

## On Phrase Structure building and Labeling algorithms: towards a non-uniform theory of syntactic structures

Diego Gabriel Krivochen

University of Reading, CINN

[diegokrivochen@hotmail.com](mailto:diegokrivochen@hotmail.com)

This paper argues that the theory of phrase structure a certain linguistic approach assumes implies taking a stance on the formal nature of the computational procedures that generate that phrase structure. We will proceed by critically evaluating theories of phrase structure and labeling -which implies taking a structure as a unit for the purposes of further computations-, and building on and opposing to the proposals we review, we will claim that syntactic objects are not computationally uniform, and therefore the computational system in charge of establishing dependencies between symbolic objects within the mind is not uniform as well. We argue in favor of a linguistic-cognitive model which dynamically chooses different grammars based on the complexity of the input, and is capable of assigning a mixed phrase marker to an object that presents more than one computational pattern. Empirical evidence is provided in favor of our approach to phrase structure building, and further implications for a theory of labeling and predication are discussed as prolegomena to further research.

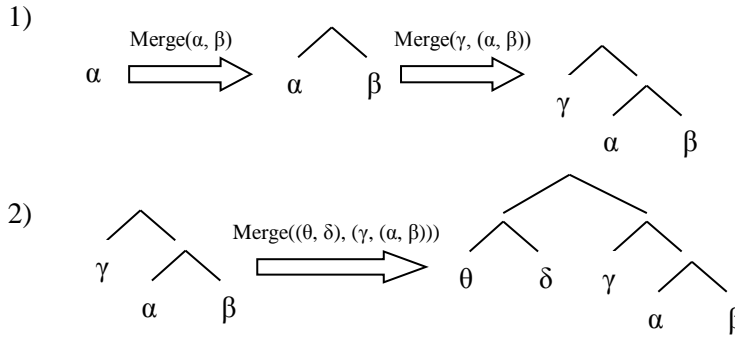
Keywords: labeling, structure building, formal languages, derivations

### **1. What is labeling?**

A generative grammar for natural language is an explicit formal system (Chomsky, 1965: 4; 1995: 162, fn. 1) based on an algorithm that generates structural descriptions of complex structures made out (in orthodox accounts) of atomic elements. This system, in short, takes lexical items from an unordered set (the so-called Numeration) and combines them step-by-step binarily, establishing hierarchical relations between pairs of objects. Current developments within transformational versions of generative grammar, from Chomsky (1995) on, have led to the proposal of a single, ‘free’ generative algorithm called Merge, which ‘*takes objects X, Y already constructed and forms a new object Z.*’ (Chomsky, 2013: 40). Needless to say, stipulations aside, X and Y can be of arbitrary complexity, either terminals (i.e., lexical items, which do not branch) or non-terminals (branching nodes) built via Merge (i.e., sub-trees), which, in Uriagereka’s (2002a) terms corresponds to the distinction between (a) Monotonic (expressible in Markovian terms<sup>1</sup>) and (b) non-Monotonic (non-Markovian) Merge. These options are illustrated in (1) and (2) respectively:

---

<sup>1</sup> Consider that Markov models allow *head-tail* recursion, insofar as loops can consist of arbitrarily long chains of symbols, linearly ordered (i.e., excluding hierarchy). If this is so, as Uriagereka (2008: 233) proposes, then the Markovian character of monotonic Merge (or “merge to the root”) follows straightforwardly: the monotonically assembled sequence [he [saw [the [book]]]] (and any other like it) can be expressed by means of a Markov chain, oversimplified as ‘he → saw → the → book’, with each lexical item



Merge was devised as an attempt to unify two defining properties of natural languages, *hierarchical structure* and *displacement*. The former (External Merge) includes a theory of proper containment, dominance, or non-connectedness relations between syntactic objects; the latter (Internal Merge, or Move), a theory of how syntactic objects are interpreted in syntactic locations that differ from their phonological location. There are some crucial notions in the previous paragraph: one of them is the ‘new object Z’ part. Within the Chomsky Hierarchy CH, an inclusive classification of formal grammars<sup>2</sup> (see Chomsky, 1956: Theorem 1), External Merge EM (where X and Y belong neither to the same object nor to each other, but are taken from the Numeration, in turn a subset of the Lexicon used to derive an expression) is regarded by Uriagereka (2008: 234) as a *context-free* operation, as it creates Z from two any X, Y available in a derivational space, and is thus available for Push-Down Automata, allowing *true recursion*, i.e., free center embedding (as opposed to Finite State Automata, which allow only *head-tail* recursion). Notice, however, that this Merge operation can be further decomposed as follows:

- 3) a. Concatenation (join X and Y together)
- b. Labeling / Projection (form Z)

Labeling implies ‘taking {X, Z} as a unit for the purposes of further computations’, which in turn determines its syntactic behavior (can it be an argument? Can it *take* arguments?, etc.) and interpretation at the sound-meaning interfaces. Such decomposition has been attempted, among others, by Boeckx (2009: 48), Hornstein (2009: 57-58), and Hornstein & Pietroski (2009), under mainstream Minimalist assumptions. Of those steps, only the first is context-free, insofar as Labeling requires the algorithm to peer into either X or Y to find categorial features to project (following orthodox assumptions), such that if  $X = V$  and  $Y = N$ ,  $Z = VP$  via percolation of V’s categorial features, labeling that determines that this object VP, as it is<sup>3</sup>, cannot be an argument, but

---

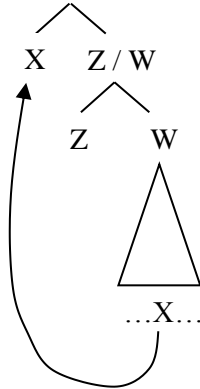
configuring a state of the system. We did not include loops here, but it is equally simple to add  $n$  loops to the Markov chain, as in [he saw the old, boring, heavy book].

<sup>2</sup> When referring to the CH, we will focus on the computational properties of the automata in which the grammars are implemented, and the constructions thereby generated. We will not focus on mathematical linguistics (but see Kracht, 2003 for introduction and discussion), but on the computational properties of the human mind-brain.

<sup>3</sup> Of course, a VP can be an argument *if dominated by T and C*, in which case we are dealing with a whole clause CP (or just by T, if we accept that ECM clauses are bare TPs, see Chomsky, 2000). The point is that a

can take an argument (e.g., an initiator). Thus, Concatenate might be context-free, as it merely puts things together, but Labeling must be as (mildly) context-sensitive as the structure mapping operation Internal Merge IM (formerly Move- $\alpha$ ), if its (mild) context-sensitive nature (in turn, requiring *at least* an extended Push-Down Automata, PDA+ to be computationally implemented, see Joshi, 1985) depends on IM *searching* anywhere within the limits of left and right boundary symbols (Uriagereka, 2008: 227-229) within an already constructed object W to extract / copy a syntactic object of arbitrary complexity  $X \in W$  and merge it to W (so-called *reprojection*, see Matushanski, 2006), or to  $Z \supset W$ , always *extending* the phrase marker in a bottom-up fashion (the so-called *Extension Condition*, see Chomsky, 1995: 189-190; Kitahara, 1997: 5-8) without modifying it (“tampering with it”) internally (see (4)). If EM is indeed free, blind, and unbounded, as many, including Chomsky (2004), Boeckx (2010), and Krivochen (2011, 2012) have claimed, then the (mild) context-sensitive character of a generative grammar is given by the Labeling algorithm: IM is sensitive to labeled structures (‘previously merged structure’, Uriagereka, 2008: 234), implying dominance relations between the syntactic objects affected by the operation, where  $\alpha$  *dominates*  $\beta$  *iff* no segment of  $\beta$  is not a proper subset of  $\alpha$ . In turn, the Extension Condition (as well as the more recent No-Tampering Condition, Chomsky, 2005) is also label-sensitive: consider that the requirement to extend the phrase marker makes sense only if there is a restriction over X extracted from W to merge Z and project W, where W and Z are relevant as *labels*, not as terminals:

4) \*W (if Z projected in the previous step) / \*X / Z



Notice that we are considering that only the two most recent elements in the syntactic workspace are plausible to project (an assumption that goes back to Chomsky, 1995; Kitahara, 1997; Uriagereka, 1998, among others): otherwise, labeling possibilities for the matrix object Z would include all terms of W, including W itself and all syntactic objects it contains, and labeling possibilities for W would also include X, a deep embedded object within W. Constraints to searching for suitable candidates for labeling is not agreed upon, as we will see: phase-based searching algorithms must allow for more possible labels to remain active in the syntactic workspace (as we will see with Gallego’s 2010 *phase-level labeling*), whereas more local accounts (e.g., *every-phrase-is-a-phase* theories, like Epstein & Seely’s 2002) in which evaluation of

---

bare VP has distributional constraints which do not apply if the label had been NP, projecting Y in our example: an NP can always be an argument.

structure takes place over smaller chunks of structure, require labeling to proceed after each structure-building / structure mapping derivational step (i.e., each application of EM or IM respectively). To summarize (see Murphy, 2014a for an extended summary), generative grammar adopts the assumption that the faculty of language is a computational system implemented in the human mind-brain. Moreover, the generative technology, so to say, is mildly *derivational* in orthodox accounts (but see Epstein et. al. 1998; Epstein & Seely, 2002, 2006 for a strongly derivational approach to syntax). This means that, in order to get from a state  $\alpha$  to a state  $\beta$  (where ‘states’ can be represented formally as symbolic ‘strings’  $\Sigma$ , in the sense of Chomsky, 1956) there is a sequence of steps to follow. In transformational grammars TG, like orthodox Chomskyan theory, there are two kinds of such steps:

- Structure building (EM)
- Structure mapping (IM)

Structure building is External Merge, an operation that takes symbolic objects  $\alpha$  and  $\beta$ , from the Numeration,  $\alpha \neq \beta$ , and builds  $\{K, \{\alpha, \beta\}\}$  (see Kitahara, 1997 for discussion). Structure mapping involves tampering with structure building, or, in other words, mapping a structure onto another via a rule. Internal Merge is an example, where either  $\alpha \subset \beta$  (assuming  $\alpha$  is a non-terminal) or  $\beta \subset \alpha$  (assuming that  $\beta$  is a non-terminal). The process can be graphed as follows:

$$5) \quad \Sigma = \{K, \{\alpha, \beta\}\} \rightarrow \Sigma' = \{\alpha_i, \{K, \{\alpha_i, \beta\}\}\}$$

Both the structures built by EM and IM are to be *labeled* in order to be targeted by further operations, which always extend the string.

The structure of the paper is the following: section 2 will be devoted to discussion of previous proposals, including Minimalism (Chomsky, 2008, 2013; and *phase-level labeling*, Gallego, 2010) and non-transformational proposals (Shieber, 1986; Jackendoff, 2011). Since labeled structures are complex structures generated via an algorithm, a discussion of labeling cannot, in our opinion, dispense with a proper discussion of the generative algorithms assumed in different frameworks, for Label is concomitant to Merge (and its alternatives). We will take into account computational and empirical consequences of each proposal, also dealing with the derived implications for the architecture of the cognitive system in which those theories are implemented, the relation between the generative system and interpretative interfaces (with our focus set on the syntax-semantics interface), and the extent to which stipulations can be eliminated in favor of principled claims. In section 3, we will present our own proposal, built upon those previous approaches, taking what we consider the best of each. We will also discuss some empirical problems that our theory can accommodate better than previous proposals, assuming that the computational substratum of language is not uniform but mixed (involving more than one type of structural dependency within CH), which entails a significant departure from orthodox claims in theoretical and computational linguistics. We will examine the consequences of assuming *n*-ary Merge as an option, and the role of the interpretative interfaces in the determination of what counts as a legitimate syntactic object for a computational system we assume to be free to assign structure to an object. The proper

inclusion relation between formal systems will thus be revisited within the framework of generative linguistics.

## 2. Discussion of previous proposals:

### 2.1 Labeling in the Minimalist Program: Chomsky (2008; 2013)

The Merge algorithm brings up the problem of *labeling*, how to signal headedness and account for endocentricity as it seems to be a pervasive feature of human language structures (see Adger, 2013 for a cognitively-oriented introduction; also Murphy, 2014b for a discussion in the context of cognitive evolution and ethology). Chomsky attempted to solve it with a simple rule: he proposed that there are two kinds of *merge*, *pair-merge* and *set-merge* (Chomsky, 1998: 58). In the former, we are talking about adjunction (i.e., elements non-subcategorized for), which is still a problem in minimalist theory, since no satisfactory theory for their derivation has been yet proposed (that we know of; see Uriagereka, 2005 for a Markovian take on adjunction), insofar as current theories (see, e.g., Hornstein, 2009: Chapter 4) rest on a series of stipulations over phrase markers and the distinction between complements and non-complements in a non-principled way. In those cases, if we *externally pair-merge*  $\alpha$  to  $\beta$ , it is always  $\beta$  that projects. Chomsky (2004: 117) has suggested that adjuncts are assembled in a parallel derivational space, and then introduced in the main tree (so-called ‘late merge’) by means of an old mechanism recently revived: a *generalized transformation* GT (see Chomsky, 1957: 113), which, simplifying, introduces a sub-tree in a terminal node of another sub-tree, as we have pointed out with non-monotonic operations under a Multiple Spell-Out model (see also Krivochen, 2012: 63, ff. for a take on nominalizations including *path-of-motion* constructions from a GT approach). Asymmetries between arguments and adjuncts are thus theoretically enhanced in purely structural terms. In *set-merge*, there is some ‘requirement’ of  $\alpha$  which is satisfied by its merger with  $\beta$  (say, argument structure, possibly coded in terms of selectional features), and it is  $\alpha$  that projects a label. The labeling algorithm that Chomsky proposes for Merge( $n$ ), where  $n$  always equals *two* distinct elements put together by virtue of carrying an Edge Feature (which basically conveys ‘I am mergeable’, see Chomsky, 2008<sup>4</sup>), can be summarized as follows (Chomsky, 2008: 145; also 2013: 43)<sup>5</sup>:

- 6) i. In  $\{H, \alpha\}$ ,  $H$  an LI [Lexical Item],  $H$  is the label

<sup>4</sup> A reviewer has pointed out that the Edge Feature is ‘the primary theoretical convention used to argue in favor of labeling’. To the very best of our knowledge, in MGG the EF guarantees (stipulatively) the unboundedness nature of Merge: ‘A property of an LI is called a feature, so an LI has a feature that permits it to be merged. Call this the edge feature (EF) of the LI. If an LI lacks EF, it can only be a full expression in itself; an interjection. [...] The fact that Merge iterates without limit is a property at least of LIs—and optimally, only of LIs, as I will assume. EF articulates the fact that Merge is unbounded, that language is a recursive infinite system of a particular kind.’ (Chomsky, 2008: 139, see also Gallego, 2010: 9). There is no direct reference to labeling in any of Chomsky’s appeals to EF, nor is there in other MGG works, like Gallego’s or Boeckx’s.

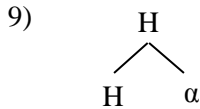
<sup>5</sup> Similarly, Hale & Keyser (2002: 62) claim that “The label of a syntactic object  $X$  is the feature set  $[F, H]$ , where  $[F, H]$  is the entire complement of phonological, morphological, syntactic and semantic features of  $H$ , the head of  $X$ .”. However, Hale & Keyser contemplate the theoretical scenario in which there is simple upwards percolation of features, without the necessity of there being an  $[\alpha, H]$  situation to force labeling.

- ii. If  $\alpha$  is internally merged to  $\beta$ , forming  $\{\alpha, \beta\}$ , then the label of  $\beta$  is the label of  $\{\alpha, \beta\}$ .

In addition to this criterion (ultimately, stipulative) to determine how the result of Merge will be interpreted for the purposes of further computations, there is a corollary to (6 i), which was examined by Uriagereka (1998) and, in less detail, Chomsky (1994): if Label is part of Merge, then, the label of Merge ( $X, Y$ ) is *either*  $X$  or  $Y$ , no other option being acceptable under Minimalist assumptions. Consider the following scenario, summarizing discussion from the aforementioned sources (see Uriagereka, 1998: Appendix for a first formalization of these possibilities within a Bare Phrase Structure model):

- 7) Merge ( $V_{[+V, -N]}, N_{[-V, +N]}$ ) =  $\{V, N\}$
- 8) Label  $\{V, N\}$ :
  - a. VP (i.e., Label  $\{V, N\} = V$ )
  - b. NP (i.e., Label,  $\{V, N\} = N$ )
  - c. Unification ( $[+V, -N], [-V, +N]$ )
  - d. Intersection ( $[+V, -N], [-V, +N]$ )

The generative orthodoxy has adopted option (a), although only on syntactic basis, not always clear. While (as we will see below) Unification can lead to a crashing representation if, as in this case, feature structures do not match (e.g., if one contains a  $[+V]$  feature and the other, a  $[-V]$  feature]), and the Intersection might be an empty set (thus leading to an empty label, rendering the object non-interpretable); there is no clear reason, beyond theory-internal stipulations, to choose (a) over (b) in (8). Two such sets of stipulations concern, on the one hand, *set Merge*, featuring selection from  $X$  to  $Y$  or *vice versa*. In this scenario (corresponding to (6 i)),  $H$  projects because it selects  $\alpha$ , and that asymmetry is captured in labeling. The asymmetry can be graphed in the following way, using familiar X-bar tree representations:



In computational terms, this implies that the structure  $\{H, \alpha\}$  will be taken as  $H$  for the purposes of future operations, including both structure building (Merge) and structure mapping (Move). There are three main problems when considering the MGG approach to Merge + Label:

- Head + head merger<sup>6</sup>

---

<sup>6</sup> As has been correctly pointed out by a reviewer, Panagiotidis (2014: Chapter 5) proposes an alternative for some instances of head-head merger, but his proposal is not part of MGG, as it relies on a theory in which lexical items are not atomic units, but complex structured elements, including roots and a rich array of functional material (see also Panagiotidis, 2010). Given the fact that MGG is mostly lexicalist, we do not think proposals like Panagiotidis' should be included in the discussion of MGG labeling algorithms. In any case, his labeling proposal is parasitic on the operation Agree (2014: 239, ff.), in turn licensed by feature valuation considerations (against which we have argued in past works). In the same vein, Borer's (2013) proposal for categorizer-projection in  $\{\text{Cat-}\sqrt{\phantom{x}}\}$  structures, as well as Distributed Morphology-related approaches, also fall out of the scope of the present discussion.

- Non-terminal + non-terminal merger
- Head adverb + non-terminal merger

Why are those situations problematic? Let us analyze each in turn. Consider a simple structure V + Prn., as in Merge(like, it): unless we assume there are extra projections for the pronoun (see Uriagereka, 1998; Lasnik, Uriagereka & Boeckx, 2005: 117, ff. for an antisymmetry-related proposal that assumes a Top-like projection in order to make a Linear Correspondence Axiom-compatible phrase marker), the relation in a bare phrase structure is of sisterhood between two lexical heads, that is, {H, H}. As far as we know, this is a situation that (6) does not contemplate, insofar as there is no way to determine (unless stipulatively) which of the two elements is H and which is  $\alpha$  (in terms of condition (6 i)), and the reasoning becomes circular (object X is H because it projects, but it projects because it is the selecting H). Chomsky, in discussion with Cedric Boeckx (Boeckx, 2009: 52) makes a strong claim at this respect:

*‘The crucial fact about Merge - the “almost true generalization” about Merge for language is that it is a head plus an XP. That is virtually everything. (...) For one thing it follows from theta-theory. It is a property of semantic roles that they are kind of localized in particular kinds of heads, so that means when you are assigning semantic roles, you are typically putting together a head and something. It is also implicit in the cartographical approach. So when you add functional structures, there is only one way to do it, and that is to take a head and something else, so almost everything is head-XP.’ (our highlighting)*

Two aspects are to be taken into account here: on the one hand, Chomsky is forcing the system to comply with (6 i), adding extra structure (thus, extra empty nodes) when a head-head situation presents (consider, for instance, V+N structures in languages in which D layers are apparently absent, see e.g., Bošković, 2008: even if those theories have been challenged, there is no reason not to take them into account when formulating a generalization this powerful). On the other, the justifications offered for this ‘almost true generalization’ are exclusively intra-theoretical (theta-roles are properties of heads, instead of, for instance, properties of construals; quite the same is the case of cartographical approaches). While it is true that, should one add functional structure, the resulting situation is H-XP, there is no *independent* justification about why one should add functional structure on the first place.

Going further, notice that condition (6 i) rules External Merge (i.e., structure building) *only* when we are dealing with *monotonic Merge*, but *non-monotonic Merge* is not considered (i.e., Externally Merge X and Y, such that X and Y are independent sub-derivations). Adjunction is thus a problem for the projection system, as Hinzen (2009) correctly points out:

*‘...basically they [adjuncts] have never fitted into the apparatus of syntax that minimalism has tried to derive from ‘virtual conceptual necessity.’ They do not receive theta roles, and do not take part in the agreement system.’*

Given this briefly summarized state of affairs, it is hard to figure out where non-subcategorized elements would be merged, since they are apparently dispensable with and thus are not L-marked (Chomsky, 1995) because they are technically not arguments. Adverbs and adverbial constructions

would fall in this category of ‘Adjuncts’, both syntactically and semantically problematic. There are no PSR in the Minimalist Program, and thus the apparent optionality of adverbs and adverbial constructions and their peripheral character as far as the X-bar schema is concerned was not easily justifiable once the notion of L-marking was left aside after initial Minimalist developments (in fact, Chomsky, 1995 is the only place in which we can find the notion within the core orthodox proposals<sup>7</sup>). When we have Merge ( $\alpha$ ,  $\beta$ ), being  $\alpha$  and  $\beta$  heads (say, V + clitic; V + P, etc.), label choice is completely arbitrary: why should Merge( $\text{ire}_v$  +  $\text{ad}_p$ ) be categorized as V and not P (based on Chomsky’s algorithm)? Only L-marking and / or selection stipulations can be summoned to solve the problem. Quite the same happens when we have two non-terminals merging, as in the case of non-monotonic merge of subjects (i.e., DP +  $v'$ ; or merger of DP to T’ on subject raising): unless resorting to semantic reasons (e.g., the fact that, if  $v$  contributes *causativity* to the construal, then it licenses the presence of an initiator argument), there is no *principled* reason to label the resulting object as a  $vP$ . Chomsky himself recognizes this when claiming that ‘*There is no more reason for NP to be SPEC-TP than for TP to be SPEC-NP* [in subject raising]’ (2013: 42). Notice we are *not* saying that the object *is not taken as a  $vP$  or a TP* for the purposes of future computations, but that the purely syntactic criterion presented by MGG is not sufficient to reach that conclusion: feature inheritance stipulations (as the ones assumed to justify some C-T relation, and thus guarantee that T will project instead of N when the subject rises to TP; see Chomsky, 2013: 43, ff.) are intra-theoretical, and do not provide an actual explanation (if anything, a re-statement of the problem in different terms). In Chomsky’s (2013) interpretation, either feature sharing / inheritance stipulations come into play (for C-T relations, possibly holding also for the  $v$ -V pair) or one of the non-monotonic objects has to move (thus multiplying the operations) in order to satisfy antisymmetry requirements such that there are unambiguous c-command paths (Moro, 2000), or else there is no need to label every syntactic object: neither explanation holds independently of the rich intra-theoretical system of current Minimalism. The situation does not become any simpler when adverbs are considered: assume an event-oriented adverb (i.e., adjunct to VP) or a subject-oriented adverb (i.e., adjunct to the causative projection  $vP$ , dominating VP) like ‘fortunately’ or ‘consciously’ respectively. Those adverbs are lexical items, in a bare phrase structure theory; thus, they are terminals. Merge(Adv, VP), if Adv is a lexical item, should result on an AdvP with a VP complement, according to (6 i), but this is prevented by assigning so-called adjuncts a special role within the system of projection (Chomsky, 1995: 329, ff.; see also Hinzen, 2006: 177), a stipulation that is not shared by non-transformational grammars (e.g., Bouma et. al., 2001, section 4). Within the HPSG tradition (see Pollard & Sag, 1994; Green, 2011), for instance, the elements a predicate subcategorized for are represented as a list of features in a lexical entry, called ARG(ument)-ST(ructure), which differs from valence features (SUBJ, COMPL, and SPR –specifier-). Bouma et. al. (2001: 6-7) explicitly say that, whereas the default case is a 1-to-1 correspondence between ARG-ST and valence, it is not always the case. This is particularly relevant for their claim that adjuncts are selected by a lexical head pretty much in the same way complements are. The

<sup>7</sup> Lasnik & Uriagereka (2005: 106) write ‘*In MPLT* [A Minimalist Program for Linguistic Theory, Chomsky, 1993, reprinted as Chapter 2 in Chomsky, 1995] Chomsky suggests that A-chains are somehow L(exically)-related to an argument-taking element (e.g., a verb), while A’-chains are not. We wish we could provide our readers with an explicit definition of L-relatedness, but we know of none.’ While this may overlook proposals like Zagana’s (1988), it is true that the notion has been left undefined as far as Minimalism is concerned.



hypothesis put forth in Bouma et. al. (2001) is that post verbal adjuncts are in the same level as complements (at least, in English). This shows that the labeling algorithm in (6) (which is not actually an algorithm) is stipulative, and does not derive from interface conditions over phrase markers. The picture of labeling in Chomsky's writings (including the very recent 2013 piece devoted to the topic) is far from clear, leaving open the possibility that some objects are not labeled under arbitrary conditions (e.g., successive cyclic *Wh*-movement, where intermediate steps in Spec-CP are unable to meet the *Wh*-criterion, as in Chomsky, 2013: 44). The question of which objects *need* to be labeled (according to MGG) arises, and so does a proposal from *phase theory*, which we will review next.

## 2.2 Phase-based labeling: Gallego (2010)

Stemming from a parametrizable version of *phase theory* (in which *v*-to-T movement, available only in *pro-drop* languages, extends the phase from *v*P upwards to TP via *phase sliding*), Gallego (2010) makes a double proposal with respect to limited domains for syntactic operations:

- 10) *Phase Condition*: Uninterpretable features (*uFF*) signal phase boundaries (2010: 51)
- 11) *Phase-level labeling*: The label of *K* [a term generated via Merge] is determined at the phase level (2010: 19)

The logical consequence of the combination of these principles, which are part of the same formal axiomatic system, is that the label of *K* is determined at the points where a head bearing [*uF*] enters the derivation. Gallego, following Chomsky, assumes those heads are *C* and *v* (based on intra-theoretical assumptions): phases and (thus) labels are defined at CP and *v*P, as both *C* and *v* are equipped from the Lexicon with features that inherit to *T* and *V* respectively.

In our opinion, apart from the highly stipulative nature of the proposal, it turns out problematic in some cases, if labels determine how a syntactic object is to be taken for future computations: consider the case of an ECM construction including an unaccusative *V* (e.g., [go]), like (12):

- 12) John wanted them to go

If we assume the framework of Chomsky (2000; 2008), ECM constructions lack a CP layer, being labeled bare TPs possibly 'defective' in some unspecified sense, insofar as

*'T manifests the basic tense features [Tense, Person, and Number] if and only if it is selected by C (default agreement aside); if not, it is a raising (or ECM) infinitival, lacking  $\phi$ -features and basic tense'* (Chomsky, 2008: 143)

and we also assume, following Chomsky (2001: 107), that '*neither finite TP nor unaccusative / passive verbal phrase is a phase*', the structure of (12) in terms of phasehood is (13):

- 13) [P John [P wanted [N them to [N go]]]] (where P = phase; N = non-phase)

If Gallego's labeling condition (10) is to be applied here, we have to assume that, until matrix *v* is merged, the syntactic object [TP them [[T to] [VP [[V go] ~~them~~]]]] is left both unlabeled *and*

untransferred. If one of the original motivations for phases was the reduction of active objects in the working bench, this is certainly a problematic example, since there is a whole proposition active *go(them)* which is not LF interpretable because it has no label, and cannot be transferred because there is no phase head to trigger the process. We also see that this kind of examples, built using the very same machinery Chomsky has proposed, prove problematic for phase theory, particularly those versions (like Richards' 2007) that aim at a regular P-N alternation: counting all projections in the ECM clause, there is not a single phase (regardless of what the Strong Minimalist Thesis could force). Interface consequences are clear: if the interpretation of N undominated by P is defective because  $\phi$ -features and semantically interpretable dimensions (like tense) are to be inherited, the whole ECM domain is uninterpretable *per se*, and T is regarded as little more than an Agr projection (with no interpretable dimensions of its own, not even Tense).

The label of the syntactic object is thus undetermined until the merger of a phase head, which poses yet another problem: according to Chomsky's (2008) No-Tampering Condition, phrase markers cannot be modified once completed; operations must always *extend* the tree. However, there are two possible scenarios in the system if we assume (10) and (11):

- a) Once matrix  $v$  is merged, *all projections within the domain of  $v$  are labeled*
- b) Once matrix  $v$  is merged, *only the strict complement of  $v$  is labeled*

Both options are problematic. If (a) is the option to take, then all phrase markers within  $\text{Dom}(v)$  (using Uriagereka's 1998: Appendix terminology) are tampered with: if this violation of the NTC is allowed, then only a stipulation could ban others (e.g., External Merge to an already closed projection). On the other hand, if (b) is adopted, then all projections *within* the domain of the strict complement of  $v$  are left label-less. If labels have any reality, they have it at the LF interface (something Gallego himself acknowledges when preferring 'label recognition' to 'label creation'), and render a syntactic object legible for that interface. If a label recognition approach is taken, combined with the assumptions in (9) and (10), the result is that the interface recognizes no label within the domain of the strict complement of  $v$ , which is labeled VP (assuming that  $v$  always select V).

While a free Merge system like the one Gallego apparently assumes is not obliged to label at each derivational step (in the 'label creation' sense), the interfaces can peek into the workspace, see what has been assembled, and interpret it in some way (i.e., label it). If they *can*, only a stipulation could prevent them from doing so, insofar as extremely local evaluation (i.e., peeking after each structure-building operation) can result in transfer smaller fully interpretable units, reducing the amount of material in the working memory without the problems of an aprioristic 'every phrase is a phase' approach (see Boeckx, 2010 for a criticism of this position).

More restricted conceptions of phrase structure building, in which feature relations play a major role (e.g., Pesetsky & Torrego, 2007; Wurmbrand, 2014) can adapt the criteria in (6) to determine the label of a construction based on a Probe-Goal relation: such is the approach taken by Cecchetto & Donati (2010), in which the label of a syntactic object is provided always by the Probe triggering Merge (a way of re-introducing the concept of selection / sub-categorization), but at the cost of

introducing a new stipulation in the formal system. A possibility, then, would be to *eliminate* labels, which we will comment on below.

### 2.3 Eliminating labels in the Minimalist Program?

As an effort in eliminative Minimalism, Collins (2002) attempted to eliminate labels as they were conceived of, and made use of label-free trees doing away with Chomsky's labeling algorithm<sup>8</sup>. In his view, no operation can make reference to 'maximal projections' or 'intermediate projections', since those terms only have sense within traditional X-bar theory. Collins' argument is that even though lexical items may have 'categorial labels' based on the feature matrices [+/- N] and [+/- V], those categorial features *do not project*: his conception of *labels* (which he replaces by the very similar notion of *locus*, a similarity he himself acknowledges; 2002: 48) is not very different in this respect to that of Cecchetto & Donati (2010), who consider that a label is a set of features that percolate from a head, and trigger further computations. Similarly, Collins focuses his criticism on four areas on which labels have been considered necessary<sup>9</sup>:

- X-bar Theory
- Selection (in terms of subcategorization for lexical features)
- Minimal Link Condition (applied to Target- $\alpha$ , for example)
- PF interface (e.g., phonological phrasing only apply to XP)

Cecchetto & Donati (2010) add a subset of Binding Theory, Principle C, which has been dubbed anomalous for decades now (see Lebeaux, 2009 for a recent perspective on Principle C's special status). In their perspective, a tension between two probes and a single goal can explain Principle C effects (a topic we will not discuss here).

According to Collins, all reference to *labels* in the traditional sense can be dispensed with if we take into account four 'syntactic relations' that can hold between lexical items<sup>10</sup>, ill-formations being mainly a matter of violations of a reformulated Minimal Link Condition, which is insensitive to labeling (Cf. Chomsky, 1995):

---

<sup>8</sup> Seely (2006) follows Collins in his eliminative enterprise (sharing many of his assumptions), but the model he describes does not have enough differences with that of Collins for us to review it separately. Seely *assumes* that representations are label-free (in Collins' sense), and asks *why* they should be so (2006: 183) in a strongly derivational syntax.

<sup>9</sup> Interestingly from a methodological viewpoint, Gallego (2010: 20) also lists phenomena whose characterization and proper explanation allegedly *require* labels. We have tackled those issues in Krivochen (2011: 34-37): in both Collins and Gallego's case, half the proof is missing, as there is no evidence that the relevant phenomena *cannot* be explained without labels, each in their respective sense (i.e., Collins replaces *labels* by *loci*).

<sup>10</sup> Strictly speaking, then, Collins' (2014) claim that '*Collins (2002) took the final step and proposed [...] the simplest possible formulation of Merge*' is exaggerated, at best. Merge(X, Y) = {X, Y} has been around since Chomsky (1995), and the proposed 'syntactic relations' further complicate the scenario, rather than offer the simplest possible scenario. It is not clear how those relations are encoded, nor why they are simpler than traditional labels; EPP being the strongest representative of a yet unprincipled stipulation.

- Theta (X, Y): X assigns a theta role to Y
- EPP (X, Y): X satisfies the EPP feature of Y
- Agree (X, Y): X matches Y, and Y values X
- Subcat (X, Y): X subcategorizes for Y

However, Collins' model can be proved wrong if one can demonstrate that those operations either do not exist or are not triggered by features (Krivochen, 2011: 34)<sup>11</sup>. For instance, Krivochen and Kosta (2013) have provided an interface-based explanation for subject-raising (taking into account the notions of theme and grammatical salience) without resorting to an EPP feature: EPP (X, Y) can thus be eliminated. Hale & Keyser's (2002) configurational theta-theory (also developed by Mateu Fontanals, 2002, and which we have adopted in Krivochen, 2012: 78, ff.) takes care of Theta (X, Y), shifting the burden to the LF interface and putting more stress on semantic construal, as well as conceptual primitives (consider, for instance, that Mateu Fontanals' original presentation of argument structure is based on Jackendoff's 1987 *thematic* and *action* tiers, as well as Hale & Keyser's configurational model). Agree and Subcat simply do not arise as operations in a free Merge system: if syntactic objects are built in order to comply with Agree requirements (probing up or down, does not make a difference), then Merge is not free, but stipulatively constrained (insofar as there is no independent evidence for the existence of formal features in the Chomskyan sense, consider the so-called Edge feature from Chomsky, 2008, which basically says 'I am mergeable': this is, we think, a stipulative constraint over an otherwise free operation even for Chomsky, see Chomsky, 2004). Feature-rich proposals like Cecchetto & Donati propose that:

*'Once a label is defined as a subset of the features of one of the two merging objects, the quest for simplification argued for by Collins can be satisfied.'* (2010: 241)

Of course, we have to note that there is no reason *why* a label is to be defined that way (for a related proposal about Merge being triggered by subset relations between feature structures, see Di Sciullo & Isac, 2008), nor is there, in our opinion, compelling evidence that labeling is part of the theory of phrase structure instead of the theory of the syntax-semantics interface.

We are omitting some proposals, insofar as we consider MGG has provided variants of the already reviewed theories instead of alternative frameworks. Provisional conclusions, which will set the starting point for our own proposal, are the following:

14)

- a) The notion of *labeling* is defined only in *syntactic* terms, without reference to the role they play at the interfaces, or whether they might be defined at the interpretative components.

---

<sup>11</sup> At this respect, Seely's model is harder to falsify, insofar as it claims that *labels are syntactically inert* (2006: 184). If they are inert, then the model does not depend on the four syntactic relations mentioned above, and a counterargumentation (within a mild BPS framework) should have to follow a different path. However, the *computational* consequences of Seely's model are just the same as Collins'.

b) Stipulations over labeling are intimately related to stipulations over X-bar theory (e.g., Hornstein’s claim that there is an operation *concatenation* which is linear, and it is labeling that counts as the central innovation in UG –Hornstein, 2009: 55), including *endocentricity* and *binarity*. There is no labeling algorithm for *n*-ary Merge, as in Merge(*X*, *Y*, ..., *n*) in current Minimalist theorizing, in fact, the mere possibility is ruled out by stipulation.

c) The theories we have reviewed (and those we have not, including Chomsky’s 1994 original presentation of Bare Phrase Structure) rest on the assumption that syntactic structure is *homogeneous*: constituency is always hierarchical, always recursive, and always binary (see Kayne, 1984, 1994 for details; for a very recent argument, Collins, 2014); and such a grammar can be implemented in a *uniform* automaton. In other words, these theories impose a homogeneous structure to the mind based on its computability capacities, which are restricted by the formal properties of the generative algorithm. The consequence of the uniformity we have just noted is that a level within the Chomsky Hierarchy *does not* presuppose the lower levels in MGG (we will expand on this below): structures are modeled upon a template (in the case of MGG, X-bar theory) which is inflexible. Therefore, as we will see in section (3) below, mixed dependencies within a syntactic object cannot be accurately modeled<sup>12</sup>.

In the next section we will review the algorithm used by non-transformational models, to the extent that it is based on different assumptions about the relation between feature structures within syntactic structures.

#### 2.4 Unification instead of Merge: Shieber (1986)

The formalism put forth by Shieber, and adopted by HPSG, LFG, and other non-transformational grammars, is heavily based on the concept of *feature structure* (which, although extensively used in Merge-based systems, e.g. Pesetsky & Torrego, 2007; Di Sciullo & Isac, 2008, is not a *necessary* component of the generative algorithm). In fact, it does not apply to undetermined *X* and *Y* objects, but, by definition, applies to *feature structures*, which are defined as a partial function of *features* (*dimensions*, in Minimalist syntax) to *values*: for instance, we can have a mapping from the feature [person] to the value [third] (which is a multivalued function insofar as there are other possible values), as in the following example:

$$15) \left[ \begin{array}{ll} \text{cat:} & NP \\ \text{agreement:} & \left[ \begin{array}{l} \text{number: singular} \\ \text{person: third} \end{array} \right] \end{array} \right]$$

<sup>12</sup> To be fair, some critics of the Minimalist program and its computational assumptions also fall in this category. Consider, for example, Bod (2013), who claims that ‘*language knowledge is hierarchical* [and] *language use is sequential*’. Notice that both objects, use and knowledge, are analyzed uniformly, without there being a possibility licensed by the theory that there are sequential dependencies in language knowledge or that there might be hierarchy in language use (e.g., embedding in adjacency pairs, of the kind  $Q_1 Q_2 A_2 A_1$ ).

Such a structure is abbreviated as  $D_{NP3sg}$ . Feature structures vary in their specificity, and in the number of features they contain (here, we have only *categorical* and *agreement* features, but different formalisms have adopted more, including specifications for grammatical functions and subcategorization frames). Unification applies to feature structures in the following way (Shieber, 1986: 14):

- 16) In formal terms, we define the *unification* of two feature structures  $D'$  and  $D''$  as the most general feature structure  $D$ , such that  $D' \subseteq D$  and  $D'' \subseteq D$ . We notate this  $D = D' \cup D''$ .

Some further clarification is necessary here, as the terminology used in Unification-based grammars does not always coincide with that used customarily in mathematical logics. Within Unification grammars,  $\subseteq$  is used to symbolize a *subsumption* relation between feature structures, in which a feature structure, abbreviated  $D$ , contains *part* of the information of another  $D'$ , such that  $D' \subseteq D$ . In more complex terms, the concept of *subsumption* is based on that of  $dom(D)$  (the *domain* of a feature structure, namely, the features it includes, regardless their mapped values), such that  $D' \subseteq D$  iff  $\forall(x)$  such that  $x \in dom(D')$ ,  $x \in dom(D)$ . A more concrete example will clarify the differences between Unify and (Chomskyan) Merge (taken from Jackendoff, 2011: 276):

- 17) a. Unification of  $[V, +past]$  and  $[V, 3\ sing] = [V, +past, 3\ sing]$   
(not  $[[V, +past] [V, 3\ sing]]$ , as with Merge)
- b. Unification of  $[VP\ V\ NP]$  and  $[V, +past] = [VP\ [V, +past]\ NP]$   
(not  $[[V, +past] [VP\ V\ NP]]$ , as with Merge)

We can recognize three possible results of the operation, based on Shieber (1986: 15):

- Unification *adds* information (e.g., feature structures are not identical, but compatible)
- Unification *does not add* information (e.g., feature structures are identical)
- Unification *fails* due to conflicting information (e.g., same features, different value)

Thus, while Merge could be formulated in such a way that derivations *must* be counter-entropic (that is, the informational load for the interfaces is always increased), this is only *one* of the possibilities with Unification, which makes the derivational dynamics, in our opinion, more interesting (insofar as the computational consequences of each possibility have to be taken into account at each derivational point, if Unification is to be implemented in a derivational model). Jackendoff (2011: 276) claims that Merge can be regarded as a particular instance of Unify, in which feature structures are reduced to two, in the following way:

- 18) Merge (X, Y)  
Unify (X,  $[x\ y]$ ), being  $[x\ y]$  a feature structure containing unspecified terminal elements =  $[X, y]$ , as  $X \supset x$ .  
Unify (Y,  $[X, y]$ ) =  $[X, Y]$ , as  $Y \supset y$ .

However, this reduction of Merge to Unify (which is quite unclear in Jackendoff's piece, more even so when he claims –p. 293– that the steps of relating  $X$ ,  $Y$ , and  $[x\ y]$  to  $[X\ Y]$  are in arbitrary order since the system is not derivational) seems highly stipulative (consider the requirement for certain feature structures and the lack of specificity of their terminal elements, for instance), and based only on labels (in the form of featural specifications, not too distant to the 'labels as percolated features' from MGG). Notice that the important factor here seems to be to maintain the label that would be obtained via Merge (e.g.,  $\text{Unify}(V, N)$  should result in a feature structure with a categorial specification  $V$  or  $VP$ ). It is not clear, also, why one-step Merge should be replaced with two-step Unification, and which computational advantages that might bring.

The standard presentation, however, sets focus on issues that are neglected from an orthodox Minimalist stance, most importantly, the inclusion of the notion of *information* when considering structure building. While we have indeed claimed that Merge *could* be expressed in such a way that structure building operations increase information, it is also true that there is no definition of 'information' within MGG, and structure building operations are triggered purely by syntactic means (e.g., feature checking / valuation). The three possibilities Shieber recognizes are, in our opinion, good candidates for interpretative procedures, if the interfaces can access the workspace in which operations apply.

Consequences for the structure of representations are also interesting, particularly if we consider that the computational system in charge of generating (16 a) might not be context-sensitive to generate (17 b): a Merge-based system, according to Jackendoff's account (with which we partially agree), does not recognize that  $[V]$  in the structures  $[VP\ V\ NP]$  and  $[V, +past]$  can be tokens of the same object, thus assigning a new feature to  $[V]$  (Tense) and projecting a value (+past) to it. A Unification-based grammar is thus a *mildly* context-sensitive grammar, including as subsets both context-free and Markovian grammars, provided it is more flexible than a strict EM-based grammar (which would correspond to a PDA, as it is context-free). The corresponding computational system is, according to Joshi (1985) and Uriagereka (2012: 230-231), an extended Push-Down Automaton (PDA+) which can operate locally within a search space that is not strictly limited to the last-stacked input, but allows limited crossing dependencies (Joshi, 1985: 225). However, it is not clear whether Unification allows mixed dependencies within feature matrices, thus dynamically 'jumping' from one grammar type in the Chomsky Hierarchy to another in order to find the simplest grammar to parse a given phrase marker, a factor we think is crucial for a model of the neuro-cognitive system in which the algorithm is to be implemented.

The scenario we will present goes along these lines: given the possibility that syntactic dependencies are not computationally uniform (which we will argue for both theoretically and empirically), the generative system that creates and interprets those dependencies cannot be uniform either. Thus, while it might be the case that

*'(...) a PDA [Push-Down Automaton] for merely CF [Context-Free] systems is 'too little', while an LBA [Linear-Bound Automaton] for fully CS [Context-Sensitive] systems is 'too much' (Uriagereka, 2012: 231)*

insofar, as Joshi (1985) claims, the grammars that can account for natural languages are slightly more powerful than CFG, but not as powerful as CSG; consequently, the languages they account for are beyond the limits of CFL, but only barely within CSL (most likely due to the limited, but existent, possibility of long distance dependencies, if we consider memory issues). (see also Michaelis, 2001 for a strictly logical analysis of Minimalist grammars); we think there is evidence pointing to the possibility that natural language grammars are not *uniform systems*, a claim so far little explored, if at all. Thus, for instance, Chomsky's (1957) alleged proof that finite-state grammars are inadequate suffers from the assumption (always implicit) that if a grammar is not good for a *subset* of constructions  $S \subset NL$ , it does not apply to *any* part of NL (see Pullum, 2011 for extensive discussion of the mathematical framework of *Syntactic Structures*, and the lack of a 'sound mathematical argument' for the case against Markov models for NL; and Kornai, 1985 for a defense of –uniform- finite state models for natural languages and their neurological substratum).

After this review of past proposals, we will now present the system of structure building and projection we argue in favor of. Before going directly to our proposal, we will review some antecedents on which we have based our theory, to point out similarities and differences, and to highlight the extent to which our analysis has theoretical and empirical advantages over orthodox and these alternative accounts.

### 3. A 'quantum' system of projection:

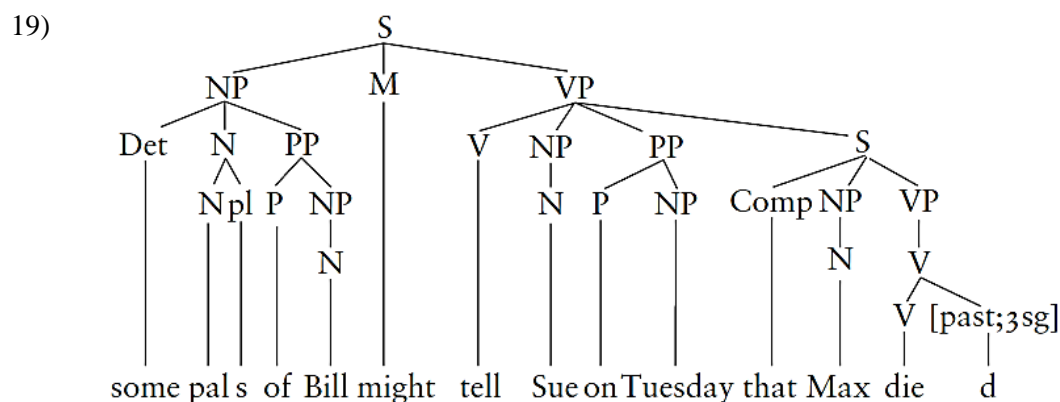
In this section we will expose the main thesis of our paper: if the labeling algorithm a theory proposes gives away (at least part of) the theory of the computational substratum in which that algorithm applies, the shortcomings of the theories reviewed above have impact on the computational possibilities they license for a mind. We will, attending to those shortcomings, and taking the elements we consider correct from those approaches, formulate our own proposal about structure building and labeling, also considering the computational implications of our 'syntactic' claims. What exactly is a 'quantum system' under this premise, then? In the sense in which we have been working the concept in previous works (Krivochen, 2011, 2012), a '*quantum*' system allows syntactic objects of varying complexity to remain in an 'uncertain' state until transferred to the interpretative components, bearing in that state all possible outcomes, all possible values that can be mapped to given dimension: in this system, *category*, *case*, *theta role*, and, possibly, *labels* need not be determined at the derivational point in which the relevant unit is inserted, but they are read off configurations at the semantic component. Moreover, all those dimensions allow more than one possible value, and if this is so, the only non-stipulative scenario (as far as we can see) is to assume that *all values are possible states of the system prior to Transfer(LF)*. We will argue that, contrarily to MGG, dependencies between constituents are not uniform (cf. (14 c)), but we can find constructions for which strictly binary Merge and Labeling would be inadequate, even *within* finite-state compatible (i.e., monotonically assembled) objects. We will focus on *iteration*, and *coordination*, also making reference to *adjunction*, as the relevant empirical phenomena to show mixed dependencies within a string, and propose a more flexible and dynamic approach to deal with them. First, we will review some antecedents for our proposal, which consider the limitations of



Kaynean strictly binary trees (Kayne, 1984, 1994) and explore alternative phrase structure theories analyzing pros and cons. Our proposal will attempt to solve the theoretical and empirical shortcomings we find in these approaches (as well as those reviewed in section 2), improving upon these weaknesses.

### 3.1 Culicover & Jackendoff (2005): Flat structure

The Simpler Syntax hypothesis postulates that syntax is a minimal and optimal mapping between form and meaning. As a result, meaning is to be read off directly from the syntax, and there is no place for hidden levels: the theory is not only non-transformational, but also monostratal. The theory of phrase structure adopted by this model, in which syntactic mechanisms are impoverished (in relation to MGG) is simplified so that, without losing empirical adequacy, there is no extra-structure assigned to a construction. This model explicitly argues in favor of *n*-ary branching, allegedly compatible with recent ‘psycholinguistic research’ (2005: 107). Syntactic representations are dissociated from phonological and semantic structures, which are all *parallel* (Jackendoff, 2002), and include only formal features, including categorial specifications. A sample representation, provided by the authors, is the following (Culicover & Jackendoff, 2005: 110, (2 c)):



The authors characterize the representation in the following terms:

*‘A number of characteristics mark these structures as ‘flat’ in our sense: (a) there is **no hierarchical distinction** in the NPs in [...] (2c) between the attachment of the Determiner and the other complements and adjuncts; (b) the adjunct on Tuesday in (2b) [and (2c)] is likewise a sister of the verb and the argument NP; and (c) the lower clause in (2c) has no special Infl or CP nodes.’* Culicover & Jackendoff (2005: 110. Our highlighting)

Their structures are *n*-ary, ‘flat’ in the relevant sense. Each node dominates as many nodes as elements depend semantically on the head of that node: VP dominates directly V, NP, PP, and S, all constituents traditionally ‘governed’ by V. The cut with traditional X-bar theory is not complete, though: each phrasal node has a single lexical head, all other dominated elements being maximal projections or functional / grammatical material (inflectional morphemes, complementizers, auxiliaries); however, unlike GB X-bar theory, there is no XP/X’ iteration for phrasal adjunction:

there is no syntactic distinction between arguments and adjuncts in the syntactic representation (although there is at the level of the semantic tier, following Jackendoff, 2002), nor is *adjunction* a phrase structure building operation within the framework. With respect to *n*-ary branching, the authors, while acknowledging conceptual advantages for binary branching, assume that, even if 2 is the simplest possibility, it does not mean 2 is actually the case (cf. Collins, 1997, and much related work): phrase structure does not derive exclusively from conceptual simplicity (or, as Chomsky sometimes claims, ‘virtual conceptual necessity’), but also from empirical requirements. We will analyze cases in which the semantics of an expression (accepting there is some sort of mapping, opaque though it might be) require more than binary branching, without precluding the possibility for binary Merge.

Computationally, the consequences of adopting a single template for branching (which we have partially sketched above) are introduced by Culicover & Jackendoff (2005: 113):

*‘One might wish to argue that binary branching is maximally simple on grounds of Structural Uniformity. The argument would be that, first, there are many relations that are binary, and second, we would gain maximum generality by assuming that all relations are binary. However, since everywhere in the universe as well as everywhere in language there will be binary relations, it would follow from this reasoning that only binary relations exist everywhere. It is difficult to see the value of such reasoning, unless it were to turn out that there were positive empirical consequences of generalizing binary relations uniformly, and doing so required no additional complications of the theory. To our knowledge this has not been demonstrated for language, let alone in general.’*

Almost any complex formal object can be modeled in terms of binary relations, from abstract patterns of plant growth to the Fibonacci or Lucas sequences, via an L-grammar (Prusinkiewicz & Lindenmayer, 1991; Uriagereka, 1998); moreover, other kinds of generative devices (e.g., *concatenation* interpreted as *n*-ary addition) can also give us mathematical ‘monsters’ like  $\pi$ , decomposing multiplication and division in series of additions (with positive integers for multiplication and negative integers for division). However, this does not mean that the (computational) *nature* of the object is itself binary, something we can see in linguistic objects if we assume derivations are semantically motivated (in a related vein to the Simpler Syntax hypothesis, but strongly derivational).

Psycholinguistic arguments in favor of binary branching are also discussed. The presented arguments amount to the following:

- 20) a. Learnability: given a structure  $[X_0 \text{ XP YP}]$ , binary labeling makes it easier for the learner to identify heads and complements, according to Haegeman (1992). However, the argument in favor of binary branching is actually a two-part argument: first, branching is uniform in all syntactic objects. Second, that branching template is binary. We have pointed out that this amounts to saying that the mind has only one option for computation, which is unlikely given the variety of stimuli it has to process. Culicover & Jackendoff (2005: 114) discuss samples of visual sets and musical phrases as examples of head-less structures, which directly derive label-less phrases. Taking ‘syntax’ in a wide sense (as *structure*, not simply

as *linguistic structure*), we fully agree with Culicover & Jackendoff in that ‘*Binary Merge is nothing but a degenerate one-dimensional version of n-ary grouping*’ (2005: 114), and there is no principled (beyond LCA-related stipulations, see Kayne, 1994) or empirical reason why syntactic representations are actually 2-D structures, be they binarily-branched or not.

b. C-command, binding, and unambiguous paths: relations of constituent dependency are (apparently) simplified when there is only one possible path from  $\alpha$  to  $\beta$ , a situation which is guaranteed in a binarily-branched tree. Thus, anaphors and pronouns (with particular focus on the former) are sure to be linked to their antecedents at LF, from Reinhart (1983) on<sup>13</sup>. However, while the data proposed by Kayne (1984) in favor of binary branching is consistent with the template, it neither *requires* it nor does it suffice as a *proof* that binary branching is the case. Apart from the NP-structural considerations Culicover & Jackendoff (2005: 116), we add the caveat that if in a double object construction both objects were in the same domain (as in Larson’s account, DO and IO appear within the VP as Spec- and Compl-), binding of reflexives and crossover effects should be possible (because relevant objects are equidistant, from a structural point of view; see Chomsky, 1995), but they are banned:

- 21) a. They showed Mary<sub>i</sub> the picture of herself<sub>i</sub>  
 b. \*They showed a picture of herself<sub>i</sub> to Mary<sub>i</sub>  
 c. \*They gave her mother<sub>i</sub> Mary’s books<sub>i</sub>

Furthermore, the impact of heavy NP-shift on binding is not explained in a theory without D-structure, except by stipulating LF covert movement. Consider, for instance, (22), discussed by Culicover & Jackendoff (2005: 119):

- 22) John showed to Mary<sub>i</sub> herself<sub>i</sub> as a young girl.

Idioms, and other ‘syntactic nuts’ (using the term of Culicover, 1999) are taken as alternative empirical evidence against uniform binary-branching representations by Jackendoff (2011), who assumes the Simpler Syntax hypothesis in order to propose alternative minimalist visions of language. Notice, however, that Culicover & Jackendoff’s work represents the exact opposite of MGG in terms of uniformity: all dependencies are, in the relevant sense, ‘flat’. Of course, as can be seen in (19), there are binary branching non-terminals, but the authors provide no criterion to determine why certain dependencies are binary and certain others are not (see Culicover & Jackendoff, 2005: 145 for a sketch of phrase structure rules with no clear justification for the branching / labeling system). Thus, whereas the projection of P branches binarily, the projection of N does not, yielding a flat structure. Part of our goal is to make the theoretical and empirical motivations this mix of binary and non-binary branching explicit, as well as the criterion to define when the algorithm builds (and interprets) one or the other kind of dependency.

### 3.2 Lasnik & Uriagereka (2011); Lasnik (2011); Uriagereka (2012): Markov models revisited

<sup>13</sup> Notice, however, that binding relations have been captured by Ross (1967) via a combination of linear order and command, without appealing to binary branching.

The inquiry about the nature of the computational system underlying human language was somehow revived, within MGG, by Howard Lasnik and Juan Uriagereka. Lasnik (2011) addresses Markovian properties of ‘lower level syntax’, which had already been spotted by Chomsky & Miller (1963), under the denomination ‘true coordination’:

23) The man comes/The old man comes/The old old man comes

For sentences like (23), a phrase structure grammar of the kind  $\Sigma, F$  ( $\Sigma$  a finite set of initial strings and  $F$  a finite set of rewriting rules) like the one argued for in Chomsky (1957) (to which we will come back below), imposes too much a structure over the adjective stacking, assuming a uniform system of binary projection. Chomsky himself (1963: 298) recognizes the structural overgeneration of phrase structure grammars:

*‘a constituent-structure grammar necessarily imposes too rich an analysis on sentences because of features inherent in the way P-markers are defined for such sentences.’*

Curiously, there was no attempt within MGG to improve the phrase structure engine in order to include Markovian dependencies, but these examples were increasingly overlooked from the ‘80s on: the rise of X-bar structure (Chomsky, 1981; Stowell, 1981), and its *binary branching* requirement (Kayne, 1984) directly excluded Markovian dependencies from the picture. Minimalism’s Merge operation maintained the essentially binary character, ‘deduced’ from apparently independent principles like antisymmetry (Kayne, 1994, 2011; Di Sciullo, 2011) and feature valuation operations (Pesetsky & Torrego, 2007; Wurmbrand, 2014).

Lasnik (2011) acknowledges the problem imposing ‘too much structure’ if a uniform ‘moving up’ in the Chomsky Hierarchy is performed. Even after acknowledging the problem of imposing extra structure on a syntactic object, Chomsky & Miller (1963: 304) insist on the binary character of phrase markers:

*‘The basic recursive devices in the grammar are the generalized transformations that produce a string from a **pair** of underlying strings.’* (Our highlighting)

Lasnik proposes to weaken the binarity requirement so that the structure building algorithm incorporates a generalized transformation mechanism that maps  $n$  phrase markers into a new phrase marker (see also Chomsky & Miller, 1963: 299), which would yield the required flatness for Markovian dependencies. In the same line, Lasnik & Uriagereka (2011) summarize the computational properties of the grammars in the Chomsky Hierarchy while acknowledging that portions of natural languages display Markovian behavior. The upper limits of finite state grammars are found in discontinuous dependencies, better formalized these days. Thus, a finite state grammar, being purely linear (and devoid of memory) could not formalize the relation between both instances of *what* in sentences like (23) (~~strike through~~ represents non-pronounced occurrences):

24) What did you buy ~~what~~

Lasnik & Uriagereka (2011: 42) make a very interesting point with respect to Chomsky's argument against finite-state models:

*'It is an interesting empirical question whether, in I-language terms, a 'more inclusive' description entails abandoning a 'less inclusive' one, when the meaning of 'inclusiveness' is less obvious in terms of a generative procedure.'*<sup>14</sup>

Discussion below will clarify why our answer to this problem is 'no', current PS models (based on binarity requirements imposed by antisymmetry, as in Kayne, 1994 and related work; or considerations about computational complexity, as in Chomsky, 2013: 40, fn. 20, and his definition / characterization of the generative operation Merge) are 'too powerful' to handle what Lasnik calls 'lower-level syntax', that is, local, non-hierarchical, purely linear dependencies based on adjacency (as in iteration and coordination). However, non-local dependencies can easily be modeled by PS grammars, as Lasnik & Uriagereka (2011) claim:

*'The descriptive advantage of Post-style PS grammars, as compared to finite state grammars, is that PS grammars can pair up things that are indefinitely far apart, and separated by dependencies without limit. The way they do that is by introducing symbols that are never physically manifested: the non-terminals.'*

Thus, extra-structure comes automatically, in the form of non-terminals. However, the claim that 'PS grammars can pair up things that are *indefinitely* apart' (our highlighting) seems a bit too strong to us. In purely formal terms, it is correct, but it does not apply to natural language: locality conditions for the establishment of dependencies are too pervasive to be overlooked in the 'design' of a formal model for natural language, and there is a limit for how apart elements can be if we want to link them at the interface levels. Thus, and paraphrasing Rizzi's (2009) Relativized Minimality, X and Z can be paired (i.e., a dependency between X and Z can be established) if and only if there is no intervenient Y, setting a lower boundary. The upper boundary on structural distance (the criterion to determine how 'far apart' X and Z are) is also to be relativized within a MSO model, in which only portions of a derivation are accessible at a time: an X and Z cannot be linked if, for instance, they belong to separate derivational cascades (Kosta & Krivochen, 2014; Krivochen, forthcoming). Turning to the implementational problems of that claim, locality (including dependencies established within command units, phases, or any other locality-inducing domain) was originally based on computational efficiency and maintaining as little material in the active workspace as possible (Chomsky, 2000; before phases were defined exclusively in terms of feature-checking domains, see Gallego, 2010; Chomsky, 2008 among others). The computational adequacy of the quotation above depends on the capacities of the hardware: unlimited structure requires unlimited working memory (thus being on the Turing-machine stage of the hierarchy), but unlimited working memory is not a feature of human minds, which is one of the reasons mild-

---

<sup>14</sup> Chomsky (1959: 143; Theorem 1) defined the relation of inclusiveness between grammars by means of a theorem surprisingly *not* followed by a detailed proof referring to natural languages (as would be expected after such a strong claim), and that relation has not been revisited within orthodox generative grammar in later years.

context sensitivity is generally chosen as the model for the language capacity instead of a Turing machine, with an unlimited memory tape, in which two units X and Z, even though indefinitely apart, are always available in any given derivational point. This has an immediate consequence for phrase structure: only phrase markers complying with (25) are possible

25) The following is a parseable phrase marker



If and only if  $S = n$  non-terminals, where  $n < \infty$

The condition may seem trivial, but it is not if we seriously consider the possibility that the human mind works *uniformly* like a Turing machine, having an unlimited memory tape (the so-called Turing Program for Linguistic Theory, see Watumull, 2014). Lasnik & Uriagereka claim that ‘*there is no dispute within generative grammar with regards to the significance of this sort of structure*’, which is correct, but whether that sort of structure provides an exhaustive characterization of the phrase structure of natural languages is an empirical question; the nature and characteristics of the computational engine that generates and interprets those structures is another. In our opinion, both have been overlooked in recent MGG. Lasnik & Uriagereka (2011) have great merit, in our opinion, insofar as they propose (even though they do not operationalize) that

*‘what we need should be, as it were, ‘dynamic flatness’. But this is the sort of concept that sounds incomprehensible in a classical computational view, while making sense to those for whom syntactic computations are psychologically real’* (2011: 21)

We will attempt to provide a fuller explication of such a dynamic phrase structure model, in which Markovian dependencies and non-Markovian dependencies are equally available, each ‘activated’ according to need, depending on the complexity of the input to be parsed (cf. the example regarding multiple geometries below). We crucially assume, as Lasnik & Uriagereka do, that *syntactic computations are psychologically real*, and, what is more, the properties of model grammars must depend on the computational properties of the neuro-cognitive substratum.

### 3.3 Excluding (pure) Markov models, including Markovian dependencies: binary labeling, *n*-ary branching

Let us experiment with the possibilities a freer and dynamic phrase structure component could give us. We will now spell our provisional assumptions out, in order to make a tentative proposal. Assume the following scenario, which departs from current proposals in a number of ways:

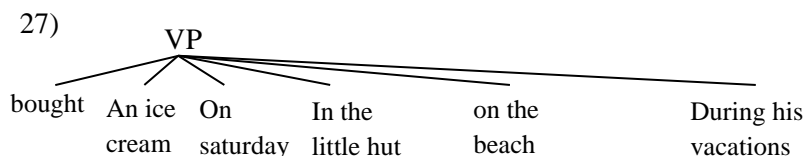
- Labels are not substantive objects *drawn* from a set provided by UG (Cf. Adger, 2013) but *recognized* at the semantic interface (Gallego’s 2010 ‘label recognition’): a label is the way of encoding diacritically how a complex unit is to be interpreted for the purposes of further computations.

- An object of arbitrary complexity can remain ‘unlabeled’ as long as it is not interpreted (what we referred to as a ‘quantum system’). Therefore, it is *Transfer(LF)* that triggers label recognition *at the semantic interface, not before*.
- There is no *a priori* ‘bar-limit’ (Cf. Jackendoff, 1977 with three bars; Bresnan, 1976 with five; Chomsky, 1986 with two), that is, a label can be recognized (and ‘percolated up’) for as long as it is necessary.
- Merge is reduced to *n*-ary *concatenation*, which takes *n* objects and establishes a dependency between them for interface purposes

If our assumptions turn out to be correct, then merge can manipulate *n* objects of arbitrary complexity without labeling them, whereas the semantic system can maintain a recognized label for as long as required. Consider the following example:

26) Juan [compró un helado [el sábado] [en el puestito] [en la playa] [en sus vacaciones]]  
*John [bought an ice-cream [on Saturday] [in the little hut] [on the beach] [during his vacations]]*

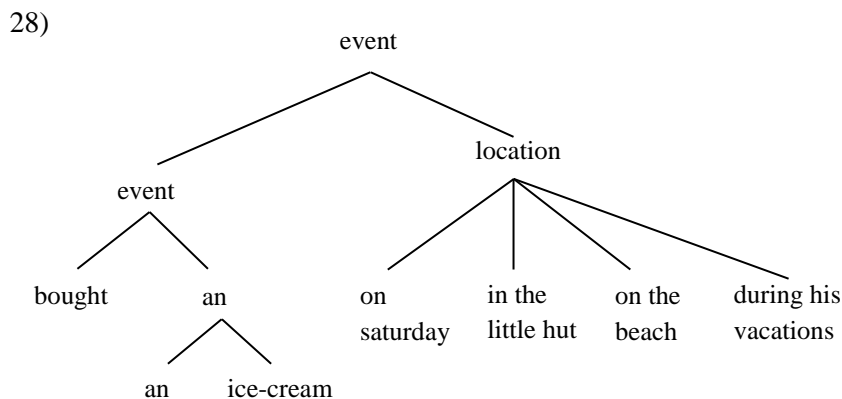
Each square bracket signals a locative domain, therefore, a single labeled node VP dominating five branches (including the complement) should be enough, as in (27):



In (27) we have a term  $K = VP$ , including a head  $V$  determining the label, and a set of sisters, without distinguishing arguments from adjuncts (quite in the line of HPSG and Simpler Syntax). However, this labeling procedure is not context-sensitive enough to separate the head from non-heads (in semantic terms, if the whole constituent is interpreted as an event, the head should be an eventive entity, grammaticalized as  $V$ ), that is, there is no principled way to know it is [bought] that defines the behavior of the whole syntactic object. Moreover, even though the ‘adjuncts’ all convey locative information (either in space or time), [an ice-cream] conveys a completely different kind of information, related to the limitation of the extension of the event [buy] via an affected object, making it telic (if Aktionsart is defined at the VP level, including the eventive terminal and its complements, not just at  $V_0$ ). The differences in interpretation might also give us a clue about the procedure used to assemble each complex unit: locative complements, without there being any scope relation among them, form a Markovian structure, assembled in a workspace  $W1$ , whereas the relation head-affected object, derived via monotonic merge, can be derived in parallel in workspace  $W2$  (recall that monotonically merged structure can be *expressed* within the limits of a finite state grammar, according to Uriagereka, 2012). We have, then, two semantically relevant

units: {V, DP} and {PP, PP, PP, PP}<sup>15</sup>. Non-monotonic Merge can then unify both structures into {{V, DP}, {PP, PP, PP, PP}} to give the complete picture at LF *after* transfer (if monotonic units are Spelled-Out independently, assuming Uriagereka's 2002a Multiple Spell-Out model).

Furthermore, the affected object has a hierarchical structure of its own, with the determiner having scope over the conceptual content, therefore delimiting extension and providing referentiality (see discussion in Krivochen, 2012). A further refinement of (27) could be, thus, (28); where the labels indicate specific LF-relevant information:



The representation is read off at the interface as an *event*: it can be bound by T, affected by Aspectual and Modality auxiliaries, etc.; which delimited in extension by a sortal entity [an ice-cream] (perhaps a generic delimitative label), and including a number of locative specifications, subsumed under a single label 'location'. The question now is why the LF interface should recognize those labels instead of assuming a uniform flat structure, following Culicover & Jackendoff: in our opinion, the answer is given by the computational properties of the parser, which is (at least mildly) context-sensitive.

We see that we have a mixed phrase marker, in which there is a hierarchical relation between [an] and [ice-cream], binarily and monotonically merged, generating a definiteness effect at the semantic interface via semantic scope<sup>16</sup>; and there are also non-scopal relations among the locative elements. We interpret this as a proof that there are at least two kinds of label recognition algorithms triggered by the semantic necessity of interpreting a complex object as a unit for the purpose of further operations, instead of just one which is parasitic on the notion of *head* (cf. (6 i, ii)):

- 29) a. Label(S), S a syntactic object, where all elements belonging to S convey the same kind of information for the semantic component

<sup>15</sup> It is worth pointing out that each PP has been monotonically assembled, such that we actually have, e.g., [on [the [beach]]]. We have jumped right to the derivational point in which all locative objects have been derived and are to be unified with the event.

<sup>16</sup> That is, DP is actually D(N), using a function notation. D is a predicate, like all functional categories (v, T, C), which takes a sortal entity as its argument. It is the procedural content of D that maps the 'value' of N to 'definite' or 'indefinite', using terminology from Unification-based grammars. See Escandell & Leonetti (2000) for an explication of the procedural content of Minimalism's functional categories.



b. Label(S), where elements belonging to S differ in their semantic contribution

Let us see some examples in which we can see (29) working:

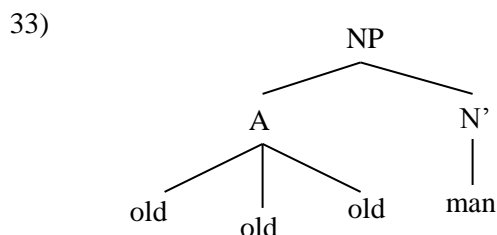
a) Iteration:

30) Sie ist sehr sehr sehr schön

31) The old old old man

32) The man was [old, tired, tall...] but [friendly] (Lasnik, 2011: 358)

In these cases, we cannot say that one instance of the A takes the others as arguments, as a binary branching traditional X-bar theoretical representation would entail: that would impose extra structure on the phrase marker, in the form of silent functional heads (Cinque's 1998 FP, for instance), or embedded APs, [old [old [old]]], which not only complicate the theoretical apparatus (as we need independent evidence for the silent functional material), but also do not account for the semantics of the construction, which is not scopal but incremental (see Uriagereka, 2008: Chapter 6 for discussion). We can circumvent these shortcomings if we allow the system to concatenate those As together (*n*-ary concatenation) and label the whole object A (or AP), insofar as their contribution to the semantic interface representation is the same, following (29 a). Taking these considerations into account, the representation Lasnik (2011) assumes is the following:



The question is: why stop labeling A when iteration finishes (that is, what triggers the situation in (29 b))? That is, unless we come back to criterion (6 i), assuming the bar level proposed by Lasnik is not really there, and we take it to be an instance of an  $\{\alpha, H\}$  merger ( $\alpha = A$ ;  $H = N$ ); it would seem that the problem is always the same. However, introducing *semantic* considerations on labeling procedures could help us solve the problem. Taking (29) into account, as a description about the possible labeling situations we can find, let us make a suitable generalization, which can be taken as an interpretative procedure at the C-I component for a semantically-driven syntax:

34) Label(S), S a syntactic object of arbitrary complexity, depending on the information conveyed by all elements belonging to S:

- a. If all elements belonging to S convey the same type of information, maintain the label for as long as elements that enter the derivation convey that same type of information;
- b. Change the label otherwise

Notice that condition (34 a) tells us nothing about the branching characteristics of S: it can have been either monotonically or non-monotonically assembled. That is, we can be dealing with sequential steps in a monotonic derivation (e.g., {D, N}) or a single step merging  $n$  elements conveying the same kind of information (e.g., locative, as in {PP, PP...}). Let us see what happens when we have  $n$ -ary Merge involving different kinds of elements. Consider the following finite-state derivational scenario:

35) Merge  $(\alpha, \beta, \gamma) = \{\alpha, \beta, \gamma\}$

In the Narrow Syntax, everything would be fine, since Merge is blind and NS is *not an interpretative component*, which means it has no access to the content of the elements it manipulates (Chomsky, 2004; Gallego, 2010; Boeckx, 2010; Krivochen, 2011). But, at the semantic interface, after *Transfer(LF)*, problems would arise. Let us assume that  $\alpha = \sqrt{\phantom{x}}$  and  $\beta$  and  $\gamma$  are procedural / functional categories, say, D and T respectively. To ensure clarity, let us define what we understand by ‘root’ and ‘procedural category’: a root, partly following Borer (2005) is an element endowed only with conceptual content (Borer uses the term ‘listeme’, but that usually includes early root phonology, a characteristic we will not deal with here), which is not usable by the linguistic system in isolation, but only under the scope of a procedural category (Escandell & Leonetti, 2000; Wilson & Sperber, 2004), which provides the semantic system with instructions as to how to manipulate that semantic content, limit its extension, and relate sortal / eventive entities within a syntactic domain. Escandell & Leonetti (2000) argue that the semantics of MGG’s functional categories is procedural in nature, which in turn helps us limit the allowable number of functional nodes: Agr<sub>0</sub> does not provide such procedural instructions, being an intra-theoretical device for feature checking, as such, it is eliminated (an argument that is compatible with Chomsky’s 1995). Consider, then, the following object to be labeled:

36) Label  $\{\sqrt{\phantom{x}}, D, T\} = ??$

Having two procedural categories results in a crash at the semantic interface, as there is no way of labeling a structure where two elements could ‘guide’ the interpretation in different directions, namely, a sortal entity ( $\{D, \sqrt{\phantom{x}}\}$ ) and eventive entity ( $\{T, \sqrt{\phantom{x}}\}$ ) readings. The same would happen if the numbers are changed, say, two roots and one procedural category: even if we think that one of those root may be ‘categorized’ in a local relation with the procedural node (V if the procedural element is T; N if the procedural element is D; see Krivochen, 2012 for discussion), there would still be an uninterpretable element, namely, an uncategorized root that cannot be assigned an interpretation at LF because of its drastic semantic underspecification (see, among others, Panagiotidis, 2013)<sup>17</sup>. Therefore, it is the ‘peeking’ of the semantic system into the syntactic workspace that determines what is concatenated with what (see Krivochen, forthcoming, for a clearer presentation and a full sample derivation). Taking these caveats into consideration, the derivation of (33) would go along the lines of (37):

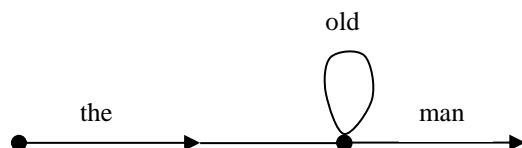
<sup>17</sup> These considerations point to a system in which derivations are *semantically* driven, and no purely syntactic accounts are possible, since linguistic phenomena are essentially always interface phenomena. For a thorough analysis of Wh- movement and parasitic gaps (and an example of a full derivation) within this system, see Krivochen (forthcoming).

- 37)  $\text{Merge}(A, A, A) = \{A, A, A\}$   
 $\text{Label}\{A, A, A\} = \{A\}$   
 $\text{Merge}(N, \{A\}) = \{N, \{A\}\}$   
 $\text{Label}\{N, \{A\}\} = \{N\}$

Notice that all adjectives convey the same kind of information, if we accept localism as a suitable account of the nature of adjectival predication: all adjectives (either individual-level or stage-level) are abstract locations, involving the incorporation of a root on a prepositional node (Mateu Fontanals, 2002: 24, ff.). However, N does not convey locative information, but sortality: once the N terminal is merged, condition (34 b) predicts the change of label.

How can we be sure that the iterative pattern in (30-32) is actually Markovian? For starters, and appealing to an economy argument, it *can* be expressed by means of a finite state grammar, and, if such a simple computational procedure is enough, and captures the semantics of the construction, why attempt to go up in the Chomsky Hierarchy? The structure of (31) can be expressed in finite state terms as follows (cf. Chomsky, 1957: 19):

38)



Chomsky (1957: 21) claims that it is *impossible* for a Markov system to produce *all and only* the grammatical sentences of the English language (a clear constructivist desideratum, see Lasnik, Uriagereka & Boeckx, 2005 for developments of Minimalism under constructivist desiderata). Thus, he arrives at the following conclusion:

- 39) *English is not a finite state language* (Chomsky, 1957: 21. Ex. 9)<sup>18</sup>

He provides examples of discontinuous dependencies like the following as evidence of his claim (1957: 22):

- 40) a. If  $S_1$ , then  $S_2$   
 b. Either  $S_3$  or  $S_4$   
 c. The man who said that  $S_5$ , is arriving today

We have to disagree with (39), even under the light of examples like (40), including discontinuous dependencies (*if-then*; *either-or*; *the man-is*). All the discussion in Chomsky (1957) leading to (39)

<sup>18</sup> Cf. Kornai (1985) for an inverse thesis: “*natural language string sets are regular* [i.e., Finite-State]”. Unlike Chomsky, Kornai draws arguments from language data and acquisition, apart from considerations of theoretical simplicity. While his bases are wider, and more thoroughly argued for than Chomsky’s, the selection of data regarding nested dependencies is limited, and no evidence is provided regarding discontinuous dependencies (as in Wh-interrogatives or parasitic gaps).

proves is that *some portions of the English language* (considered as a set of well-formed formulae, as was the use in the late '50s, particularly within transformational grammar) are not generable by a finite state grammar, when a dependency between  $\alpha$  and  $\beta$  is not established via adjacency, and thus requires some memory (which systems displaying the so-called 'Markov property' do not have). Chomsky's claim, clearly, is a plea for uniformity which is, we think, a stipulation over the limitations of the automaton in which a formal procedure is implemented: if several grammars (parsing procedures) are available (from finite state grammars to mildly context-sensitive grammars, in the case of natural languages), why limit the generative-interpretative capacities to a single step in the hierarchy? The problem, which is primarily methodological, is not exclusive of generative linguistics: an analogy with geometry (elaborated on from Krivochen & Mathiasen, 2012) might help clarifying the scenario. Pylyshyn (2007: 156-158) proposes a series of problems that arise in a geometrical conceptualization of the mental space. These are:

1. *If we represent the fact that A is further from C than from B ( $AC > AB$ ), then there would be a greater quantity of represented space (as distinct from a representation of more space, which makes no commitment about "amount of represented space") between A and C than between A and B. [...]*
2. *If we represent A, B, and C as being ordered and collinear, then there would be an explicit representation of B as being between A and C (where by an "explicit representation" I mean that the relation "between" need not be inferred, but can be "read off" by some non inferential means such as by pattern matching).*
3. *If we represent three objects A, B, and C, then it would always be the case that the distance from A to B plus the distance from B to C would never be less than the distance from A to C. [...]*
4. *If we represent three objects A, B, and C so that AB is orthogonal to BC, then for short distances AB, BC, and CD it would be the case that  $AC^2 + AB^2 = BC^2$  (i.e., distances would be locally Euclidean so that the Pythagorean theorem would hold for short distances).*

These problems, fully valid within a Euclidean framework, dissolve as epiphenomenal if we consider that it is a different geometry that describes mental spaces. Our thesis, **laid out** in Krivochen & Mathiasen (2012), is that *several geometrical models coexist in our minds and their frames are activated whenever necessary*, the simplest model compatible with the data being activated in real time to parse a given stimulus<sup>19</sup>. The performance of multitask computations is a known fact in human brain, controlling both conscious and unconscious bodily functions, and

---

<sup>19</sup> The model proposed here is perfectly compatible with Wegner's (1997, 1998) interactive computation, in which outputs are not direct functions of inputs, as systems are context-sensitive, allow the incorporation of information from various sources during the derivation (perceptual, LTM, inferences, etc.), and display emergent behavior. In MGG, in contrast, outputs (i.e., PF-LF representations) are a direct function of the input (the Numeration), via successive application of Merge. In most incarnations of the MP, even, for a derivation to be well-formed, *each and every step must be well formed*, a strong constraint over generation (Epstein & Seely, 2006). Our model, crucially, is dynamic and interactive, not function-based.

sometimes interchanging these functions. In the light of such a state of affairs, we will **restrict** ourselves to the best known geometrical models, each of which is understood as a model of apprehension of the phenomenological world, and ‘activates’ in a workspace *W* according to a scale of simplicity (from more complex to simpler):

41) Fractal >> Elliptic >> Hyperbolic >> Euclidean

For the time being, we will assume this scale as being based on, among other things, ‘intuitiveness’ (on Aristotelian bases). In Euclidean geometry, the basis is the *point*, defined as that which has no dimension. Subsequent definitions of *line* (1 dimension) and *plane* (2 dimensions) build on the aforementioned definition of *point* which, following the Aristotelian way, is an axiom derived from simplicity and perceptual evidence. What we are saying is that to conceive a Euclidean space requires less use of working capacity because its basic tenets are the default case for our perception. This means that if a phenomenon can be conceptualized using Euclidean geometry, then there is no need to activate hyperbolic working spaces, as Euclidean geometry is simpler in the following crucial sense: its basic assumptions are derived, like all axioms in ancient science, from their simplicity and, mostly, from evidence from a perceptual point of view. Moreover, recent research (Izard et al. 2011) has shown that Euclidean-modelable mathematical intuitions exist among people that have never been exposed to systematic education. If a Euclidean workspace is not enough, i.e., the interface effects we get from activating a hyperbolic *W* outnumber those we would get from a Euclidean *W*, then the hyperbolic frame is activated by an interpretative interface requirement. The human mind can conceptualize what it has never perceived (like hypercubes), but, apparently, there is a limit determined by interface conditions. Our model of dynamic grammar activation, therefore, has more promising properties than that of Pylyshyn, due to the fact that we do not restrict its capacities *a priori*: multidimensional workspaces are a mathematical reality, and if the physical world is in itself a mathematical structure (e.g., Tegmark 2007), then our model has a great potential to arise as a plausible theory of the mind-brain as a physical system, its properties being determined by more general requirements of the structure of reality: for example, an operation applies to an object, and not to an operation. Taking a very basic concatenation operation as a primitive required by virtual conceptual necessity, the rest of the system is developed taking into account the specific properties of the relevant interpretative systems, physically and biologically constrained. We claim that the same operations apply when elaborating hypotheses about the grammar that generates a certain structure within the Chomsky Hierarchy, and the structure that is to be assigned to a certain fragment of language: in our view, Chomsky’s claim (39), with consequences not only for grammar elaboration, but also for the hypotheses about the automaton in which the grammar is implemented, is to be at least *relativized* taking into account the variability of the computational properties of structural descriptions. Notice that, just as it is possible to approximate Euclidean geometry from hyperbolic geometry, as an extreme case of the latter<sup>20</sup> (although complicating the formulae

<sup>20</sup> For instance, consider that the area of a hyperbolic triangle equals  $(\pi/180^\circ) \cdot k^2 \cdot (180^\circ - (A + B + C))$ , *A*, *B*, and *C* being the inner angles and *k* a proportionality factor, as the area of a triangle is proportional to the result of  $180^\circ - (A + B + C)$ . If *A* + *B* + *C* approach  $180^\circ$ , then the formula gives results similar to  $(b \cdot h)/2$ , as in Euclidean geometry, with an error of the order of  $10^{-9}$  (Gómez, 2010). Euclidean geometry is in this case an extreme case of hyperbolic geometry, that in which the sides of the triangle are short enough so that the

unnecessarily), it should be also possible to approximate a level in CH from its immediately containing level (with the caveat that Lasnik & Uriagereka made about the subset relations being an empirical question in current formal syntax), but it implies extra effort at an implementational level (Marr, 1982), as extra structure is required. In our terms, the interfaces' interpretative procedures range dynamically from finite state to mildly-context sensitive operations (i.e., the interfaces, particularly LF, are not computationally uniform, but parse syntactic objects selecting the simplest grammar that can generate the relevant object, while losing the least possible amount of information, particularly in structural terms), the better model being, for us, one including *interactive* procedures in the sense of Wegner (1997, 1998). The human mind, in our proposal, is thus not a uniform PDA+ automaton, but a dynamic, non-function-based computational device that adapts actively to the nature of the input. Of course, Chomsky has a point when he claims that '*there are process of sentence formation that finite state grammars are intrinsically not equipped to handle*' (1957: 23), but this does not lead to the stronger claim that *no portion of English* (or any other natural language) is *finite-state*.

b) Coordination

Coordination structures have been forced into the X-bar template, in an attempt to make coordination an antisymmetric structure with the coordinator as a head (either overt or covert) and the terms of the coordination as Specifier and Complement, respectively. A uniformly X-bar theoretical approach to coordination, like the one defended in Kayne (1994), Progovac (1998, 1999), Zoerner (1995), Chomsky (2013)<sup>21</sup>, among others. These authors argue that coordination structures are *uniformly* headed by a conjunction, and the Spec-Compl dynamics are homogeneous throughout the grammar. However, a uniform approach, to the best of our understanding, cannot explain the following paradigm:

- 42) a. La subida y la bajada de la bolsa preocupan al Gobierno (Spanish)  
*The rise **and** the fall of the stock Exchange worry<sub>[Pl]</sub> the government.*  
 b. La subida y bajada de la bolsa preocupa al Gobierno  
*The rise **and** fall of the stock Exchange worry<sub>[Sg]</sub> the government.*

In the first case, arguably, both DPs [la subida] and [la bajada] motivate plural agreement in the V, being independent entities. Even if there is no hierarchical relation between the constituents (as a c-

---

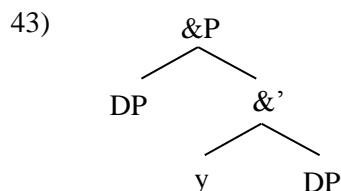
effects of the curvature of the pseudosphere, **on** which hyperbolic geometry is based, are reduced to a minimum.

<sup>21</sup> Chomsky's approach, however, differs in a number of details: while adopting an X-bar template, he also assumes a transformational operation applying to a base form (following Moro's 2000 dynamic antisymmetry), such that (2013: 46):

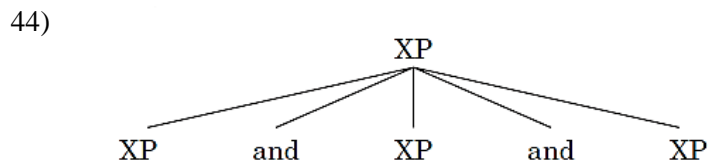
- i) [Conj [X, Z]]  
 ii) [X [Conj [**X**, Z]]]

Notice, however, that (ii) equals (43), even though (43) has been base-generated, and is thus *derivationally* more economical.

command relation would entail), it is possible (though *not necessary*, as we will see) that a &P (or any other headed account of coordination) analysis applies to the example, in the following manner:



However, if *all* coordinate structures were XPs (&P, PredP, etc.), it is hard to see how to account for the agreement paradigm in (42), among other empirical phenomena (basically, lack of scopal relations between coordinated terms in some constructions; while binding effects can appear in others, see Progovac, 1998: 4 for some binding-related effects). A syntactically uniform template is semantically inadequate, which is not acceptable in a semantically-driven derivational system like ours. Our proposal for coordinated structures in instances like (42 b) goes quite along the lines of Jackendoff (1977), basically, the representation in (44):



However, there is no reason for the whole constituent to be labeled identically to the categorial label of one (or all) of its constituents: if the functional potential of the whole constituent is that of a nominal construction, the label will be, say, N; if all coordinated elements (regardless their category) indicate *manner*, that will be the label. (e.g., [*manner* [[PP] and [Adv]]]; as in ‘They ate quickly and in an anxious way’). All in all, the *notation* we use for the label is not **significant** (i.e., *event* instead of VP; *cause* instead of *vP*), as long as there is an explanation of the semantic contribution of each node, and semantically superfluous nodes are eliminated from the construal.

Quite the same happens when we are dealing with clauses instead of DPs. In our terms, the coordinator is a ‘pivot’ between two clauses, each a separate derivational cascade (i.e., independently derived monotonic objects), forming a non-monotonic object. Otherwise, extraction should be possible:

45) \*Who<sub>i</sub> did Mary greet John and t<sub>i</sub>?

Ross (1967: 89) made this situation explicit, via the so-called Coordinate Structure Constraint (CSC):

*In a coordinate structure, no conjunct may be moved, nor may any element contained in a conjunct be moved out of that conjunct.*





An important caveat at this point is that, if the headed representation in (43) were along the right lines for all instances of coordination, then extraction from the second term of the coordination should be possible (since complements are derived in the same derivational cascade as heads, thus configuring a single monotonic object), a prediction that is at odds with the data presented in (46 e), where the whole VP [wash ~~what~~] is derived, in a bottom-up fashion, monotonically in the same derivational cascade in which the head [and] is introduced (following the model in Uriagereka, 2002a). However, a finite-state compatible derivation along the lines of (47), in which all terms of a coordination are independently (and monotonically) generated in separate workspaces, then unified via non-monotonic Merge (the finite-state compatible object having complex stages, each a clause), predicts CSC effects by appealing to the computational properties of the resulting phrase marker without additional stipulations: you cannot extract an element from a phrase marker that is not in the same active workspace as the target of that extraction (see Krivochen, forthcoming; Kosta & Krivochen, 2014 for details about the implementation of this idea). Needless to say, much empirical research is pending, but this seems a promising line of inquiry.

It is to be noticed that, so far, every syntactic object (of arbitrary complexity) we have claimed to enter a labeling process is ultimately binary: this scenario emerges as the simplest non-trivial option<sup>23</sup> we can think of:

48) Binarity requirement for Labeling:

*For any non-terminal syntactic object  $K = \{\alpha, \beta \dots n\}$ , a label  $L$  is determined at the semantic interface for a pair  $(\alpha, \beta) \subset K$  when Analyze applies, after each derivational step.*

Let us clarify the scenario we have in mind. We adopt a version of the 3-dimensional, Caldermobile-like syntactic structure proposed primarily by Uriagereka (1998: 276-277), Lasnik, Uriagereka & Boeckx (2005: 35), and radicalized in Krivochen (2012) to  $n$ -dimensionality and  $n$ -ary branching.  $n$ -dimensionality, that is, the localization of structures in conceptual spaces defined by  $n$  coordinates, is compatible with semantic conditions insofar as conceptual structure can be mapped in  $n$ -dimensional vector spaces (see Graben et. al., 2008 for discussion and some references): a phrase marker, if defined in terms of the location of its terminals in the conceptual space, must extend in more than 2 dimensions (we will come back to this below). However, PF being essentially Markovian (see Idsardi & Raimy, in press; Krivochen, 2013 for a *dynamical frustration* approach to the semantics-phonology tension, in which a globally  $n$ -dimensional semantically driven linguistic derivation is dimensionally flattened and locally chunked into finite-state compatible sub-units for materialization, along the lines of Uriagereka, 2002a), Spell-Out takes a ‘snapshot’ of the  $n$ -ary structure  $S$  at a derivational point  $T$ , more specifically, when there is a set  $F$  of phonological matrices to (sub-)optimally materialize  $\text{Sem} \subset S$  and there is no  $F'$  such that  $F'$  can materialize more of those features (in a word, when there is a distributionally specified phonological matrix corresponding to a set of syntactic nodes, in very much a Distributed

<sup>23</sup> We will not discuss here the proposals of Unary Merge (Adger, 2013) and Merge involving a null terminal (e.g., De Belder & van Craenenbroeck, 2011), insofar as, for interface purposes, both  $\{\{\alpha\}\}$  and  $\{\alpha, \emptyset\}$  are equal to  $\alpha$  at the semantic interface. See Krivochen (2011) for discussion.

Morphology way), performing a forced ‘Markovization’ (or dimensional ‘flattening’) of the symbolic structure<sup>24</sup>. Relations of linear precedence<sup>25</sup> are mapped from that snapshot, without any command requirement, revisiting the ‘snapshot’ proposal made by Lasnik, Uriagereka & Boeckx (2005: 115) loosening the binary-branching requirements, and eliminating the LCA. Of course, since the structure is like a Calder mobile (but with  $n$  branches instead of two, cf. Uriagereka, 1998: 276-277; Lasnik, Uriagereka & Boeckx, 2005: 115, ff.), some of those snapshots will be ruled out by the interfaces, more specifically, via a process of neurological reorganization and dynamic adaptation (e.g., Cherniak, 2009; Amari, 2003): if a snapshot is ruled out  $n$  times (a threshold to be determined empirically), it is just not taken anymore<sup>26</sup>. Labeling, understood as ‘label recognition’ (despite the notation), works in a similar way, applying to pairs of objects when the elements involved do *not* convey the same kind of information (see the discussion above).

An essential feature of the binarity requirement in (48) is that it does not make any claim with respect to the inner complexity of the objects ( $\alpha, \beta, \dots n$ ):  $\alpha$  can be a Markovian object constructed via  $n$ -ary Merge, and  $\beta$  can be a terminal, or a command unit itself. Thus, we avoid the complications of limiting the scope of our labeling operation to H- $\alpha$  or XP-YP relations (cf. 6), but maintain the advantages of binary labeling. This model requires, of course, the additional assumption that the mind is not computationally uniform: a syntactic object can contain *both* linear and discontinuous dependencies, such that a fixed parser might assign either too much or too little a structure to a certain object, scenarios we equally reject. The former situation is common in orthodox accounts of adjective iteration and coordination; consider Cinque’s (1998, 1999) FP for adjectives, and Progovac’s (1998, 1999) &P for coordination (both assigning a rich –functional-structure to Markovian dependencies). The latter (too little structure), admittedly less common, can be found in some pure flat structure accounts, proposed in Culicover & Jackendoff (2005), as well as Hale’s (1983) analysis of Walpiri as a non-configurational language. Culicover & Jackendoff’s analysis of nominal structures (2005: 140), for instance, fails to formalize the semantic relation between D and N, in which the D limits the denotation of N, thus making it referential (see Krivochen, 2012; Escandell & Leonetti, 2000, 2011). Let us confront both structures for a sentence like (49):

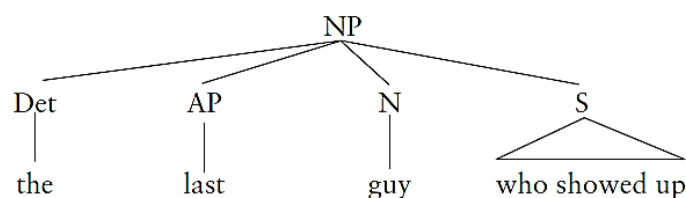
49) The last guy who showed up

<sup>24</sup> The ‘(sub-)optimal’ qualification to Spell Out is compatible with relativizations of the formulation of ‘super-optimality’, see Blutner (2000). However, we are focusing only on the PF-side of the matter here.

<sup>25</sup> Here we take precedence to be the relation by default, although it does not have to be so: Uriagereka (2012) proposes (as a logical alternative) a Mirror LCA, which maps c-command relations to phonological ‘posteriority’ relations. We will not deal with this issue here.

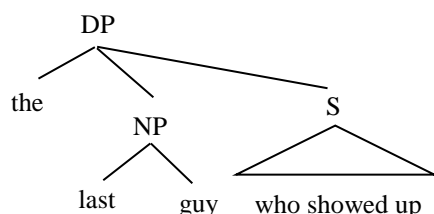
<sup>26</sup> A reviewer has posed the question of whether the process is stochastic in nature. This is in fact a possibility, since there is some indeterminacy: even if we know the initial condition, there are several directions in which the process may evolve. This possibility is particularly interesting, since discrete-time stochastic processes are compatible with (hidden) Markov models, which means that the description of the successive stages of the neural network that is to learn which snapshots converge at PF and which do not might be done via FS grammars. For reasons of scope and space, we will not go deeper into this matter, but leave it for future research.

50) a.



From Culicover & Jackendoff  
(2005: 140), ex. (74 b)

b.



The differences between both structures are clear: (50 b) encodes the asymmetric (semantic) relation between semantic substance (NP) and the constituents that restrict the reference of that semantic substance, a semantic relation that requires (under standard logical assumptions) a scope relation between those elements, such that the constituents limiting the reference of the NP (the D and the relative clause) are also dominating categories hierarchically speaking. This semantically-sensitive representation is also better equipped than purely syntactic accounts (which rely on a rich array of formal features) to account for semantic-pragmatic effects like those formalized in Relevance Theory (Escandell & Leonetti, 2000), as our representations, being devoid of purely formal material, can be taken as the direct input for the process of explicature and implicature building (what is customarily referred to in Relevance Theory as a ‘Logical Form’). In this proposal, functional categories D, T, and C (*v* was not included in their first account) convey instructions as to how to interpret the relation between conceptual content under their scope, thus being *procedural categories* (2000: 368). A model of mixed phrase markers, including phrase-structural *and* Markovian dependencies which is in turn semantically-driven has not only empirical (as we saw for coordination and iteration), but also theoretical advantages, insofar as the computational system in charge of structure building and structure mapping operations is made fully explicit. If our representations constitute legitimate inputs for post-syntactic inferential processes, we will have shed some light on the syntax-semantics/pragmatics interface, an area little explored but crucial, for a model of syntax that is semantically driven (consider, for instance, our labeling recognition proposal (34)). Being focused on a dynamical recognition of the formal properties of the objects to be parsed, and the real-time assignment of the simplest structural description compatible with that object, our model is not, as has been suggested, ‘a mere notational variant’ of flat structure: on the one hand, we propose an explicit semantically-based criterion to build and label, which the Simpler Syntax version of flat structure (which we find quite representative) does not. On the other, we explicitly argue in favor of a *mixed* approach, which implies recognizing the semantic implications of binary concatenation, both monotonic and non-monotonic, and differentiate them from the semantic implications of *n*-ary concatenation (for  $n > 2$ ), claiming that syntactic objects in natural language display *both* kinds of dependencies (see the differences between (50 a) and (50 b), for instance). If a Markov model can accommodate purely

linear, adjacent dependencies, there is no reason to complicate things by assuming extra structure just in order to make the description comply with, say, X'-theory axioms. At the same time, it would be inadequate to claim that all syntactic dependencies are based on adjacency, as has been argued from Chomsky (1957) on: our model can accommodate these seemingly contradictory facts and provide a semantically-based account of empirical phenomena which would require extra assumptions for the theories we have reviewed. While possibly departing from the 'simplest scenario' Chomsky has in mind (see also Culicover & Jackendoff's 2005 claim that *structural uniformity* is the simplest option), we believe this departure is necessary for both theoretical and empirical reasons.

### 3.4 Some further consequences and extensions: towards a dimensional theory of predication

The  $n$ -ary branching phrase structure model we have sketched in previous sections has far-reaching consequences for a theory of predication relations, particularly when we are dealing with phrasal predication (in traditional terms, an XP modifying another XP, or, crucially, more than one). So far, we have worked out the following assumptions:

- 51) a. Merge manipulates  $n$  objects at once, being blind to their inner characteristics
- b. An object of arbitrary complexity can remain 'unlabeled' before *Transfer(LF)* –what we referred to as a 'quantum system'–, including specifications for category, case, and theta-role<sup>27</sup>
- c. Labels are *recognized* at the semantic interface (not *created* by some syntactic algorithm)
- d. Phrase structure dependencies are *not uniform*

In this section we will sketch a theory of predication based on an extension of the formal system entailed by premises (51 a-d). However, it is to be noticed that the extension is not a necessary entailment of those premises, and a consistent system is also attainable if the extension is negated. Let us make this extension explicit:

- 52) a. Symbolic objects are defined in terms of coordinates in a topological mental space
- b. There is no *a priori* limit to the number of coordinates (i.e., axes) an element is to be defined by

How would this work, under the labeling assumptions we have been sketching so far? Assume a Cartesian system of two axes, so that we can define some basic relations taking  $n$  syntactic objects:

- 53) a. Sisterhood: different value for  $x$ , same value for  $y$
- b. Immediate containment: same value for  $x$ , different value for  $y$

Of those relations, only (53 a) is of some interest, as (53 b) is dependent on X-bar theoretical assumptions on projection (i.e., every head projects *both* X' and XP, even if there are no Spec- or Compl- in the phrase). Interestingly, the Markovian dependencies we have discussed can be defined

---

<sup>27</sup> Notice that, crucially, premises (a) and (b) considered together imply that Label is not necessarily a part of Merge in human language, *contra* Hornstein & Pietroski (2009), among others. That is, there can be Merge without Label, and they both target potentially different objects.

taking into account (53 a),  $y$  being constant: if the syntax of adjuncts, coordination, and iteration (that we have dealt with here) is essentially Markovian (see Uriagereka, 2005; Uriagereka & Pietroski, 2002 for more discussion), we need a formal system to account for semantic effects (e.g., lack of scope relations between them). A geometrically-based system is, in our opinion, just as good as any other, empirically speaking (and perhaps theoretically simpler), thus worth pursuing. Notice that we are not appealing, like Uriagereka (2002b) does, to the notions of *paradigm* and *syntagm*, but to more abstract tools to characterize topological mental spaces used for linguistic derivations as well as any other computational process, nor do we base our proposal in the notion of asymmetry, as Uriagereka & Pietroski (2002) do, because we have already characterized some portions of natural languages as finite state in nature, there being no asymmetry involved. This does not mean that our analysis is incompatible with that of the aforementioned authors, but that analysis is a particular case of our proposal: consider, for instance, that Uriagereka & Pietroski take causative constructions and sub-event modification as examples, segmenting the whole event into sub-events to which different modifiers apply. However, their semantic analysis says nothing about how those events and the respective modifiers have been assembled, and what is the resulting syntactic object recognized as at LF, which is the primary concern of the present paper. Their notion of ‘dimension’ is not a tool for the topological description of mental workspaces, but is rather an analogy via which modification over events take place, a perfectly valid take, but different in aims and scope from the one presented here. With respect to the dimensional system proposed by Uriagereka (2002b), it is interesting to consider his claim that

*I make much of the fact that the species which is able to speak is also capable of counting, subtracting, doing fractions, and so on. I think the structure of both systems is not just related, but in fact identical, when seen at the appropriate level of abstraction. This suggests that the stipulation required in order to obtain implicational structure is derivable from deeper properties of cognition. (2002b: 288)*

This is by no means incompatible with our proposal, just like the previous proposal, however, Uriagereka’s dimensional characterization is aimed at syntactic categories, rather than general properties of linguistic structure (to which, no doubt, his ideas could be extended): his development includes the distinction of dimensions such as ‘animacy’, ‘countness’, ‘measurability’ within sortal entities, layers of meaning that we do not distinguish as dimensions, adopting a more topological stance when referring to ‘dimensions’. Uriagereka does apply his proposal to other cognitive mechanisms, adopting a mild (or ‘porous’) modularity approach in which each vertical capacity (in Fodor’s terms) is layered in several dimensions (layers that can communicate), being otherwise encapsulated. We make no such claim about the architecture of the mind but, as we have said before, it is not incompatible with the development we make here. The reader can thus see our proposal as complementary of that of Uriagereka (2002b) and Uriagereka & Pietroski (2002), particularly insofar as our empirical material also differs from theirs, in any case enhancing the empirical coverage of the ‘warps’ proposal.

The central point of this section is to define predication relations and projection (i.e., labeling) within this system we have sketched here. Departing from a Euclidean framework, we assume that

the first element in a derivation is defined as a point, with a single coordinate. The second element is defined in relational terms with the first, and thus we need a further coordinate. Then, as the derivation unfolds, a Calder mobile-like element is assembled within a mental workspace, increasing the dimensional complexity of the overall object as predicates are introduced and related to their arguments. Notice that, even if there were reasons to claim that the first element to enter a derivation is defined by more or fewer coordinates than we have argued for here, the derivational dynamics would not change. This leads us to the central claim of this section, which is (provisionally) (54):

- 54) If a predicate is to have scope over a referential (either sortal or eventive) variable, the number of the predicate's coordinates in the mental working area properly contain the number of the arguments' coordinates.

The reader might find it useful to compare (54) with the following proposal, taken from Uriagereka (2002b: 297):

*If operation  $\sim O$  is not closed in system  $X$ , applying  $\sim O$  to the objects  $x$  of  $X$  creates new sorts of objects  $x'$ , so that a new system  $X'$  is created with the  $x'$  objects, such that  $X$  is a part of  $X'$ .*

It is not clear to us what exactly the meaning of 'system' is in this statement, even though the previous discussion leads to an interpretation of systems as Cartesian axes. In any case, we share the condition that predication relations require the predicate to properly contain the argument in terms of dimensions, but that does not necessarily mean the creation of a new system  $X'$ , nor does such creation need to occur at every application of  $\sim O$ . The creation of  $X'$ , in our proposal, is limited to non-monotonic Merge, where independent derivational cascades are involved. Furthermore, not every application of Merge yields predication relations: we claim that there is no scopal relation of the kind created by  $f(x)$  functions (which capture the basic formal properties of predication) between iterated elements, nor is there any predication relation between members of a conjunction [ $X$  and  $Y$ ]; however, in the latter case there *is* a new system created as non-monotonic Merge is involved. Therefore, the subset relation and the creation of a new system  $X'$  are not only divorced from each other, but also subject to different conditions, in the theory we have sketched here. There are crucial aspects of our proposal, thus, that cannot be expressed in Uriagereka's terms, making the two theories compatible in some respects, but easily differentiable.

For instance, let us see what a predication relation over a sortal entity would look like:

- 55) a)  $N = (x, y)$   
       b)  $A = (x, y, z)$   
       b)  $A(N) = A \times B = (x, x') (y, y') (\emptyset, z)$

And so on. For instance, if a category  $X$  ( $X \neq A$ ) should take  $(A(N))$  as an argument, it would have to be defined in  $(x'', y'', z'', w)$ , introducing a further dimension represented by the  $w$  axis. As the reader may have noticed, the predication relation involving coordinates is expressed as the Cartesian product of the coordinate sets involved, as we have proposed in earlier works. If this is

correct (much empirical research pending, particularly focused on the consequences this proposal has at the semantic interface, including phenomena like binding and constraints over the establishment of dependencies, whether by structure building or structure mapping –i.e., movement-), many essential properties of the generative operation will follow, properties independently derived from set theory and widely proven in other domains. This allows us to define very precisely the scope of multiple modifiers, as in iteration (e.g., if  $n$  As modify an N, but neither has scope over one another, then they are all defined by the same  $n$ -plet of coordinates in the cognitive workspace). At this point in the development of the proposal, it is not really crucial whether N is a pair, a triplet, or a single coordinate, what is crucial is that if  $N = (n)$ , the predicate that takes N as its argument will be defined as  $(n+1)$ -dimensional. In other words, *for  $n$  dimensions*<sup>28</sup> of  $\alpha$ ,  $f(\alpha) = n+1$  (Cf. Uriagereka, 2002b: 296, who proposes that dimensions can ‘warp’ in order for us to get higher-level dimensions; similar arguments are used in multidimensional string theory, with dimensions stretching and folding in specific topological configurations). It is not clear whether the relevant dimension has ‘warped’, but, rather, that  $n + 1$  properly contains the relevant object described within  $n$ , such that a predication relation can be established. Interestingly, this proposal goes along with the distinction between monotonic and non-monotonic merge, insofar as monotonic Merge builds structure while maintaining the number of relevant dimensions; non-monotonic Merge, ‘bringing’ objects from different workspaces, introduces further dimensions in the construal (without the formal *necessity* of ‘bending / warping’ space onto itself, but not precluding it either).

There is a potentially problematic point to address here: let us assume the NP [old man], with [man] being an argument of [old], as the adjective is a one-place predicate. Most linguists would agree that we are selecting a subset of men, those who are old, rather than a subset of old things, those which are men. However, following standard logical notation, we adopt the *predicate(argument)* form. The problem is, if we are actually picking an element that belongs to the set of [men] rather than to the set of [old], should we not consider the coordinate set of [men] in our workspace properly contains that of [old]? While we think that caveat might be true from the point of view of comprehension (progressive referent specification), it is not obvious that it applies to the dimensional formalism to account for relations in the cognitive workspace. Sortal entities are better seen as *figures* against a *ground*, which is a location, either literal (e.g., [the house]) or metaphorical (a property): a localist view of cognition tends to favor the proposal we have sketched thus far, assuming the *ground* properly contains the *figure*, the latter being more salient due to different factors (e.g., inner complexity). A consequence of our proposal, to be further researched, is that the notions of labeling and predication do not coincide: the predicate in a simple, binary {X, Y} structure is not necessarily the label of {X, Y}. If labels do not derive syntactically (as in (6), or via percolation of categorial features), then ‘exocentric’ labeling is possible<sup>29</sup>, where a construction is ‘exocentrically labeled’ if {X, Y} is labeled as neither X nor Y, but, for instance, as Z (where  $Z \neq Y$

<sup>28</sup> We assume here that each axis corresponds to a dimension.

<sup>29</sup> The co-existence of endocentric and exocentric constructions in human language has been noticed and discussed by several linguists before within the structuralist tradition, see Hockett (1958: 184, ff.); also Lyons, within early generativism, (1968, section 6.4.3) for the first ‘problem of projection’ (which arose within the Standard Theory) we know of.

$\neq X$ ), where  $Z$  is an arbitrary name for the semantic contribution  $\{X, Y\}$  make at the semantic interface (see (58) below for an example with a conjunction).

The proposal we have just introduced has far reaching consequences for the analysis of problematic data involving discontinuous dependencies that, to the best of our knowledge, cannot be modeled by a standard PS grammar. In (46 c), for instance, there is no need to assume subject ellipsis if we consider that, since [Bill] is the subject of two VPs, the latter are defined in  $n$ -axes (say, 2, since we have an argument and its selecting V) and the former, in  $n+1$  axes, thus modifying both VPs without the need to include a further copy / token of the subject DP in the representation plus a deletion rule. Notice that the proposal could work in typologically different languages, and different constructions as well, consider (56):

- 56) Iturnam misero succurrere fratri / suasi et pro vita maiora audere probavi (Verg. *Aen.* XII, 813-814)  
*Iturna*<sub>ACC</sub> *unfortunate*<sub>DAT</sub> *rescue*<sub>INF</sub> *brother*<sub>DAT</sub> / *persuaded*<sub>ISgPerf</sub> *and for life better venture*<sub>INF</sub> *judged*<sub>ISgPerf</sub>  
 ‘I persuaded Iturna to rescue her unfortunate brother and judged (it appropriate) [for her] to venture for a better life’

The accusative argument [Iturnam] is the subject of two ECM clauses dependent on the finite Vs [probavi] and [suasi], coordinated by [et]:

- a) [ego probavi] Iturnam<sub>ACC</sub> succurrere<sub>INF</sub> misero fratri  
 b) [ego suasi] Iturnam<sub>ACC</sub> audere<sub>INF</sub> pro vita maiora

The derivation would proceed along the same lines of (46), with [Iturnam] defined within the mental workspace by an  $n$ -tuple of coordinates properly containing the  $n$ -tuple defining the VPs, derived monotonically in separate workspaces. Notice that examples like (45 c) and (56) force us to refine the principle in (53): in this case, an argument is forcing the inclusion of a further dimension (i.e., a further axis) insofar as it is an argument of two separate VPs, derived separately and coordinated (recall the discussion about coordination as an essentially Markovian relation between symbolic objects). It seems that the relevant condition has to do with the establishment of dependencies in general, despite the argument / predicate distinction. While (53) might do the job for single-workspace operations, when the picture gets more complicated, a more complete theory of dependencies is to be invoked. Such a theory that is currently under research within a dynamic model of the mind, allowing (and even requiring, in stronger formulation) mixed dependencies within a single symbolic object.

So far, we have dealt with ‘argumental’ dependencies, involving subjects. Let us see a more extreme case, what would happen with an ‘adjunct’ element, in this case, an ablative absolute. Consider the following Latin example:

- 57) Postquam Saturno tenebrosa in Tartara miso / sub Iove mundus erat, subiit argentea proles (Ov. *Met.* I, 113-114)



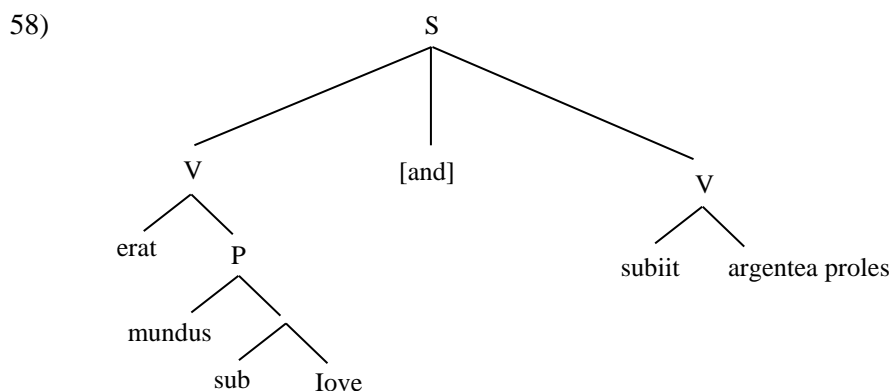
*After Saturn<sub>ABL</sub> dark to Tartarus sent<sub>PartPastPassABL</sub> / under Jupiter world was, rose silver offspring*

‘After Saturn was sent to the dark Tartarus, the world was under Jupiter(‘s command), and the silver offspring rose’

In (57) we have a single ablative absolute [Saturno tenebrosa in Tartara miso] modifying two finite clauses, each with its own nominative subject, [sub Iove mundus erat] and [subiit argentea proles]. In turn, both propositions are coordinated via asyndeton, there being no hierarchical relation between them: the temporal reference of each V [erat] and [subiit] is understood in relation to the ablative absolute (i.e., the world was under Jupiter’s command *and* the silver offspring rose both after Saturn was sent to the dark Tartarus), and not in relation to each other. In our opinion, there are two ways to get round this situation:

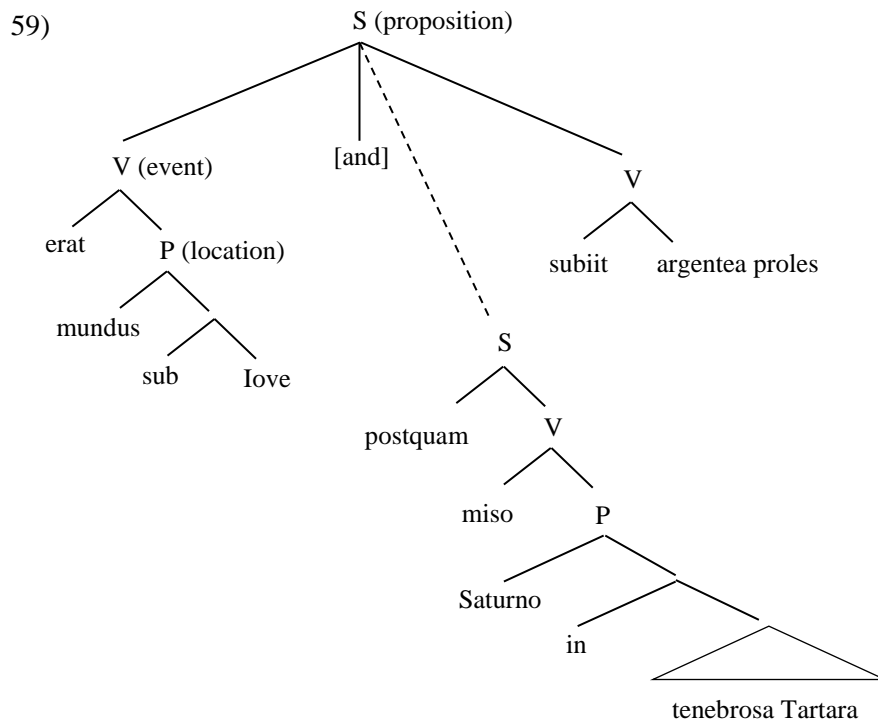
- a) Assume there is an elided copy of the absolute ablative within the domain of one of the VPs
- b) Assume the *same* structural token of the absolute ablative actually modifies both VPs

To the best of our knowledge, there is no literature expanding on (a) for cases like these (involving an element whose specific location within the periphery of the clause is uncertain, unless assuming an extremely rich functional skeleton, to the detriment of representational economy), particularly for a ‘non-configurational’ language like Latin. Let us develop option (b) within the framework sketched above: VPs (without considering the external argument, which can be non-monotonically merged *and* thus would complicate the point we are trying to make here) are monotonic objects, each derived in a separate workspace. We present the coordinated structure as (58):



In (58), arguably, we have only dealt with two axes for each VP, since they have been monotonically assembled (notice that both verbs are unaccusative, therefore, subjects originate as V complements, in traditional terms). The need for a null version of [and] instead of asyndetic coordination is not clear, it is possible that mere non-monotonic Merge of both command units can do the job just as well. However, in this particular case, the element that we have to account for is the ablative absolute, which has scope over *both* VPs. In this case, should we accept the premises in (53) with the caveat made above, the absolute ablative is to be defined in a workspace by a set of coordinates including (say) 3 axes: in that way, it can simultaneously have scope over both VPs

without affecting the label of the whole object, which remains S(entence) or any other symbol we can use to model a whole proposition (including symbols drawn from Montague grammar, for instance, or just a formal representation of the proposition dominated by the node, as in HPSG or other unification-based models, see Shieber, 1986 for discussion). A graphic representation would go along the lines of (59), with the dotted line denoting the added *z* axis for the construction that modifies both members of the asyndetic coordination equally (see Hale & Keyser, 2002; Mateu Fontanals, 2002 for a discussion about the lexical-conceptual structure of unaccusatives, which we assume for [subeo], [sum], and [mitto]):



We have to take into account that each monotonic cascade is in turn expressible in finite-state terms, therefore linearizable independently (in accordance with Uriagereka's 2002a version of Multiple Spell-Out). In this framework, a unit is Spelled-Out as soon as it can, therefore eliminating the necessity to wait until a phase head is merged, as is the case in Chomsky's (2000, 2004) model. The multiple dimensions we have used here are 'flattened' when Spelled-Out, in order to convert a type-0 object into a type-3 object. The direct consequence this proposal has for the architecture of the mind and its computational properties is that the syntax-phonology interface is by no means 'transparent', as it involves dynamic Markovization of *n*-dimensional structure. On the other hand, the semantic component, as pointed out above, is compatible with more complex forms of computation, perhaps exceeding (or being incompatible with) Turing-computability, as Uriagereka (2012: 7) suggests (see Siegelmann, 1995, for a proposal of hypercomputation beyond the Turing-limit; also, Wegner, 1997, 1998 for details about alternative, interactive models of computation, closer to our own stance). As we see once again, the assumptions we make about structure building

operations have far-reaching consequences for the computational properties we assign the mind in charge of performing those operations.

#### 4. Conclusions:

In this paper we have analyzed a range of proposals concerning structure building and structure mapping operations, and the way the resulting symbolic objects are recognized for the purposes of further computations, what is usually referred to as ‘labeling’. We have seen that orthodox generative grammar limited the power of the generative algorithm to two objects, while empowering transformational operations: as a consequence, if natural language is claimed to have a context-free grammar, implemented in a uniformly Turing-machine-like automaton (e.g., Watumull et. al., 2014), the power of the context-free grammar is limited from including linear, Markovian dependencies, to *strictly* 2-D, binarily-Merged-binarily-labeled symbolic structures (Kayne, 1984, 1994). As Lasnik & Uriagereka problematize, it is possible that the generative engine assumed in the Minimalist Program sets both upper and lower boundaries to its own generative power based on intra-theoretical stipulations on labeling and phrase structure (a problem already present in Chomsky & Miller, 1963; as we noticed above): surprisingly, the same problem is found in arguments in favor of *uniform* finite-state grammars (see Kornai, 1985 for an example of the latter perspective; and Baltin, 2003 for discussion on whether considerations of ‘optimal design’ would not favor Markovian models, which are limited to the present state –thus blocking by principle both *look ahead* and *backtracking*- instead of Turing models). We reviewed non-orthodox approaches, in which *n*-ary concatenation is assumed, and argued in favor of a version of *n*-ary Merge in *n*-D mental workspaces, a theory which, we think, adequately captures the semantic properties of problematic structures, including (but not limited to) iteration and coordination. What is more, our proposal *allows* binary Merge as an option (monotonic Merge), thus maintaining the descriptive and explanatory adequacy of the strong points within the Chomskyan Minimalist Program and other forms of phrase structure grammar, while increasing the possibilities of inter-theoretical work by revisiting the generative engine and loosening (or, directly, eliminating) some stipulations constraining its application: the semantic interface determines the best phrase structure for capturing the semantic properties of a certain conceptual object to be linguistically instantiated, provided the syntactic engine consists of free-Merge only (Krivochen, forthcoming). With respect to its implementation on an automaton, we claim that the mental grammar, being dynamic (allowing mixed dependencies within symbolic objects), can go ‘back and forth’ within the Chomsky Hierarchy (for a strict, non-dynamic formalization of grammar types, see Chomsky, 1959: 142-143), selecting at each derivational point the simplest grammar that allows it to parse a given symbolic structure. If labels mark how a syntactic object is to be interpreted in subsequent derivational steps, the algorithm we select for labeling is crucial in *parsing*, and so we have tried to *extend the dynamic character of structure building to structure labeling*. Needless to say, much empirical work is pending. Hopefully this article will contribute (together with some of those we have reviewed here) to generating awareness of the computational consequences of the structure building / mapping algorithms assumed in a particular framework as well as the physical / neurological substratum that licenses the aforementioned computational properties assigned to a certain formal automaton.

## 5. References:

- Adger, David. 2013. *A Syntax of Substance*. Cambridge, Mass.: MIT Press.
- Baltin, Mark. 2003. Is Grammar Markovian? Invited forum lecture to the Korean Association of English Language and Linguistics (Korea University, Seoul).  
<http://linguistics.as.nyu.edu/docs/IO/2637/KASELL-paper.pdf> [Consulted on 09/02/2014]
- Blutner, Reinhard. 2000. Some aspects of optimality in natural language interpretation. *Journal of Semantics*. 17 (3): 189-216.
- Boeckx, Cedric. 2009. The Nature of Merge. Consequences for Language, Mind and Biology. In Piatelli Palmarini et. al. (eds.) *Of Minds and Language*. Oxford, OUP. 44-57.
- Boeckx, Cedric. 2010. Defeating Lexicocentrism. Ms. <http://ling.auf.net/lingbuzz/001130> [consulted on 02/02/2014. Printed as Chapter 1 of *Elementary Syntactic Structures*. Oxford: OUP]
- Bod, Rens. 2013. How hierarchical is language? Nijmegen Lecture, Max Planck Institute, 19 February 2013. Retrieved from <http://staff.science.uva.nl/~rens/hierarchical.pdf> [Consulted on 30/01/2014]
- Borer, Hagit. 2005. *Structuring Sense, Vol I: In Name Only*. Oxford: Oxford University Press.
- Borer, Hagit. 2013. The Category of Roots. Ms. Queen Mary University of London.
- Bošković, Željko. 2008. What will you have, DP or NP? Retrieved from <http://web.uconn.edu/boskovic/papers/nels.illinois.proceedings.final.pdf> [consulted on 02/02/2014]
- Bouma, Goose, Rob Malouf, & Ivan Sag. 2001. Satisfying Constraints on Adjunction and Extraction. *Natural Language and Linguistic Theory*, 19. 1-65.
- Bresnan, Joan. 1976. Transformations and categories in syntax. In Ronald Butts & Jaakko Hintikka (eds.) *Basic problems in methodology and linguistics*. Dordrecht: D. Reidel. 261–282.
- Cecchetto, Carlo & Caterina Donati. 2010. On labeling: Principle C and head movement, *Syntax*, 13: 3. 241-278.
- Chomsky, Noam. 1956. Three models for the description of language. *IRE Transactions on Information Theory* 2: 113–124.
- Chomsky, Noam. 1957. *Syntactic Structures*. The Hague: Mouton.
- Chomsky, Noam. 1959. On Certain Formal Properties of Grammars. *Information and Control* 2. 137-167.
- Chomsky, Noam. 1963. Formal Properties of Grammars. In R. D. Luce, R. R. Bush, and E. Galanter (eds.), *Handbook of Mathematical Psychology*. New York: John Wiley & Sons. 323–418.

- Chomsky, Noam. 1986. *Barriers*. Cambridge, Mass.: MIT Press.
- Chomsky, Noam. 1995. *The Minimalist Program*. Cambridge, Mass.: MIT Press.
- Chomsky, Noam. 2000. Minimalist Inquiries: The Framework. In R. Martin, D. Michaels and J. Uriagereka (eds) *Step by Step: Essays on Minimalist Syntax in Honor of Howard Lasnik*, Cambridge, MA: MIT Press, 89–155.
- Chomsky, Noam. 2004. Beyond Explanatory Adequacy. In Belletti, Adriana (ed.), *Structures and Beyond*. Oxford: Oxford University Press. 104–31.
- Chomsky, Noam. 2008. On Phases. In Freidin, Robert, Carlos P. Otero & María Luisa Zubizarrieta (eds.) *Foundational Issues in Linguistic Theory: Essays in Honor of Jean-Roger Vergnaud*. Cambridge, Mass.: MIT Press. 133-166.
- Chomsky, Noam. 2013. Problems of Projection. *Lingua. Special Issue in Syntax and cognition: core ideas and results in syntax*. 33–49.
- Chomsky, Noam & George Miller. 1963. Introduction to the Formal Analysis of Natural Languages. In Duncan R. Luce, Robert R. Bush, and Eugene Galanter (eds.), *Handbook of Mathematical Psychology 2*. New York: John Wiley & Sons. 269–321.
- Cinque, Guglielmo. 1998. On Clitic Climbing and Other Transparency Effects, talk given at NYU and MIT, February/March 1998.
- Cinque, Guglielmo. 1999. *Adverbs and Functional Heads*. Amsterdam, John Benjamins.
- Collins, Chris. 2002. Eliminating Labels. In S. Epstein & T. D. Seely (eds.) *Derivation and Explanation in the Minimalist Program*. Oxford: Blackwell. 42-61.
- Collins, Chris. 2014. Merge(X,Y) = {X,Y}. Ms. <http://ling.auf.net/lingbuzz/002186> [consulted on 23/08/2014]
- Culicover, Peter. 1999. *Syntactic Nuts: Hard Cases in Syntax*. Oxford: Oxford University Press.
- Culicover, Peter & Ray Jackendoff. 2005. *Simpler Syntax*. Oxford: OUP.
- De Belder, Marijke & Jeroen Van Craenenbroeck. 2011. How to Merge a Root. LingBuzz 001226. To appear in *Linguistic Inquiry*.
- Di Sciullo, Anna-Maria. 2011. A Biolinguistic Approach to Variation. In Di Sciullo, A-M. & C. Boeckx (eds.) *The Biolinguistic Enterprise: New Perspectives on the Evolution and Nature of the Human Language Faculty*. Oxford: OUP. 305-26.
- Di Sciullo, Anna-Maria & Daniela Isac. 2008. The Asymmetry of Merge. *Biolinguistics* 2.4: 260-290.

- Escandell, Ma. Victoria & Manuel Leonetti. 2000. Categorías Funcionales y Semántica Procedimental. In M. Martínez, et. al. (eds.) *Cien años de investigación semántica: de Michél Bréal a la actualidad. Tomo I*, Madrid: Ediciones Clásicas, 363-378.
- Epstein, Samuel & T. Daniel Seely (eds.). 2002. *Derivation and Explanation in the Minimalist Program*. Oxford: Blackwell.
- Epstein, Samuel & T. Daniel Seely. 2006. *Derivations in Minimalism*. Cambridge: Cambridge University Press.
- Gallego, Angel. 2010. *Phase Theory*. Amsterdam: John Benjamins.
- Gómez, Joan. 2010. *Cuando las rectas se vuelven curvas*. Madrid, RBA.
- Green, Georgia. 2011. Elementary Principles of HPSG. In Borsley, R. & K. Börjars (Eds.) *Non-Transformational Syntax. Formal and Explicit Models of Grammar*. London, Blackwell. 9-53.
- Hale, Ken. 1983. Warlpiri and the grammar of non-configurational languages. *Natural Language and Linguistic Theory* 1.1: 5–47.
- Hale, Ken & Samuel Keyser. 2002. *Prolegomenon to a Theory of Argument Structure*. Cambridge, Mass.: MIT Press.
- Hockett, Charles. 1958. *A Course in Modern Linguistics*. New York: Macmillan.
- Hornstein, Norbert. 2009. *A Theory of Syntax: Minimal Operations and Universal Grammar*. Cambridge, Mass.: CUP.
- Hornstein, Norbert & Paul Pietroski. 2009. Basic Operations: Minimal Syntax-Semantics. In *Catalan Journal of Linguistics* 8. 113-139.
- Idsardi, William & Eric Raimy. in press. Three types of linearization and the temporal aspects of speech. In T. Biberauer and Ian Roberts (eds.) *Principles of linearization*. Berlin: Mouton de Gruyter.
- Izard, Véronique, Pierre Pica, Elizabeth Spelke, and Stanislas Dehaene. 2011. Flexible intuitions of Euclidean geometry in an Amazonian indigene group. *PNAS* 108(24). 9782–9787.
- Jackendoff, Ray. 1977. *X' Syntax*. Cambridge, Mass: MIT Press.
- Jackendoff, Ray. 2011. Alternative Minimalist visions of Language. In Borsley, R. & K. Börjars (Eds.) *Non-Transformational Syntax. Formal and Explicit Models of Grammar*. London, Blackwell. 268-296.
- Joshi, Aravind K. 1985 How much context-sensitivity is required to provide reasonable structural descriptions: Tree adjoining grammars. In David Dowty, Lauri Karttunen, & Arnold Zwicky, (eds.)

*Natural language parsing: Psychological, computational, and theoretical perspectives*. Cambridge: Cambridge University Press. 206–250.

Kayne, Richard. 1984. *Connectedness and Binary Branching*. Dordrecht: Foris.

Kayne, Richard. 1994. *The Antisymmetry of Syntax*. Cambridge: MIT Press.

Kayne, Richard. 2011. Antisymmetry and the Lexicon. In Di Sciullo, A-M. & C. Boeckx (eds.) *The Biolinguistic Enterprise: New Perspectives on the Evolution and Nature of the Human Language Faculty*. Oxford: OUP. 329-53.

Kitahara, Hisatsugu. 1997. *Elementary Operations and Optimal Derivations*. Cambridge, Mass.: MIT Press.

Kornai, Andras. 1985. Natural language and the Chomsky hierarchy. In *Proceedings of the EACL 1985, Association for Computational Linguistics*. Stroudsburg: PA. 1–7.

Kosta, Peter & Diego Krivochen. 2014. Flavors of Movement: Revisiting A vs. A'. In Kosta, P., S. Franks, T. Radeva-Bork & L. Schürks (eds.) *Minimalism and Beyond: Radicalizing the Interfaces*. Amsterdam: John Benjamins. 251 – 282.

Krivochen, Diego. 2011. An Introduction to Radical Minimalism I: on Merge and Agree. *IBERIA*, Vol. 3 n° 2. 20-62.

Krivochen, Diego. 2012. *The Syntax and Semantics of the Nominal Construction*. Potsdam Linguistic Investigations 8. Frankfurt am Main, Peter Lang Publishers.

Krivochen, Diego. 2013. On The Necessity of Mixed Models: Dynamical Frustrations in the Mind. Ms. University of Reading.

Krivochen, Diego. Forthcoming. Copies and Tokens: Displacement Revisited. To appear in *Studia Linguistica*.

Krivochen, Diego & Katarzyna Mathiasen. 2012. *Numerals, Numbers and Cognition: Towards a Localist Theory of the Mind*. Poster presented at *Cognitive Modules and Interfaces*, international workshop at SISSA, Trieste, Italy. <http://ling.auf.net/lingbuzz/001653>

Lasnik, Howard. 2011. What Kind of Computing Device is the Human Language Faculty?. In Di Sciullo, A-M. & C. Boeckx (Eds.) *The Biolinguistic Enterprise: New Perspectives on the Evolution and Nature of the Human Language Faculty*. Oxford: OUP. 354-65.

Lasnik, Howard & Juan Uriagereka. 2011. Structure. In R. Kempson, T. Fernando, & N. Asher (eds.) *Handbook of Philosophy of Science Volume 14: Philosophy of Linguistics*. Elsevier. 33-61.

Lasnik, Howard, Juan Uriagereka & Cedric Boeckx. 2005. *A Course in Minimalist Syntax*. Oxford: Blackwell.

- Lebeaux, David. 2009. *Where does Binding Theory Apply?* Cambridge, Mass.: MIT Press.
- Lyons, John. 1968. *Introduction to Theoretical Linguistics*. Cambridge: Academic Press.
- Marr, David. 1982. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. New York: Freeman.
- Mateu Fontanals, Jaume. 2002. *Argument Structure. Relational Construal at the Syntax-Semantics Interface*. PhD Dissertation. Bellaterra. Downloadable at <http://www.tesisenxarxa.net/TDX-1021103-173806/>
- Matushanski, Ora. 2006. Head-movement in linguistic theory. *Linguistic Inquiry* 37(1). 69-109.
- Michaelis, Jens. 2001. Derivational Minimalism is Mildly Context-Sensitive. In M. Moortgat (ed.), *Logical Aspects of Computational Linguistics (LACL '98), Lecture Notes in Artificial Intelligence* Vol. 2014. 179-198.
- Moro, Andrea. 2000. *Dynamic Antisymmetry*. Cambridge, Mass.: MIT Press.
- Murphy, Elliot. 2014a. *Computation and Continuity in the Evolution of Grammar*. MA dissertation, University College London.
- Murphy, Elliot. 2014b. Labels, Cognition and Evolution: Converging Themes. Ms. University College, London.
- Panagiotidis, Phoevos. 2010. *Categorial Features and Categorizers*. Ms. University of Cyprus.
- Panagiotidis, Phoevos. 2014. *Categorial Features: A Generative Theory of Word Class Categories*. Cambridge Studies in Linguistics. Cambridge, UK: Cambridge University Press.
- Pesetsky, David & Esther Torrego. 2007. The syntax of valuation and the interpretability of features. In S. Karimi, V. Samiian, and W. Wilkins (eds) *Phrasal and Clausal Architecture*. Amsterdam: John Benjamins. 262–294.
- Pollard, Carl & Ivan Sag. 1994. *Head-Driven Phrase Structure Grammar*. Chicago: University of Chicago Press.
- Progovac, Ljiljana. 1998. Structure for coordination. *Glott International* 3(7). 3-6.
- Progovac, Ljiljana. 1999. Events and the economy of coordination. *Syntax* 2(2). 141-159.
- Prusinkiewicz, Przemyslaw & Aristid Lindenmayer. 1991. *The Algorithmic Beauty of Plants*. New York: Springer-Verlag.
- Pullum, Geoffrey K. 2011. On the mathematics of *Syntactic Structures*. *Journal of Logic, Language and Information* 20, 277-296.



- Pylyshyn, Zenon. 2007. *Things and Places. How the Mind Connects with the World*. Cambridge, Mass.: MIT Press.
- Richards, Mark. 2007. Dynamic linearization and the shape of phases. *Linguistic Analysis* 33: 209-237.
- Rizzi, Luigi. 2009. Movement and Concepts of Locality. In Piatelli Palmarini et. al. (eds.) *Of Minds and Language*. Oxford, OUP. 155-168.
- van Rooy, Robert. 2004. Relevance and bi-directional OT. In R. Blutner and H. van Zeevat (Eds.), *Pragmatics in Optimality Theory*. New York: Palgrave-Macmillan, 173-210.
- Ross, John Robert. 1967. *Constraints on Variables in Syntax*. PhD dissertation. MIT.
- Seely, T. Daniel. 2006. Merge, Derivational c-command and subcategorization in a label-free syntax. In Boeckx, C. (ed.) *Minimalist Essays*. Amsterdam: John Benjamins. 182-217.
- Shieber, Stuart. 1986. *An Introduction to Unification-Based Approaches to Grammar*. Brookline, Mass.: Microtome Publishing.
- Siegelmann, Hava. 1995. Computation Beyond the Turing Limit. *Science*, 268. 545-548.
- Stowell, Tim. 1981. *Origins of phrase structure*. PhD Diss. Cambridge MA: MIT.
- Uriagereka, Juan. 1998. *Rhyme and Reason*. Cambridge, Mass.: MIT Press.
- Uriagereka, Juan . 2002a. Multiple Spell-Out. In Uriagereka, J. *Derivations: Exploring the Dynamics of Syntax*. London: Routledge. 45-65.
- Uriagereka, Juan. 2002b. Warps: Some thoughts on Categorization. In Uriagereka, J. *Derivations: Exploring the Dynamics of Syntax*. London: Routledge. 288-317.
- Uriagereka, Juan. 2005. A Markovian Syntax for Adjuncts. Ms. UMD.
- Uriagereka, Juan. 2008. *Syntactic Anchors: On Semantic restructuring*. Cambridge: CUP.
- Uriagereka, Juan. 2012. *Spell-Out and the Minimalist Program*. Oxford: OUP.
- Uriagereka, Juan & Paul Pietroski. 2002. Dimensions of natural language. In Uriagereka, J. *Derivations: Exploring the Dynamics of Syntax*. London: Routledge. 266-87.
- Watumull, Jeffrey. 2014. A Turing Program for Linguistic Theory. *Biolinguistics* 6(2). 222–245.
- Watumull, Jeffrey, Marc Hauser, Ian Roberts & Norbert Hornstein. 2014. On Recursion. *Frontiers in Psychology: Language Sciences* 4: 1017.
- Wegner, Peter. 1997. Why Interaction is More Powerful Than Algorithms. *Comm. ACM*, May 1997. 80-91.

Wegner, Peter. 1998. Interactive foundations of computing. *Theoretical Computer Science* **192**. 315-351.

Wurmbrand, Susi. 2014. The Merge Condition: A syntactic approach to selection. In Kosta, P. et. al. (eds.) *Minimalism and Beyond: Radicalizing the interfaces*. Amsterdam: John Benjamins. 130-167.

Zagona, Karen. 1988. *Verb Phrase Syntax*. Studies in Natural Language and Linguistic Theory. D. Reidel Publishing Co., Dordrecht (Kluwer Academic Publishing.)

Zoerner, Edward. 1995. *Coordination: The Syntax of &P*. Doctoral Dissertation, University of California, Irvine.