (Chomsky, 2000: 133)

  - Still the way in which things work in Stabler-style Minimalist Grammars
  - Also recent theoretical work such as Zyman (2023), Newman (2022)
  - Argument structure plays a weird role in current Minimalism

<u>A brief excursus: pair-Merge</u>

Remember: in 'old' Minimalism, Merge(X, Y) = {K, {X, Y}} (with K the *label* of {X, Y}, identical to either X or Y)

Chomsky (2000: 133): set-Merge is what used to be called *substitution*.

This is because of the original definition of Merge in (1995), which is a generalised transformation:

<u>Input:</u> *K*

<u>Insert</u> Ø

<u>Result:</u> *K** = {*K*, Ø}

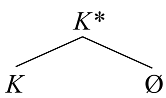
<u>Input:</u> {*K*, Ø}

<u>Substitute</u> Ø with K^1

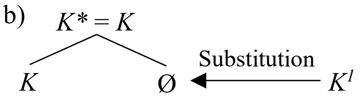
<u>Result</u>: *K** = {*K*, K^1}

<u>Label *K**</u>: *K** = *K*

a) *K* + Ø =



b)



That does not give us (Chomsky-)adjunction: following the labelling procedure in Chomsky (1995), we would get the following, linguistically undesirable, object:

6) Merge(yesterday, {read, {read, {a, {a, book}}}}) = *{yesterday, {yesterday, {read, {read, {a, {a, book}}}}}}

Adjuncts do not change the category or semantic type of their inputs (Dowty, 2003 is a wonderful introduction to the argument/adjunct distinction, from the perspective of Categorial Grammar):

7) a. read a book = VP, type <e, t>
   b. read a book on the beach = VP, type <e, t>

We want (we *need*) a kind of Merge that delivers this.

Chomsky (2000: 133):

> *Adjunction has an inherent asymmetry: X is adjoined to Y. Exploiting that property, let us take the distinction between substitution and adjunction to be the (minimal) distinction between the set {a, b} and the ordered pair <a, b>, a adjoined to b. The constructed objects K, then, are of the form {g, {a, b}} (substitution) or {g, <a, b>} (adjunction), where g is the label of K.*
> […]

[Pair-Merge] *adjoins a to b to form <a, b>. Given the asymmetry, it is natural to conclude that the adjoined element a leaves the category type unchanged: the target b projects. Hence, adjunction of a to b forms K = {g, <a, b>}, where g is the label of b. Eliminating redundancy, the operation forms K = <a, b>.*

I am no set theorist, but '*{g, <a, b>}, where g is the label of b*' should be <b, a>, not <a, b>, if <b, a> ≝ {b, {a, b}} (e.g. Dipert, 1982). Anyways…

- Pair-Merge is still around!

Pair-Merge has been mentioned by Chomsky in connection to:

- experiencers in raising constructions (Chomsky, 2021: 35),
- adjectival modification in NPs (2020a: 49-50),
- 'adjuncts' more generally (2004: 117 -also van Gelderen, 2022: Chapter 6-), and
- at CI it is supposed to deliver '*predicate composition*' (2004: 118).

Pair-Merge would involve syntactic objects derived in parallel (assuming some kind of multidimensional workspace or set thereof, cf. 2020a: 49; 2021: 35), and after its application the adjoined object becomes invisible for both labelling and extraction (2020a: ix, x, 49).

<u>Minimalism Part 2</u>

- 'Simplest Merge' (Epstein et al., 2015, 2022): free, untriggered, unordered set formation

Simplest MERGE is not triggered; featurally-constrained structure-building requires a distinct, more complicated operation (Chomsky et al., 2019: 237)

- Merge maps a workspace WS to WS', increasing the size of WS by *one* (in terms of accessible objects), never decreasing it

What is a workspace?

MERGE operates over syntactic objects placed in a workspace: the MERGE-mates X and Y are either taken from the lexicon or were assembled previously within the same workspace

All syntactic objects in the lexicon and in the workspace WS are accessible to MERGE; there is no need for a SELECT operation […]. *WS represents the stage of the derivation at any given point.* (Chomsky et al., 2019: 236, 245)

MERGE is defined as follows (Chomsky, 2020a: 34, 42; 2021; Komachi et al., 2019: 275):

$$MERGE(P, Q, WS) = [\{P, Q\}, X_1, ..., X_n] = WS', \text{ where if } Y \text{ is a term of } WS, \text{ it is a term of } WS'$$

Does WS include the lexicon? Well…

- Kato et al. (2016: 35): *We assume that WS is the set consisting of SOs already constructed and LIs in the Lexicon, that is,* $WS = \{\Sigma_1, \ldots, \Sigma_n\} \cup Lexicon = \{\Sigma_1, \ldots, \Sigma_n, LI_1, \ldots, LI_m\}$.
- Chomsky (2020a: 45): *for a given I-language, the set of workspaces—the set notice, not the least set—is the set containing the lexicon and containing MERGE (P, Q, WS) for any P, Q and WS that has already been generated*.

- Chomsky (2021: 16): *At each stage of the derivation, we have a set of already generated items that are available for carrying the derivation forward (along with LEX, which is always available). Call this set the Workspace WS. WS determines the current state of the derivation. Derivations are Markovian. The next step doesn't have access to the history, but* […] *WS includes everything previously generated.*

- Small note: 'generated' = 'produced', **not** 'recursively enumerated'

A WorkSpace WS is a finite (multi)set of syntactic objects in SO. The size of the workspace WS is the sum of the number of syntactic objects and the number of accessible terms
[…]
The Merge action on workspaces can be given an axiomatic formulation by imposing a list of desired properties. Some of the fundamental required properties of Merge are:
(1) it is a binary operation (it applies to only two arguments in WS);
(2) any generated syntactic object remains accessible for further applications of Merge;
(3) every accessible term only appears once in the workspace;
(4) the result of Merge applied to two arguments α, β does not add any new syntactic properties to α and β nor it removes any of their existing properties (structure preserving principle);
(5) workspace size does not decrease and increases at most by one. (MCB: 2-3)

Marcolli defines *rooted trees* in WS, which is explicitly something Chomsky argues against

Accessible terms

- tree $T \in \mathfrak{T}_{\mathcal{SO}_0}$ and $v \in V(T)$: subtree T_v rooted at v
- $V_{int}(T)$ non-root vertices of T
- accessible terms of T

$$Acc'(T) = \{T_v \mid v \in V_{int}(T)\} \text{ and } Acc'(T) = \{T_v \mid v \in V(T)\}$$

- workspace $F \in \mathfrak{F}_{\mathcal{SO}_0}$ with $F = \sqcup_{a\in\mathcal{I}} T_a$

$$Acc(F) = \bigsqcup_{a\in\mathcal{I}} Acc'(T_a)$$

Marcolli, lecture slides (2023). More below.

Chomsky (2015):

> *LA* [Labelling Algorithm] *does not yield a new category as has been assumed in PSG and its various descendants, including X' theory. Under LA, there is no structure [α X], where α is the label of X. LA simply determines a property of X for externalization and CI.* ***It is therefore advisable to abandon the familiar tree notations, which are now misleading***

Chomsky (2020a: 38-39):

> ***The tree notations are kind of convenient, but they're very misleading and you should really pay no attention to them.*** *For one thing, a tree notation kind of leads you to suggest that there has to be something at the root of the tree. But that's conflating compositionality with projection. And in fact you often don't have anything at the root of the tree—for example, every exocentric construction. That's what labeling theory takes care of, which eliminates that conflation. Another reason is that when you draw trees, it looks easy to do lots of things that don't make any sense.*

Chomsky et al. (2019: 246):

> [complaining about multidominance *à la* Gärtner, 2002] *complex* [?] *graph-theoretic objects are not defined by simplest MERGE.*

- Binarity is justified in terms of not increasing the size of WS by more than 1 and not decreasing it, but this itself is stipulated
  - Binarity goes way back… Chomsky (1955), Katz & Postal (1964) already apply generalised transformations to two phrase markers
  - Chomsky & Miller (1963) recognised that strict binarity assigns too much structure to iteration (e.g., *the old old old man*), but did nothing about it

- Accessible terms appear only once… but there are copies. And the two-place relation *copy-of* is assigned to '…*certain identical inscriptions*…' (Chomsky, 2021: 17). That's called FORMCOPY.
  - Marcolli (2023) implements a *delete+contract* approach to copies: subtrees dominating copies of IM objects get deleted
  - What happens with *reconstruction*?
  - If embedded subtrees get deleted, how can Agree apply across phase boundaries? (Chomsky et al., 2019: 241)
  - Are 'identical inscriptions' identified *before* deletion?

- These '*identical inscriptions*' existing depends on a principle called STABILITY (Chomsky, 2021: 16):

> Take LEX = {p, q, r, … , ¬ , ∨}. The rules allow us to form the elements of (5), step by step:[24]
>
> (5) p, ¬ p, ((¬ p) ∨ q)
>
> At each stage of the derivation, we have a set of already generated items that are available for carrying the derivation forward (along with LEX, which is always available). Call this set the *Workspace* (WS). WS determines the current state of the derivation. Derivations are Markovian. The next step doesn't have access to the history, but that doesn't matter here since WS includes everything previously generated.
>
> Suppose WS = (5). The inscription *p* occurs three times in (5). The computation is "dumb": it sees three different inscriptions that happen to look alike. There is a convention — call it STABILITY — that the three are all *occurrences* of *p*.

- FORMCOPY must follow STABILITY.
  - See Gärtner (2022) for discussion about the consequences of this for the Copy theory of movement.

- IM is further constrained by another principle: DETERMINACY

> *If Structural Description (SD) for a rule holds for some* [Workspace], *then Structural Change (SC) must be unique*. (Goto & Ishii 2019: 91, as cited in van Gelderen, 2022: Chapter 1. Chomsky does not define it in 2020a, b or 2021, but mentions it somewhat informally)

- This seems to me to be incompatible with the idea that Merge is completely free. Either we have DETERMINACY (and we need to specify SD, SC, and rules, which imho would be great) or the kind of scenario argued for in Chomsky (2020b: 165-166), also Epstein et al. (2015, 2022):

Consider next counter-cyclicity. The problems arise because of the assumption that Internal Merge IM ("Move") is triggered by a probe-goal relation. While conventional, it has always been clear that the assumption cannot be correct, if only because of successive-cyclic movement [...]

The guiding intuition can be preserved if we drop the triggering assumption, and simply assume that Merge (both IM and EM) applies freely, like all rules. ***Free application of rules can yield deviant expressions, but that is unproblematic, in fact required.*** *Deviant expressions should be generated with their interpretations for reasons that go back to Chomsky (1956) and have been amplified in subsequent years. It would radically complicate the generative procedure if, for example, EM were required to yield non-deviant structures; redundantly, because the distinctions are made in any event at CI, and incorrectly, as just noted. There is no more reason to suppose that IM always must yield non-deviant structures.* (highlighting mine)

Collins (2017: 50-53) gives a list of the properties of Merge: summarising,

- Merge is iterable
- Merge is binary (the input of Merge is always a pair of objects)
- Merge is commutative (Merge(X, Y) = Merge(Y, X))
- (The output of) Merge is unspecified for linear order (Chomsky, 2013: 40)
- (The output of) Merge is unlabeled
- Merge is not triggered (by a head, a feature, etc.)
- Merge is never counter-cyclic (*Extension Condition*)
- Merge is all there is structure building-wise: there is no Move or Copy
- Merge *cannot produce* {XP, YP} or {X, Y} objects (where X and Y are heads) (Kayne, 2018: 11)
- Merge allows to dispense with traces, indices, and copies (*Inclusiveness Condition*)
- Merge allows to dispense with the notion of Chain

Simplest Merge is controversial, given the lack of explicitness about filters

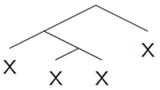
- Hard to see how it actually works when doing grammar. Not used by computational linguists either: massive overgeneration.

This type of free merge would create a huge computational burden because our model would have to compute an enormous number of unsuccessful derivations in order to arrive at a successful derivation. While free merge may have its merits [...] *it is not clear to us how to implement free merge in a computationally efficient manner.* (Ginsburg, 2016, fn. 14).

- Does it look like a *free, commutative, non-associative magma* (MCB: 3, MBC)? Sure[1]. It's the following bit that worries me:

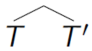
- equivalent description: elements are finite binary rooted trees
(without any choice of planar structure!)

$$\{\{x\{xx\}\}x\} \longleftrightarrow$$



- operation on trees

$$\mathfrak{M}(T, T') =$$



(Marcolli, lecture slides 2023)

Operating on trees creates a new node. Can syntactic operations refer to those nodes? (classically, yes, as in PSGs, but…)

- Labels are determined by a 'labelling algorithm': Minimal Search (seen above) looks for a head and that head provides the label at the interfaces:

  Suppose SO = {H, XP}, H a head and XP not a head. Then LA will select H as the label, and the usual procedures of interpretation at the interfaces can proceed. The interesting case is SO = {XP, YP}, neither a head (we return to the only other possibility, {H, H}). Here minimal search is ambiguous, locating the heads X, Y of XP, YP, respectively. There are, then, two ways in which SO can be labeled: (A) modify SO so that there is only one visible head, or (B) X and Y are identical in a relevant respect, providing the same label, which can be taken as the label of the SO. (Chomsky, 2013: 43, see also Chomsky, 1994: 68)

Summarising:

- Merge(X, {Y, Z}) = {X, {Y, Z}}, X labels

Suppose that we have X = *v* and {Y, Z} a VP. Then,

---

[1] However… Watumull & Roberts (2023) argue, in a 'rebuttal' to Gärtner (2023), that Merge indeed *is* associative. Merge can generate *everything*. A fragment is worth citing, if nothing else to show how much Minimalist grammar and Minimalist philosophy differ:

> *Associativity is a problem only if one takes a myopic picture of Merge: that is, only if one sees it as a constructive operation—analogous to a constructive proof—where the order of operations is all that matters. However, as Watumull and Chomsky (Forthcoming)* [the references contain only a title without any way to access the actual text, so we can only imagine what their argument looks like] *argue, there is the other side of the coin: the classical side, analogous to a classical proof, where all possible applications of Merge apply, such that {X, {Y, Z}} and {{X,Y}, Z}—amongst other structures—are generated. (They are enumerated in the range of the function.) Analogously, the axioms of arithmetic generate—the extension of the intension includes—(A + B) + C and A + (B + C), and every other possible combination. The order of operations only matters when we seek to understand what parts of our knowledge we can use, factoring in third factors, etc. (see Chomsky, 2023)* [literally, 'see', as it is a video lecture].

So: is Merge associative or not? Chomsky appears as co-author on works that argue for what seem to be two mutually contradictory positions.

minimal search finds v as the label of SO since v is unambiguously identifiable (Epstein et al., 2015: 203)

In symmetrical MERGE, if you happen to have a head and an XP, then the head will provide the label – in earlier versions, what projects. But that's a case of MS […]. (Chomsky, 2020a: 48-49)

Labels are identified when an object is sent to the interfaces

- There are weird issues of timing, but no biggie

For example:

8) Merge(read, those books) = {read, {those, books}}, *read* labels

   Merge(v, {read, {those, books}}) = {v, {read, {those, books}}}, *v* labels

- Merge({X, W}, {Y, Z}) = {{X, W}, {Y, Z}}, one of the two sets must be removed. By IM'ing one of the sets, its copy becomes invisible for MS and the remaining thing provides a label:

9) a. {{D, N}, {v, VP}} → MS locates two heads: D of DP and v. No shared features. Cannot label.
   b. {T, {{D, N}, {v, VP}}} → Merge T
   c. {{D, N}, {T, {vP, {<{D, N}>, {v, V}}}}} → IM {D, N} at the root

   Having dissolved the symmetry point, now *v* can label. This is the famous 'labelling account of EPP effects'.

   Under free Merge, IM can move the subject anywhere in principle, but if a labeling failure occurs, then the CI representation pays the price. So labeling indirectly determines the departure site as well as the landing site of the subject. We leave the details of such derivations to further research (see EKS 2013 for relevant discussion). (Epstein et al., 2015: 218)

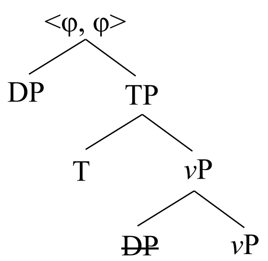
   - If deviant expressions are not a problem, what price is there to pay?
   - We can just derive stuff with no consequence

Not all {XP, YP} situations are resolved via IM, only those where XP and YP do not share a feature:

10) {{D, N}, {T, {vP, {<{D, N}>, {v, V}}}}} (vP is the 'label' created when the {D, N} object moves and its copy is rendered invisible for labelling, ignore it if you will. Strictly speaking, we should also have TP since T labels when it is Merged. See the tree in (11))

In (10), {D, N} and {T, …} share phi-features (inherent in N). These features provide the label for (10) (which is a {DP, TP} situation): <φ, φ>

- Clauses are no longer TPs: they are <φ, φ>
  - The departure from Immediate Constituent analyses is even more evident now
- **Important point**: illustrations of this usually use trees, which is misleading because a root node is created, as we said above (some, such as Ott, 2012: 9 and van Gelderen, 2022: Chapter 1, explicitly say that they will use trees and sets interchangeably):

11) 

MCB (2023: 4): *a set of the form {a} stands for the abstract tree* 

.

- The same labelling stuff motivates successive cyclic movement

If a complex object is IM'd with a non-interrogative C, that C will not have a wh-feature (call it Q), and thus there'll be a labelling issue that will force the complex object to keep raising

12) *They think [[in which Texas city] [C [the man was assassinated?]]]

The intermediate object {{in which Texas city}, {C, {TP…}}} is an {XP, YP} situation with no shared feature.

- If you're wondering about the internal structure of *in which Texas city*, that makes two of us.
- How the Q feature of *which* percolates under set-theoretic assumptions is similarly mysterious. We'll just assume that somehow this whole syntactic object is marked Q
- Also don't ask about pied piping vs. P stranding

The ungrammaticality is due to a labelling failure (Chomsky, 2013; Epstein et al., 2015): that SO is unlabelled.

Intermezzo: Chomsky et al. (2019: 238) say:

> *Featural diacritics typically amount to no more than a statement that "displacement happens"; they are thus dispensable without empirical loss and with theoretical gain, in that Triggered Merge or equivalent complications become unnecessary*

So… no more Q? They rhetorically eliminate '*selectional and discourse-related features; the latter in addition violate IC*' (Op. Cit.).

Anyway, back to our scheduled programming.

Suppose that we have a matrix interrogative C, as in (13) (trying to amend the sketchiness of Epstein et al. and Chomsky. Note that 'in which Texas city' is what we'd call a VP adjunct, so it must have been introduced by Pair Merge):

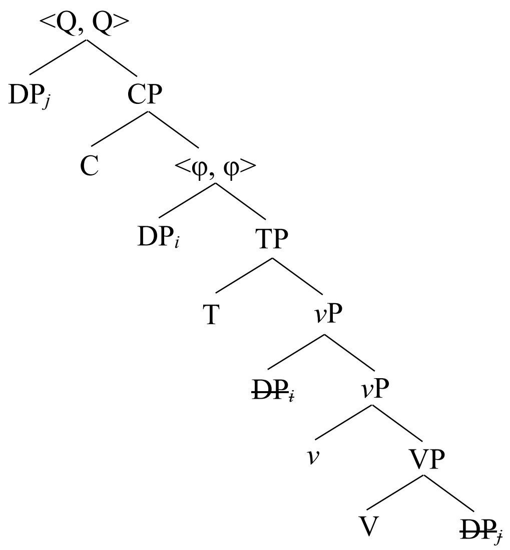
13) {C_Q, {They, {T {<they>, {v, {think {{in which Texas city}$_Q$, {C_{-Q} {{the, man}, {was, {assassinated, <{the, man}>}}}}}}}}}}}

Now, if we IM {in which Texas city} to the root, we'll get

14) {{in which Texas city}$_Q$, {C_Q, {They, {T {<they>, {v, {think {{in which Texas city}$_Q$, {C_{-Q} {{the, man}, {was, {assassinated, <{the, man}>}}}}}}}}}}}

Now, the root is an {XP, YP} situation where there's a shared feature, Q. Wh-interrogatives are thus labelled <Q, Q>:

15)



- Because time and patience are finite (and scarce) resources, I have omitted pointing out phases and adding intermediate landing sites on phase edges when drawing the tree. And also features, of course.
- For current intents and purposes, just accept the claim that C and *v* are sort of 'endmarkers' for probing: they define relatively impenetrable syntactic domains
    - The complement of *v** (*v** is just *v* for all we care, but if you go to the literature you'll find *v** for the endmarker) and of C gets sent to the interfaces for interpretation as soon as it is complete
    - These endmarkers (*phase heads*) are always endowed with uninterpretable phi-features, and thus trigger basically all operations within their syntactic domain
    - If something needs to move outside a phase, it needs to IM to a Spec position of the phase head (the *phase edge*): equivalent to Barriers' 'escape hatches'. See e.g. Citko (2014) for a very accessible introduction to phase theory
    - MCB/MBC say nothing about locality (!!!), but let's see what the derivation of a declarative clause under phase theory looks like:

16) a. {buy, books}
b. {v*, {buy, books}} → Agree(v*, books), Accusative is assigned to *books* as a by-product of phi-agreement with v*
- Head movement?
- Not formulable under set-theoretic Merge. Possibly 'post-syntactic', a PF thing (Chomsky, 2021). Problem: it has (or may have) semantic effects. It obeys successive cyclicity (or it seems to).

c. {John, {v*, {buy, books}}} → transfer of the complement of v*
d. {T, {John, {v*, {buy, books}}}} → labelling problem
e. {John, {T, {<John>, {v*, {buy, books}}}}} → lower copy of *John* is invisible for labelling, v* can now label
- Labelling is counter-cyclic
- Do we have a new element in the set v*P? Depends on what labels are:

   - Seely (2006): labels are proxies for sets
   - Epstein (2000): labels are elements of sets (here we also have a violation of Inclusiveness)
  - Few people have addressed this issue directly

f. Label (e) as <φ, φ>

g. {C, {John, {T, {<John>, {v*, {buy, books}}}}}} → transfer of the complement of C (and presumably C itself. Transfer of matrix clauses is a problem; see e.g. Bošković, 2019)

Now with movement:

17) a. {buy, what$_Q$}

b. {v* {buy, what$_Q$}} → Agree, Accusative valuation, etc.

c. {John, {v*, {buy, what$_Q$}}}

d. {what$_Q$, {John, {v*, {buy, <what$_Q$>}}} → transfer at the v*P phase. *what* is IM'd at the phase edge, as otherwise the derivation would crash

e. {T, {what$_Q$, {John, {v*, {buy, what$_Q$}}}}

f. {John, {T, {what$_Q$, {<John>, {v*, {buy, <what>}}}}

g. {C$_Q$, {John, {T, {what$_Q$, {<John>, {v*, {buy, <what>}}}}}

h. {what$_Q$, {C$_Q$, {John, {T, {<what>, {<John>, {v*, {buy, <what>}}}}}} → transfer at the CP phase

References:

Adger, David. 2003. *Core syntax: a Minimalist approach*. Oxford: OUP.

Bauke, Leah & Andreas Blümel. 2017. Introduction. In Leah Bauke & Andreas Blümel (eds.), *Labels and roots*. Berlin: de Gruyter. 1-16. https://doi.org/10.1515/9781501502118-001

Bošković, Željko. 2019. What is sent to spell-out is phases, not phasal complements. *Linguistica* 56(1). 25-66. https://doi.org/10.4312/linguistica.56.1.25-66

Branan, Kenyon & Michael Yoshitaka Erlewine. 2021. Locality and (minimal) search. To appear in Kleanthes Grohmann & Evelina Leivada (eds.) *Cambridge handbook of Minimalism*. https://ling.auf.net/lingbuzz/005791

Butt, Miriam, Tracy Holloway King, María-Eugenia Niño & Frédérique Segond. 1999. *A Grammar Writer's Cookbook*. Stanford: CSLI Publications.

Chomsky, Noam. 1955. *The Logical Structure of Linguistic Theory*. Mimeographed, MIT. Available online at http://alpha-leonis.lids.mit.edu/wordpress/wp-content/uploads/2014/07/chomsky_LSLT55.pdf

Chomsky, Noam. 1994. Bare phrase structure. *MIT occasional papers in linguistics* 5.

Chomsky, Noam. 1995. *The Minimalist Program*. Cambridge, Mass.: MIT Press.

Chomsky, Noam. 2000. Minimalist Inquiries: the framework. Minimalist Inquiries: The Framework. In Roger Martin, David Michaels & Juan Uriagereka (eds.) *Step by Step – Essays in Minimalist Syntax in Honor of Howard Lasnik*, 89-155. Cambridge, Mass.: MIT Press.

Chomsky, Noam. 2001. Derivation by Phase. In Michael Kenstowicz (ed.) *Ken Hale: A Life in Language*. Cambridge, MA: MIT Press. 1-52.

Chomsky, Noam. 2008. On Phases. In Robert Freidin, Carlos Otero & María Luisa Zubizarreta (eds.) *Foundational issues in linguistic theory*, 133-166. Cambridge, Mass.: MIT Press.

Chomsky, Noam. 2013. Problems of Projection. *Lingua* 130. 33-49.

Chomsky, Noam. 2015. Problems of Projection: Extensions. In Elisa Di Domenico, Cornelia Hamann & Simona Matteini (eds.) *Structures, Strategies and Beyond: Studies in honour of Adriana Belletti*, 1-16. Amsterdam: John Benjamins.

Chomsky, Noam. 2019. Some Puzzling Foundational Issues: The Reading Program. *Catalan Journal of Linguistics*. 263-285.

Chomsky, Noam. 2020a. The UCLA lectures. https://ling.auf.net/lingbuzz/005485

Chomsky, Noam. 2020b. Puzzles about phases. In Ludovico Franco & Paolo Lorusso (eds.) *Linguistic variation: structure and interpretation*. Berlin: de Gruyter. 163-168.

Chomsky, Noam. 2021. Minimalism: where are we now, and where can we hope to go. *Gengo Kenkyu* 160. 1-41. https://doi.org/10.11435/gengo.160.0_1

Chomsky, Noam, Angel Gallego & Dennis Ott. 2019. Generative Grammar and the Faculty of Language: Insights, Questions, and Challenges. *Catalan Journal of Linguistics*. 229-261.

Chomsky, Noam & George Miller. 1963. Introduction to the Formal Analysis of Natural Languages. In R. Luce, R. Bush & E. Galanter (eds.), *Handbook of Mathematical Psychology*. New York: Wiley & Sons. 269–321.

Chow, K. J. 2022. A novel algorithm for minimal search. *Snippets, 42*, 3–5. https://doi.org/10.7358/snip-2021-042-chow

Citko, Barbara. 2005. On the nature of Merge: external Merge, internal Merge, and parallel Merge. *Linguistic inquiry* 36(4). 475-496.

Citko, Barbara. 2014. *Phase theory*. Cambridge: CUP.

Collins, Chris. 2002. Eliminating labels. In Samuel Epstein & Daniel Seely (eds.), *Derivation and explanation in the Minimalist Program*. Oxford: Blackwell. 42–64.

Collins, Chris. 2017. Merge(X, Y) = {X, Y}. In Leah Bauke & Andreas Blühmel (eds.) *Labels and Roots*. Berlin: Walter de Gruyter. 47-68.

Collins, Chris & Edward Stabler. 2016. A Formalization of Minimalist Syntax. *Syntax*, 19(1). 43-78.

Collins, Chris & Erich Groat. 2018. Distinguishing Copies and Repetitions. http://ling.auf.net/lingbuzz/003809

Dipert, Randall. 1982. Set-theoretical representations of ordered pairs and their adequacy for the logic of relations. *Canadian Journal of Philosophy* 12(2). 353–374. https://doi.org/10.1080/00455091.1982.10715803

Dowty, David. 2003. The dual analysis of adjuncts/complements in Categorial Grammar. In E. Lang, C. Maienborn & C. Fabricius-Hansen (eds.), *Modifying adjuncts*. Berlin: de Gruyter. 33–66. https://doi.org/10.1515/9783110894646.33

Epstein, Samuel. 2000. Un-principled syntax and the derivation of syntactic relations. In Samuel Epstein (ed.) *Essays in syntactic theory*. London: Routledge. 141-162.

Epstein, Samuel, Hisatsugu Kitahara & Daniel Seely. 2015. *Explorations in Maximizing Syntactic Minimization*. London: Routledge.

Epstein, Samuel, Hisatsugu Kitahara & Daniel Seely. 2022. *A minimalist theory of Simplest Merge*. London: Routledge.

Fong, Sandiway, Robert Berwick & Jason Ginsburg. 2019. The combinatorics of merge and workspace right-sizing. Evolinguistics Workshop, 2019. https://www.osaka-kyoiku.ac.jp/~jginsbur/WebPresentations/EvolingPresVers11.pdf

van Gelderen, Elly. 2022. *Third factors in language variation and change*. Cambridge University Press.

Ginsburg, Jason. 2016. Modelling of problems of projection: A non-countercyclic approach. *Glossa*, 1(7), 1-46. http://dx.doi.org/10.5334/gjgl.22

Gärtner, Hans-Martin. 2002. *Generalized transformations and beyond. Reflections on Minimalist syntax*. Berlin: Akademie Verlag.

Gärtner, Hans-Martin. 2022. Copies from "standard set theory"? A note on the foundations of Minimalist syntax in reaction to Chomsky, Gallego and Ott (2019). *Journal of Logic, Language and Information* 31. 129–135. https://doi.org/10.1007/s10849-021-09342-x

Gärtner, Hans-Martin. 2023. Merge vs. 'Lerge': Problems of association. *Biolinguistics*, 17, Article e11715. https://doi.org/10.5964/bioling.11715

Goto, Nobu & Toru Ishii. 2019. The Principle of Determinacy and Its Implications for MERGE. In *Proceedings of the 12th GLOW in Asia & 21st SICOGG*, 91-110. http://sicogg.or.kr/GLOW-Asia-12-2019/proceedings/

Hale, Kenneth & Samuel J. Keyser. 2002. *Prolegomena to a theory of argument structure*. Cambridge, Mass.: MIT Press.

Hayashi, Norimasa. 2021. *Labels at the Interfaces: On the Notions and the Consequences of Merge and Contain*. PhD dissertation, Kyushu University.

Hiraiwa, Ken. 2001. Multiple Agree and the Defective Intervention Constraint in Japanese. *Proceedings of the MIT-Harvard Joint Conference* (HUMIT 2000). Cambridge, Mass.: MITWPL. 67–80.
https://www.researchgate.net/publication/228892306_Multiple_Agree_and_the_Defective_Intervention_Constraint_in_Japanese

Hornstein, Norbert, Jairo Nunes & Kleanthes Grohmann. 2005. *Understanding Minimalism*. Cambridge: CUP.

Katz, Jerry & Paul Postal. 1964. *An integrated theory of linguistic descriptions*. Cambridge, Mass.: MIT Press.

Kayne, Richard. 2018. *The place of linear order in the language faculty* [Talk handout].
https://as.nyu.edu/content/dam/nyu-as/linguistics/documents/Kayne%200118%20Venice%20The%20Place%20of%20Linear%20Order%20in%20the%20Language%20Faculty.pdf

Ke, Alan. 2019. *The syntax, semantics, and processing of agreement and binding grammatical illusions*. PhD dissertation, University of Michigan.

Ke, Alan. 2022. Can Agree and Labeling be reduced to Minimal Search? *Linguistic Inquiry*. Advance online publication. https://doi.org/10.1162/ling_a_00481

Kitahara, Hisatsugu. 2020. 'Multiple specifier' configurations revisited. *Reports of the Keio Institute for Cultural and Linguistic Studies* 51. 207-216.

Kitahara, Hisatsugu. 2021. On the notion Copy under MERGE. *Reports of the Keio Institute for Cultural and Linguistic Studies* 52(3). 133-140.

Kitahara, Hisatsugu. 2022. A quick review of some of the recent developments in the Minimalist Program. *Reports of the Keio Institute for Cultural and Linguistic Studies* 53. 151-164.

Komachi, Masayuki, Hisatsugu Kitahara, Asako Uchibori & Kensuke Takita. 2019. Generative procedure revisited. *Reports of the Keio Institute of Cultural and Linguistic Studies* 50. 269-283.

Kratzer, Angelika. 1996. Severing the External Argument from its Verb. In Johan Rooryck & Laurie Zaring (eds.) *Phrase Structure and the Lexicon*. Dordrecht: Springer. https://doi.org/10.1007/978-94-015-8617-7_5

Krivochen, Diego Gabriel. 2023. The search for Minimal Search: A graph-theoretic approach. *Biolinguistics* 17, Article e9793. https://doi.org/10.5964/bioling.9793

Lam, Olivia S-C. 2008. *Object Functions and the Syntax of Double Object Constructions in Lexical Functional Grammar*. D.Phil. thesis, University of Oxford.

Larson, Richard. 1988. On the double object construction. *Linguistic Inquiry* 19(3). 335-391.

Marantz, Alec. 1984. *On the nature of grammatical relations*. Cambridge, Mass.: MIT press.

Mateu Fontanals, Jaume. 2002. *Argument structure: relational construal at the syntax-semantics interface*. PhD dissertation, Universitat Autònoma de Barcelona.

Milway, Daniel. 2023. A formalization of Agree as a derivational operation. *Biolinguistics* 17, Article e9877. https://doi.org/10.5964/bioling.9877

Newman, Elise. 2022. C-selection and the verb phase. https://esnewman.github.io/elisenewman/newman_arg_struc_22.pdf

Nunes, Jairo. 1998. Bare X-bar theory and structures formed by movement. *Linguistic Inquiry* 29(1). 160-168.

Nunes, Jairo. 2004. *Linearization of chains and sidewards movement*. Cambridge, Mass.: MIT Press.

Ott, Dennis. 2012. *Local instability*. Berlin: de Gruyter.

Pesetsky, David & Esther Torrego. 2007. The syntax of valuation and the interpretability of features. In Simin Karimi, Vida Samiian & Wendy K. Wilkins (eds.) *Phrasal and clausal architecture: Syntactic derivation and interpretation*. Amsterdam: John Benjamins. 262-294.

Preminger, Omer. 2019. What the PCC tells us about "abstract" agreement, head movement, and locality. *Glossa* 4(1). 1–42. https://doi.org/10.5334/gjgl.315

Radford, Andrew. 2009. *Analysing English sentences*. Cambridge: CUP.

Rizzi, Luigi. 2006. On the form of chains: Criterial positions and ECP effects. In Lisa LaiShen Cheng & Norbert Corver (eds.) *Wh-movement: Moving on*. Cambridge, MA: MIT Press. 97–134.

Seely, Daniel. 2006. Merge, derivational c-command, and subcategorization in a label-free syntax. In Cedric Boeckx (ed.), *Minimalist essays*. Amsterdam: John Benjamins. 182–217. https://doi.org/10.1075/la.91.13see

Smith, Peter, Johannes Mursell & Katharina Hartmann. 2020. Some remarks on agreement within the Minimalist Programme. In Peter W. Smith, Johannes Mursell & Katharina Hartmann (eds.) *Agree to Agree: Agreement in the Minimalist Programme*. Berlin: Language Science Press. 1-29. DOI:10.5281/zenodo.3541743

Watumull, Jeffrey & Ian Roberts. 2023. Rebuttal to "Merge Is Not 'Lerge'". *Biolinguistics* 17, Article e12393. https://doi.org/10.5964/bioling.12393

Woodford, Ellen. 2006. Lexical Case, Inherent Case, and Argument Structure. *Linguistic Inquiry* 37(1). 111-130.

Zeijlstra, Hedde. 2012. There is only one way to Agree. *The Linguistic Review* 29(3): 491-539.

Zyman, Erik. 2023. On the definition of Merge. *Syntax*. Forthcoming. Available online at https://drive.google.com/open?id=1KWyZrOhrf-exnZpEqyYrm0UYdAqauZ6R&authuser=0