On workspaces in syntax

Diego Gabriel Krivochen

Abstract:

Syntactic theory has traditionally adopted a *constructivist* approach, in which a set of atomic elements are manipulated by combinatory operations to yield derived, complex elements. Syntactic structure is thus seen as the result or discrete recursive combinatorics over lexical items which get assembled into phrases, which are themselves combined to form sentences. This view is common to European and American structuralism and different incarnations of generative grammar, transformational and non-transformational. Since at least Uriagereka (2002), there has been some attention paid to the fact that syntactic operations must apply *somewhere*, particularly when *copying* and *movement* operations are considered. Contemporary generative theory (e.g., Chomsky, 2019) has thus somewhat acknowledged the importance of formalising aspects of the *spaces* in which elements are manipulated, but it is still a vastly underexplored area. In this paper we explore the concept of *workspace* and its role in current generative theory, aiming at a precise characterisation of what workspaces are and how their properties determine possible syntactic configurations. We further analyse the consequences of conceptualising 'syntax' as a set of operations that affect spaces rather than combine discrete elements.

Keywords: Syntax; workspace; phrase markers; copy; topological space

1. Introduction

Phrase markers, or structural descriptions for natural language sentences, are usually conceived of as *sets*. Transition rules (as in classical formal language theory; Hopcroft & Ullman, 1969; Chomsky, 1957) or stepwise discrete combinatorics (the Minimalist Program's *Merge*; Chomsky, 1995 and much subsequent work) produce sets of terminal and non-terminal elements; in linguistic terms, these are usually identified as lexical items and phrasal nodes, respectively. In transformational generative grammar, a recurrent topic has been the need to *hold on* to structure, either because it needs to be kept within probing memory for further operations (for instance, indexing) or because it has been subject to a reordering rule. We can exemplify these cases in (1a-c)

1) a. Mary$_i$ thinks [that Peter likes [an old picture of herself$_i$]]
   b. Which picture of herself does Mary think that Peter likes ~~which picture of herself~~?

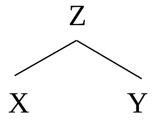
Operations that make reference to previous derivational steps, or to chunks of structure, imply that syntactic objects of variable complexity are *stored* somewhere, where they can be accessed and where rules of the grammar can relate initially distinct and separate derivations. In recent generative theory there has been mentions of *workspaces* (e.g., Müller, 2004: 298; Kato et al., 2016; Collins & Stabler, 2016; Jayaseelan, 2017; Chomsky, 2019; Chomsky et al., 2019), but very often without deep formal discussion or definitions: a detailed discussion about what a workspace actually is, what adding a workspace to the grammar actually means, and what the consequences for the architecture of the grammar and the generative power of the system are is still missing in contemporary generative theory. The purpose of this paper is to make explicit exactly what thinking about *workspaces* commits us to in formal terms, and how we can make use of mathematically explicit characterisations of spaces to our advantage in syntactic theory. In order to do this, we will need to define what spaces are, which in turn will require the introduction of some core notions in topology. This is important and necessary since the tools that topology puts at our disposal can help us define the properties of spaces that are relevant for syntactic computations and therefore evaluate recent proposals critically at both theoretical and empirical levels.

It is important to note that viewing phrase markers as topological objects is not necessarily a new idea: already Bach (1964: 71) formulates conditions on phrase markers (P-markers) in terms of their 'topological' [sic] properties:

> *A proper P marker (when represented in tree form) is* ***a topological structure of lines and nodes*** *conforming to the general requirement that a unique path be traceable from the termination of every branch to the point of origin of the whole tree (or for that matter from any node to any other node)* (our highlighting)

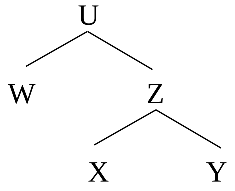
This perspective allowed for the formalisation of conditions over structural descriptions in graph-theoretic terms (e.g., Zwicky & Isard, 1967; McCawley, 1968; Morin & O'Miley, 1969; Kuroda, 1976; also Arc Pair Grammar; Johnson & Postal, 1980 and its spiritual successor, Metagraph Grammar; Postal, 2010. More orthodox generative analyses in graph-theoretic terms are to be found, e.g., in Kracht, 2001; Beim Graben & Gerth, 2012). In these works, operations apply to nodes, creating or deleting edges, in order to establish syntactic dependencies. Let us see a simple example. Assume that we have a phrase marker in which objects X and Y are in a local relation, as represented in (2):

2)



Now suppose that there is some relation *R* between X and Y: for instance, say X theta-marks Y. That relation needs to be maintained throughout the derivation, or reconstructed at the level of semantic interpretation if disrupted by a reordering or deletion rule. We have seen some problems with the latter option, so we would like to give some general prospects to explore the former. Let us now introduce a further element in the derivation, W, which requires a local relation with Y in order to satisfy some requirement (which one in particular is not relevant for the present argument). W is external to {X, Y}, following a cumulative approach to derivational dynamics (which Chomsky, 1995: 190 encodes in the so-called *Extension Condition*[1]):

3)



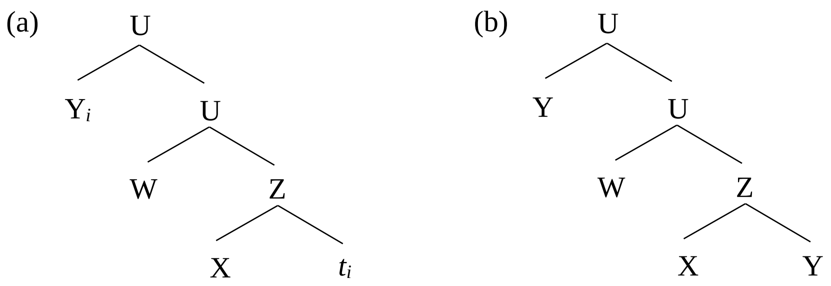
But, what happens if a local configuration between W and Y is required (because, for instance, Y satisfies a criterial feature on W), and such relation cannot hold if X is in between? A *displacement-*

[1] The EC was initially formulated as follows:

> *Suppose we restrict substitution operations still further, requiring that Ø be* ***external*** *to the targeted phrase marker K. Thus, GT and Move-α extend K to K*, which includes K as a proper part* (Chomsky, 1995: 190).

as-movement approach can either (a) move Y to a higher position in the checking domain of W (extending U), outside the scope of X leaving a co-indexed trace behind (the so-called *trace theory*), or (b) *copy* Y and *re-introduce* Y in the derivation (the so-called *Copy Theory of Movement* CTM, or *Copy+Re-Merge theory*; Chomsky, 2000; Uriagereka, 2002; Nunes, 2004; Johnson, 2016 and much related work). Both options are diagrammed below:

4)



In both cases, the structure is extended by means of extra nodes: in (a), we add a *trace* of Y, an index to Y and *t*, and expand U; in (b) we add a copy of Y and similarly expand U. In both cases, there is a local relation between W and Y, as required (because there is no other head between these two nodes), but at the cost of introducing inaudible structure. Moreover, the very idea of *copying* requires not only an operation that takes Y and somehow produces another Y, but this has to happen *somewhere*: an explicit workout of the concept of *workspace* seems to be unavoidable, yet there is no explicit characterisation of what *workspaces* are and how they interact with the generative procedure in mainstream Minimalism. To give a recent example, Chomsky et al. (2019: 236, 245) explicitly say that

> *MERGE operates over syntactic objects placed in a workspace: the MERGE-mates X and Y are either taken from the lexicon or were assembled previously within the same workspace*
>
> *All syntactic objects in the lexicon and in the workspace WS are accessible to MERGE; there is no need for a SELECT operation (as in, e.g., Chomsky 1995). WS represents the stage of the derivation at any given point.*

In this context, the lack of specific accounts of the properties of the workspace where operations are supposed to apply, or even an explicit definition of *what a workspace is* is rather surprising, and many questions arise: does the stage of the derivation at any given point include all objects in the lexicon? How is lexical insertion formulated in such a system? What are the consequences of this assumption for algorithms of probing within the workspace? (e.g., the establishment of filler-gap dependencies), among others. These questions cannot be answered if we do not first know exactly what a workspace is. The absence of detailed discussion about the nature and properties of the workspace is also concerning because the properties of the workspace may impose hard conditions on the operations that can apply; if syntactic operations apply to objects in a space, those objects need to be characterised as array of components in that space. In turn this has deep consequences for an account of *dependencies* between objects (see **Section 5**): the very notions of *local* and *non-local* dependency need to be reconceptualised depending on what is available at every derivational step. Chomsky et al. (2019: 236) refer the reader to Collins & Stabler (2016), who in turn define that:

> ***Definition 10.*** *A stage (of a derivation) is a pair S = <LA, W>, where LA is a lexical array and W is a set of syntactic objects. In any such stage S, we will call W the workspace of S.*
>
> […]

> *by convention we will reserve the term "syntactic object" for those elements built up in the course of the derivation and contained in the workspace.*

In this context, and considering Chomsky's view in comparison to Collins & Stabler's (Müller's 2004: 298 goes along the lines of Collins & Stabler's), it is unclear whether the workspace is (i) a syntactic object (or set thereof) or (ii) a buffer / memory stack / working memory where syntactic operations apply[2]. If the workspace is the phrase marker itself (or the structure plus the *entirety of the lexicon*, as in Kato et al., 2016: 35), then how does it help at all in properly formulating a theory of *copies* (vs. occurrences or repetitions)? Does the workspace play any role in defining locality relations (can it, if it also comprises the lexicon)? Let us focus on *copying*. In both senses of *workspace* the problem of having to define exactly how and where copies are temporarily stored (among other issues) still arises. Interestingly, Collins & Stabler (2016) do note that

> *In minimalist literature, the term "workspace" is also used in a sense where two syntactic objects which are being built in parallel occupy two different workspaces. These two different workspaces are combined at some point in the derivation (see Nunes 2004: 140). We do not use the term "workspace" in this sense in our formalization. At any stage in the derivation there is only one workspace. Formalizing the alternative in our framework would not be difficult.*

However, that formalisation has not been made, to the best of our knowledge. Thus, the veil around the concept of *workspace*, what it is, and what it does, remains firmly in place.

The mention of a *workspace* in the formulation of the structure building operation (External Merge) is only in some sense a novelty, since it has been around (more or less explicitly) in several Minimalist accounts of structure mapping (i.e., Move / Internal Merge): 'moving' a syntactic object has been looked at in terms of Copy + Re-Merge of that object (see Chomsky, 2000; Nunes, 2004; Johnson, 2016 for a variety of perspectives). The Copy-based approach has some fundamental problems, pertaining to the lack of explicitness about the specific mechanism involved in Copying. Stroik & Putnam (2013: 20) formulate the issue very clearly:

> *To "copy X" is not merely a single act of making a facsimile. It is actually a complex three-part act: it involves (i) making a facsimile of X, (ii) leaving X in its original domain D1, and (iii) placing the facsimile in a new domain D2. So, to make a copy of a painting, one must reproduce the painting* ***somewhere*** *(on a canvas, on film, etc.), and to make a copy of a computer file, one must reproduce the file somewhere in the computer (at least in temporary memory).* (highlighting ours)

Note that this issue arises in both versions of *workspace* seen above: the workspace as a syntactic object or as the space where syntactic operations apply (and which properly contains the phrase marker being built). In addition to the problem posed by lexical selection and lexical insertion in the new definition of MERGE, new questions arise. Is the workspace co-extensive with the phrase marker or does it contain the phrase marker? When a syntactic term is being copied (including sidewards

[2] In programming languages like Python, expressions are assigned addresses, which are accessed in the execution of a program by means of variables. It is even possible to access the memory address by means of a specific function, *id*(…). It is also possible to create *local variables*, which get created every time a function is called and erased when the function returns an output; this kind of operation would be analogous to the procedure to create *copies* in movement operations (since the copy is created and stored only temporarily, to be merged later in order to satisfy some featural requirement). In this case, we need two 'spaces': the 'shell' (where the program is executed) and the list of objects in memory (which contains the address, type, and value assigned to each variable). We will come back to the issue of addresses in **Section 4**.

movement, as in Nunes, 2004), the original position of this object and its target position are structurally distant, which also means there are derivational steps in between the introduction of these two positions: where is the copy kept active / accessible throughout this process? And, furthermore, if syntactic movement is triggered by the need to valuate / check / delete uninterpretable features on the moved element, the *goal* of Agree (and possibly also on the probe), how is the system capable of relating the pre-movement and the post-movement instances of the relevant syntactic object if their featural composition is different (Krivochen & Saddy, 2016)?

This problem, the lack of attention to *where* syntactic operations apply (i.e., the lack of a systematic analysis of syntactic workspaces and the way in which they interact with, affect, or constrain operations), is not exclusive to the Minimalist *copy theory of movement* or the revival of Generalised Transformations in the original definition of Merge (see fn. 1). Consider Fiengo's (1977: 44-45) decomposition of the general rule *Move-NP* within the framework of the *trace theory* (which was replaced by the *copy theory* in Minimalism; see Chomsky, 2000 for some discussion):

> *…movement of NP_i to position NP_j (where A and B are the contents of these nodes) in (30) yields (31) as a derived constituent structure.*
>
> *(30) … NP_j … NP_i …*
> | |
> *A*   *B*
>
> *(31) … NP_i … NP_i …*
> | |
> *B*   *e*
>
> *On this view, NP_i and its contents are copied at position NP_j, deleting NP_j and A, and the identity element e is inserted as the contents of (in this case the righthand) NP_i, deleting B under identity.*

Note that, once again, we need to copy NP*$_i$* at B and store it somewhere for a derivational step[3], before inserting it at A. It is important to point out that the problem of specifying *where* operations take place arises both in the case of External and Internal Merge. Stroik & Putnam (2013: 21) point out that the distinction between EM and IM can be rethought within a *Copy-only* system in which differences are determined by the source and the goal of the *Copy* operation (Lexicon-to-phrase marker vs. phrase marker-to-phrase marker): this reworking of IM and EM, unlike the orthodox Chomskyan version, makes it explicit that IM and EM differ in terms of the *spaces* that get accessed in each case and how the targeted syntactic object is affected –whether the space gets extended or not- (see also Stroik, 2009 and Putnam & Stroik, 2010). EM is actually more complex than the latter, in some (informal)

[3] It is worth remembering at this point that the computational system in generative grammar, from the early days of LSLT and the Standard Theory to contemporary models, is an example of the so-called Von Neumann architecture (Von Neumann, 1945). In this architecture, which is implemented by Turing Machines, at any given time only a *single agent* in the computational architecture could be active. This condition translates to an inherent *sequentiality* of computation. Consider, for instance, that in rewriting rules, the 'Traffic Convention' ensures that rewriting applies from left to right, one symbol at a time. Thus, in a sequence XYZ, where X, Y, and Z are intermediate, or 'nonterminal', symbols, a rule rewriting X must apply before a rule rewriting Y. In the case in which we have XY, however, rewriting either first results in equivalent derivations, at least formally. See Chomsky (1956: 117) for some discussion; and Chomsky (2008, 2013) for a phase-head driven derivational system which requires all probing operations triggered by a phase head to proceed simultaneously (this is particularly problematic because even in a Turing machine there should be one instruction per search and instructions are carried out sequentially).

sense, as it involves a relation between *two* distinct spaces, and possibly a further operation of *selection* such that only some elements of the Lexicon are used in a given derivation (e.g., Chomsky, 2000: 101; see also Chomsky, 2012: 3). This is a direct and unavoidable (as far as we can see) consequence of dissociating *lexicon* from *syntax* and these two from the 'interfaces': a core assumption in lexicalist generative grammar is that syntax, semantics, and morpho-phonology are distinct components and that of these only the syntactic component produces structure. It is interesting to note that the operation Transfer, which takes syntactic domains and sends them to the interfaces, has been looked at from the perspective of *what gets transferred* (Chomsky, 2001; 2004; 2013; 2015); this is a crucial aspect of *phase theory*. Deciding which the phase heads are (only C, *v*? Also T, varying parametrically? -e.g., Gallego, 2010- Maybe D as well? -Chomsky, 2000-) and whether it is the complement of the phase head or the full phase that gets transferred (e.g., Epstein et al., 2015; Bošković, 2016) have been rather major questions in the Minimalist agenda. However, little if anything has been said about *where* these syntactic objects are transferred *from* and whether the space *to* which they are transferred has the same properties as the source. In other words: are the interfaces isomorphic to the syntactic workspace? If so, why and how? If not, why not and in which ways? One way to think about this is in terms of multiple-tape automata and multiple stacks instantiating memory buffers (see Uriagereka, 2014, 2018 for some discussion that seems to go in this direction). In such a view, workspaces are *tapes* in a traditional automata-theoretic sense (Turing, 1936; Hopcroft & Ullman, 1969), and syntactic computation proceeds by transitioning between states until the computation halts because the input tape does not satisfy the structural description for any rule. We will not review the extensive literature on formal language theory or its applications to natural language grammar, but it is worth noting that a strongly derivational system based around the notion of *cycle* faces certain difficulties when mechanically implemented in an automaton of the kind suggested in the literature[4]. Coincidentally, non-local dependencies are the ones that motivated the notion of workspace in syntactic theory to begin with.

In this paper we address some of the problems and questions that arise when the notion of a *workspace* is embedded in the context of syntactic theory with the purpose of aiding in assigning structural descriptions to natural language strings, and propose a way in which thinking about syntactic workspaces in terms of *topological spaces* (rather than stack tapes or other kinds of mechanistic memory buffers) has some important empirical and theoretical consequences. We are primarily concerned with two aspects in the analysis of syntactic dependencies involving X, Y, W as in (4): (i) the *distance* between X and Y and W and Y in the definition of syntactic relations, and (ii) the properties of the *spaces* where these dependencies are defined. These concerns are (not so) implicit in the idea of *long distance dependencies* and *discontinuity*, a major topic in syntactic research (see, e.g., Wells, 1947; Ross, 1967; Postal, 1998; Sag, 2010; Putnam & Chaves, to appear, for a healthy variety of perspectives). In this paper we will aim at defining *workspace* in a mathematical sense: the syntactic workspace will be defined as a *topological space*. In this context, we will focus on a topological interpretation of the notion of *distance* between elements in that space that we can use to shed new light on the problem of syntactic dependencies assuming with MGG that

[4] As far as we know, the only explicit implementation of a derivational system without representations in automaton form is Medeiros' (2018) ULTRA model, which is based on a Context-Free stack-sorting mechanism which acts as a mapping between two arrays: the input string and the so-called *functional sequence* (Cinque, 2004 and much related work). The ULTRA model is a very robust method to derive legitimate orders *within a single derivational space* (as can be seen in Medeiros' account of Greenberg's Universal 20), but its applicability is less straightforward when we deal with instances of *substitution* or *adjunction* (Joshi, 1985; Frank, 2002, 2013). This is so because there does not seem to be a clear way to implement dependencies across tapes or sorting mechanisms, each corresponding to a local workspace.

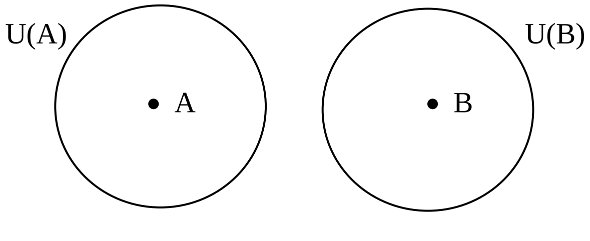
operations (in particular, Copy and Re-Merge, but also indexing and the determination of structural contexts for occurrences of syntactic objects) occur *somewhere*. However, we will depart from MGG in the characterisation of the operations themselves: their input, their inner workings, and their output. In MGG, MERGE (and previously, IM and EM) combine recursively syntactic objects in a workspace, the syntactic operations affect the syntactic structure making it grow (*as per* the Extension Condition) but do not, at least explicitly, affect the workspace. In addition to providing a definition of *workspace* and analysing its consequences, we will propose that computational operations transform a *space* rather than combining (Merging, concatenating, etc.) a set of discrete syntactic objects. It must be borne in mind, however, that we present *a* topological view of syntax, not the only possible one. The mathematical properties of spaces that we will describe here may be compatible to different extents with several versions of MGG including Chomsky's recent 'reformulation' of Merge (Chomsky, 2019; Chomsky et al., 2019; see also Epstein et al., 2015; Kato et al., 2016).

2. On some properties of topological spaces

In order to fully understand what assuming *workspaces* in syntax commit us to, some definitions are in order. First, we need to introduce the concept of *topological space* (see Sutherland, 2009: Chapter 5 for basic notational and terminological points). A topological space is defined as *a set of points*, along with *a set of neighbourhoods for each point*, which satisfy a set of axioms relating points and neighbourhoods. This is a very general definition, and we need to get into some details.

Let A and B be points in a space X. Then, we need to define the neighbourhoods of A and B, call them U(A) and U(B). If *X* is a topological space and A is a point in X, the neighbourhood of A is a subset *U* of *X* that includes an open set *V* containing A (Reid & Szendroi, 2005: 108). In simpler terms, the neighbourhood of A in X is a set properly containing A where one can move that point some amount without leaving the set. If we now consider distinct points A and B in X, we can now define conditions pertaining to the relation between U(A) and U(B), which will define different kinds of topological spaces. A topological space *X* is *Hausdorff*, or T_2, or *separated*, if any two distinct points in *X* are separated by disjoint neighbourhoods. It is *completely Hausdorff* if any two distinct points in X are separated by disjoint *closed* neighbourhoods. The distinction between closed and open neighbourhoods is essential, since bringing points closer together (thus affecting the *distance* function between them) can make their neighbourhoods intersect if these are open[5]. A set is *open* iff it is a neighbourhood for every one of its points, and *closed* otherwise. We can provide a graphical representation of disjoint closed neighbourhoods for A and B:

5)

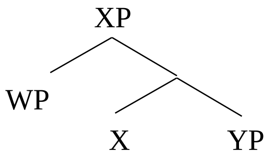


The notions of open and closed neighbourhoods should resonate with the syntactician; after all, the projection of a head H can be characterised as its neighbourhood and edge phenomena also

[5] We can choose between different axioms which relate points and their neighbourhoods, and the specific axioms that we choose gives us a classification of spaces. Furthermore, we can define functions that take us from one kind of space to another (for technical details, see e.g., Willard, 2004; Hazewinkel, 2001; Sakai, 2013).

require the definition of an appropriate *metric* with respect to the head of an endocentric structure. In other words, WP and YP can be defined as the neighbourhood of X in (6):

6)



In this context, if WP is accessible from outside XP, then we can define the neighbourhood of X to be *open*. Accessibility is a central concept in contemporary generative theory, which is heavily based on operations over features: valuation, inheritance, sharing, donation (Chomsky, 2008; Ouali, 2010; Epstein et al., 2015; Zeijlstra, 2020); to such an extent that other operations, like labelling, are seen as parasitic on Agree (Chomsky, 2013, 2015; see also Zeiljstra, 2020). This of course represents a drastic departure from the framework of formal language theory that generative grammar was born out of, since labels no longer constitute a set of nonterminal nodes to be manipulated by rewrite rules, but are rather the output of other syntactic operations (Minimal Search, Agree, etc.). These operations rely on syntactic objects being *close enough* to establish a dependency: from the beginning of Minimalism, economy principles like Minimal Link and Shortest Move (Chomsky, 1995: 297) were aimed at reducing computational complexity in derivations by re-casting long-distance dependencies into sets of local chains (see Martin & Uriagereka, 2014 for discussion). We need to consider the issue of *accessibility* in more detail.

Specifically, for instance, Chomsky's (2000, 2001) Phase Impenetrability Condition depends on how much the *edge* of a syntactic object comprises; in other words, what counts as *close enough* to the phase head to undergo Transfer:

PIC$_1$ = *In phase α with head H, the domain of H is not accessible to operations outside α; only H and its edge are accessible to such operations* (Chomsky, 2000: 108)

PIC$_2$ = *The domain of H* [a phase head] *is not accessible to operations at ZP* [the next phase]*; only H and its edge are accessible to such operations.* (Chomsky, 2001: 14)

Second-order conditions over operations like Agree (including varieties of Minimality; see e.g., Rizzi, 2016) also crucially depend on there being an unambiguous definition of *distance* between points in the space where the phrase marker is defined. Counting the number of edges between objects A and B, where A probes for B (as in Kayne, 1984) requires taking the graph-theoretic view of phrase markers literally (such that structural descriptions are sets of *vertices* connected by *edges* where dependencies between syntactic objects are defined in *paths*), which in turn undermines its cognitive plausibility (as has been pointed out by Chomsky himself). This view has additional empirical problems, analysed in detail in Krivochen (2018b) (see also Hornstein, 2009 for a different perspective). But even if we wanted to maintain the requirement that structural descriptions have only the format of binary-branching trees, the problem of defining the properties of the spaces where these trees are derived and how to establish (or block) relations between distinct local domains in a way that does not require introducing additional principles (like the PIC itself) but rather follows from fundamental properties of these spaces remains unaddressed. It may thus be worth turning towards the question of whether there are properties of the 'workspace' which can deliver these properties.

We must get into some detail about *types* of spaces. Intuitively, points in a space can be close or far apart to different degrees: we can call the function that defines just how close or far apart points are the *metric* of the space. A *metric space* is a set of points together with a *metric* defined over that set, which specifies the *distance* between members of that set. We have suggested above that *distance* is a crucial notion in theoretical syntax; now we will see how a formal definition of *workspace* in topological terms can help us capture the theoretical insights. The distance *d* over a set X is a function defined on the Cartesian product X × X; *d* will be called a *metric* iff the following properties hold (Searcóid, 2006; Kaplansky, 1977; Sutherland, 2009):

7) For *x*, *y*, *z* points in a *metric* topological space,

a. $d(x, y) > 0$ if $x \neq y$ (*positive property*)
b. $d(x, y) = 0$ *iff* $x = y$ (*identity property*)
c. $d(x, y) = d(y, x)$ (*symmetric property*)
d. $d(x, z) \leq d(x, y) + d(y, z)$ (*triangle inequality*)

The distance function $d(x, y) \rightarrow |x - y|$ defined on $\mathbb{R} \times \mathbb{R}$ (the set of real numbers) is called an *Euclidean metric* on $\mathbb{R}$. Distances in Euclidean spaces, except in special cases, are not only real, but also positive. We also want to preserve *topological distinguishability*: two *distinct* points A and B in X can be arbitrarily near or far apart, but never have 0 distance (given 7a, b). A and B have a 0 distance in a *metric space* iff A = B.

The *triangle inequality* is a crucial property: it determines that distances in metric spaces *sum*: informally, if A is *m* away from B and B is *n* away from C in a line defined in X, then A is *m + n* away from C. This is an essential property of metric spaces, because it allows us to formulate the notion of *closeness* in comparative terms, such that *A* is *closer* to *B* than *C* if $d(A, B) < d(B, C)$ (i.e., if $m < n$). What *distance* is measured in terms of may vary (total number of nodes, number of cyclic nodes, number of nodes and edges, number of potential governors for a specific syntactic relation…), but its importance cannot be denied for the theory of syntax. We can see a very early example of an explicit use of *distance* in the formulation of a transformational rule in Rosenbaum's (1965: 10) formulation of what later became *equi NP deletion*:

A NP_j *is erased by an identical* NP_i *if and only if there a* S_α *such that*

i) NP_j *is dominated by* S_α

ii) NP_i *neither dominates nor is dominated by* S_α

iii) *For all* NP_k *neither dominating nor dominated by* S_α ***the distance between*** NP_j ***and*** NP_k ***is greater than the distance between*** NP_j ***and*** NP_i***, where the distance between two nodes is defined in terms of the number of branches in the path connecting them*** (highlighting ours)

More recently, the *anti-symmetric* perspective on phrase markers (Kayne, 1984, 1994, 2018; Moro, 2000), which strongly advocates for *a priori* reasons to have uniform binary-branching trees, also makes explicit reference to *distances* between nodes in trees; the heavy use of set-theoretic terminology must not obscure the fact that graphs are sets as well (see e.g., Wilson, 1996). Kayne's (1984) take on the Empty Category Principle and Chomsky's (1995) Shortest Move and Minimal Link Condition economy principles, to give just two examples, crucially depend on there being a way to determine, given syntactic objects A, B, and C, whether a 'minimal' operation (e.g., MERGE) can relate A and B or A and C. Syntactic operations over features or feature bundles may thus be constrained in terms of how much structure is there available to *probe into*; if such operations apply to syntactic objects in a workspace, then defining a metric on that space becomes an important part of an adequate meta-theory for these syntactic operations. Chomsky (2019: 280) goes as far as equating *accessibility* in the workspace with *recursion*, although no argument is provided to this effect (and recall the difficulties posed by defining the workspace as the phrase marker itself). We may speculate that this identification is due to the assumption that if something is accessible it may be used in future operations, but this leaves aside and unaccounted for the issue of lexical selection and the construction of lexical arrays. In other words, it side-steps the issue of how the system knows what elements it has to work with. The notion of accessibility is left undefined in Chomsky (2019), but if we take the idea that operations apply in a workspace seriously, then we can provide an answer: accessibility is defined in terms of A being contained in an open neighbourhood if it is to be targeted by a distinct syntactic object B. Closed neighbourhoods, thus, prevent accessibility.

Let us go back to the formulations of the PIC above. We can compare the probing space that each of them allows for (see also Müller, 2004, 2011):

8) [$_{\text{CP}}$… [C…[$_{\text{TP}}$… [T…[$_{v\text{P}}$ Ext Arg [*v*…[$_{\text{VP}}$… [V Int Arg]]]]]]]]

In (8), the search space for an operation triggered by T is only Spec-*v* according to PIC$_1$, but all the way to the complement of V according to PIC$_2$ (since T is not a phase head). The extra probing space in PIC$_2$ seems to be required for the movement of VP internal subjects (unaccusative / ergative subjects) to Spec-TP, triggered either by a feature in T before the merger of C or by a feature *inherited* by T from the phase head C. In any case, and empirical issues notwithstanding (e.g., pertaining to the choice of phase heads, the issue of whether more than one specifier position is indeed allowed, etc.), it seems clear that (a) the notion of *distance* is essential in the formulation of syntactic operations and conditions over these, and (b) if this notion is to be implemented in a system that also assumes the existence of a *workspace*, then it stands to reason that *distance* be defined in terms of properties of the workspace.

The concept of *neighbourhood* may provide us with more than we realise *prima facie* in terms of linguistic analysis: in phase theory it is necessary to identify designated nodes as ‘phase heads’, and cyclicity is driven by the opacity effects defined by the PIC; however, since phases are defined purely in terms of syntactic properties (being probes for Agree), there is currently no semantic (propositional) or lexical motivation for phasehood in the Chomskyan version of the theory (see e.g., Chomsky, 2008; Gallego, 2010; Bošković, 2020). However, this need not be so: if cycles are structured as the neighbourhood of lexical heads (as opposed to being the complement of functional heads, as in phase theory), then we can use the concept of *workspace* and the auxiliary notions it forces us to define to our advantage, pursuing a better theory of the syntax-semantics interface. Concretely, we have in mind a proposal like Frank’s (1992, 2002), in the context of Tree Adjoining Grammars (TAGs):

> *Condition on Elementary Tree Minimality: Each elementary tree consists of the extended projection of a single lexical head* (Frank, 1992: 53)
>
> *The syntactic heads in an elementary tree and their projections must form an extended projection of a single lexical head.* (Frank, 2002: 22)

The appeal of TAGs for a syntactic approach that takes workspaces seriously is hard to overstate. In a TAG, the grammar generates a set of elementary trees, which are combined into derived trees by means of operations of *substitution* and *adjunction* (Joshi, 1985). But the main point here is that the development of TAGs found in Frank (1992, 2002) and Joshi & Schabes (1991), known as *lexicalised*

TAGs (LTAGs) allows for a lexically-semantically motivated definition of local domains, in particular if we add the requirement that

> *Every syntactic dependency is expressed locally within a single elementary tree* (Frank, 2013: 233)

We can exemplify the descriptive power of such a definition of local domains. Krivochen & García Fernández (2019, 2020) argue, on empirical grounds, that a monotonic, Merge-based phrase structure grammar provides inadequate structural descriptions for sequences of auxiliary verbs in Spanish, and that considering the notion of *neighbourhood* in syntactic terms is empirically fruitful. Consider (9a) and (9b):

9) a. Juan tiene que estar trabajando
J. Aux.Mod.deont Aux.prog working
b. Juan está teniendo que trabajar
J. Aux.prog Aux.Mod.deont work

Both (9a) and (9b) are grammatical and acceptable sentences in Spanish. Crucially, they are *not synonymous*: in (9a) the obligation pertains to a current, progressive event of working; in (9b) the progressive affects the obligation, but not the event of working. Therefore, (9b) does *not* entails 'John is working', since the progressive only affects the modal, but not the lexical verb. In the aforementioned works, it is proposed that this is so because *estar* does not have scope over *trabajar*, which in phrase structural terms means that *estar* cannot c-command *trabajar* (Ladusaw, 1980; May, 1985 and much related work). Furthermore, there is no evidence that either order is derived from the other via movement or any other kind of structure mapping: both are equally 'basic'. We may consider another example, which features a modal auxiliary <*tener que* + infinitive> and an aspectual phasal auxiliary <*empezar a* + infinitive>:

10) Juan tiene$_{\text{[Modal]}}$ que haber empezado$_{\text{[Phasal]}}$ a trabajar
J. has that to-have started to work
'John has to have started working'

The lexical verb [trabajar] is *not* accesible to [tener que], only to [empezar], with perfective [haber] being a functional auxiliary modifying [empezar]. Specifically, the obligation ([tener que]) pertains to *start* working, not to the whole event of working. Moreover, it is possible that lexical information is lost for further computations in a *phase-based* model: If the projection of Aux, call it AuxP, is defined to be an impenetrable domain, how can the lexical verb be interpreted? Particularly, under the assumption that VP (or *v*P, the difference is immaterial here) is a domain of its own.

This brief discussion serves the purpose of illustrating two points: on the one hand, the inadequacies of structural uniformity and the necessity to chunk structure into smaller domains (in TAG terms, elementary trees), on the other, the relevance of the concept of the *neighbourhood* of a lexical head in structural terms: in a lexicalised grammar, the units of computation are precisely the neighbourhood of lexical heads (Frank, 1992, 2013; XTAG group, 2001). The notion of *metric space* allows us to formulate an explicit definition of *workspace* which is instrumental in defining a framework where locality and accessibility can be linked to properties of the spaces where derivations take place. Furthermore, it allows us to have a characterisation of syntactic terms in that space: as in Sarkar &

Joshi (1997), we can define for each element in the neighbourhood of a lexical head (and the lexical head itself) an *address* which corresponds uniquely to a region of the space[6]

2.1 *An alternative view: ultrametricity*

However, metric spaces instantiate only one kind of topological space. Thus, we need to consider possible alternatives before settling for metric spaces as the way to formalise syntactic workspaces. Here is where *ultrametricity* comes into play. An *ultrametric* space (the term is due to Krasner, 1944), which is a specific instance of topological space, is a set of points with an associated distance function *d* mapped onto the set of real numbers ℝ such that the following conditions hold (Kaplansky, 1977; Artin, 1967; Murtagh, 2004a, b):

11) a) $d(x, y) \geq 0$ (*positive property*)
b) $d(x, y) = 0$ *iff* $x = y$ (*identity property*)
c) $d(x, y) = d(y, x)$ (*symmetric property*)
d) $d(x, z) \leq \max\{d(x, y), d(y, z)\}$ (*Ultrametric Inequality*)

Above, we defined *Hausdorff* and *completely Hausdorff* spaces. As a reminder, a topological space X is *Hausdorff* if, for *x* and *y* distinct, topologically distinguishable points in X, there exists a neighbourhood *U* of *x* and a neighbourhood *V* of *y*, and $(U \cap V) = \{\text{Ø}\}$ (this is called the 'separation axiom', and the Hausdoff characteristic of ultrametric spaces will be very important below). A space is *completely Hausdorff* if any two distinct points in X are separated by *disjoint closed* neighbourhoods: *x* and *y* are separated by *closed neighborhoods* if there exists a closed neighborhood U of *x* and a closed neighborhood V of *y* such that U and V are disjoint (Munkres, 2000; Willard, 2004). This condition on separation imposes stronger restrictions on *completely Hausdorff* spaces than in *Hausdorff* spaces, because the former specify *closed* neighbourhoods: as illustrated in (5a, b) having disjoint closed neighbourhoods for *x* and *y* entails that *x* is not accessible to *y* and *y* is not accessible to *x*, furthermore, no point in the neighbourhood of *x* is accessible to any point in the neighbourhood of *y* and vice versa. In syntactic terms, think of completely opaque domains: a syntactic term all of whose terms are inaccessible to operations triggered from outside that term. It is not clear whether such terms exist in natural language, but this is an empirical question. Plausible candidates for units with closed neighbourhoods are syntactic objects that cannot be embedded (and thus not even their root can be targeted by syntactic rules -like substitution or adjunction-): imperatives (Schmerling, 1982), vocatives, and interjections (Chomsky, 2008: 139).

Ultrametric spaces have interesting topological properties, some of which we summarise here. For instance, only a subset of isosceles triangles is allowed, given the replacement of the *triangle inequality* that holds for metric spaces by the *ultrametric inequality* in (11d). Equilateral triangles are also allowed. But perhaps most relevantly, as a consequence of the ultrametric inequality, every point within a sphere is the center of the sphere (given a constant distance function between distinct points);

[6] In a lexicalised grammar like the one assumed here, the distinction between lexical and functional categories is not one of featural composition or anything of the sort, but rather of the size of its neighbourhood: because neighbourhoods of lexical heads contain (in the version explored in Krivochen & García Fernández, 2019, 2020) the lexical head, plus functional modifiers (aspectual and temporal auxiliaries), and the arguments of the lexical head, the neighbourhood of a lexical head is a superset of the neighbourhood of the same element in a functional use (e.g., Spanish *ir* 'to go' as a lexical unaccusative verb vs. <*ir a* + infinitive>, the periphrastic future form; Italian *avere* -lexical transitive- vs. <*avere* + participle>, a perfective auxiliary). See also Manzini & Savoia (2011) for related discussion.

this is known as *ergocentricity* (Hughes, 2003: 159). This is intimately related to the following consequence of the Ultrametric Inequality[7]:

e) $||x + y|| = ||x||$

where $||\cdot||$ is a *length function* which assigns a number to each element of a group; in this case the number pertains to the distance of a point to the origin of coordinates. Note that this means that, in ultrametric spaces, *distances do not sum*. Gajić (2001: 96) puts it in the following terms:

> ***Remark:*** *Let* [a topological space] $X \neq \emptyset$, [with a] *metric d defined on X by*
>
> $$d(x, y) = \begin{cases} 0, if\ x = y \\ 1, if\ x \neq y \end{cases}$$
>
> [then, the] *so-called discrete metric is ultrametric*

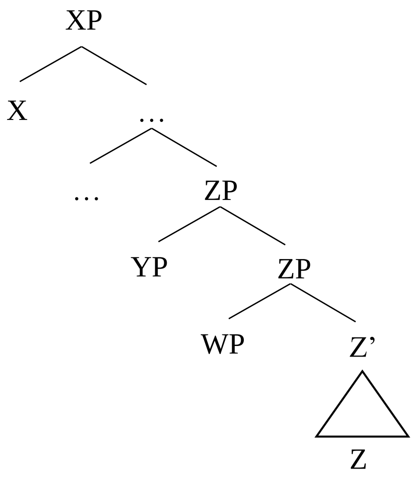
We have introduced ultrametricity because (a) we need to evaluate some alternative to metric spaces in order to properly argue for their adequacy, and (b) ultrametricity has a prominent role in some proposals about the mathematical nature of phrase markers in generative grammar that also adopt a topological perspective (Roberts, 2015; Uriagereka, 2012: 43-45). After all, ultrametricity has been used as a model for taxonomy and hierarchical structure in a variety of disciplines, including physics, data science, and biology (Murtagh, 2004a; Rammal et al., 1986; Gavryushkin & Drummond, 2016). Furthermore, Hughes (2003) presents an explicit equivalence between infinite single-rooted trees (including Cantor and Fibonacci trees) and finite ultrametric spaces (see also Dovgoshey & Petrov, 2019); this is relevant since models of syntactic structure that propose a Fibonacci provenance for X-bar theory are forced to expand the X-bar schema infinitely (e.g., Uriagereka, 2014; Medeiros & Piattelli-Palmarini, 2018); however, we need to note that Roberts (2015) sticks to local X-bar trees. A careful consideration of ultrametricity seems thus to be justified at least conceptually.

Recall the two properties of ultrametric spaces that we highlighted above: (i) every point contained within a sphere is the center of the sphere and (ii) distances do not sum (we are distinguishing these two for expository issues, but they follow from the same property). We need to consider the following question seriously: is it tenable, under present assumptions, to claim that syntactic workspaces are (or can be) ultrametric?

A potential argument in favour of ultrametric phrase markers comes from the use of *equidistance* in the formulation of mapping operations, in particular Agree. In these cases, there is a probe and multiple goals that, despite occupying distinct positions in a binary-branching phrase marker (and thus being more or less distant from the probe in terms of number of nodes), are considered to be at the same distance from that probe for purposes of some specific operation. Let us illustrate the relevant configuration:

[7] A simple proof can be found at https://planetmath.org/UltrametricTriangleInequality

12)



In this context, define the *domain* of X as the set of nodes contained in the maximal projection of X (i.e., XP) which excludes X. Furthermore, let the *minimal domain* of Z be the set of categories that are only locally related to the head[8]. Consider the case where X is a probe for movement: will it target YP or WP? Chomsky (1995: 169) says

> *If α, β are in the same minimal domain, they are equidistant from γ.*
>
> *In particular, two targets of movement are equidistant if they are in the same minimal domain.* (Chomsky, 1995: 169)

(See also Lasnik, 2009; Hornstein 2009: 42, ff.; Boeckx, 2008: 145; for similar definitions of *equidistance*)

This means that YP and WP are equidistant for purposes of operations at X, despite the fact that, strictly speaking, YP c-commands WP asymmetrically. The theory of locality sketched in Chomsky (1995) requires this notion of *equidistance* for reasons related to feature-checking and movement (thus, intra-theoretical requirements). We may provide a couple of examples (which can also be found in the references above). In a case like

13) T seem [to him$_i$] [they$_k$ to like John$_{*i/j}$]

The question Chomsky considers is whether there is anything blocking raising of *they* to Spec-T: note that there seems to be a Principle C violation if *John* is coindexed with *him,* which suggests that *him* c-commands *John*; crucially, Chomsky assumes that *seem* '*has two internal arguments*' (1995: 280), the PP [to him] and the clause [they to like John] (i.e., the experiencer is not an adjunct). Under this assumption, a possible 'solution' (the details of which we will not consider here) is that *they* and *him* are equidistant, which allows *they* to move to Spec-T to check a Case feature without *him* being an intervening element in terms of Minimality.

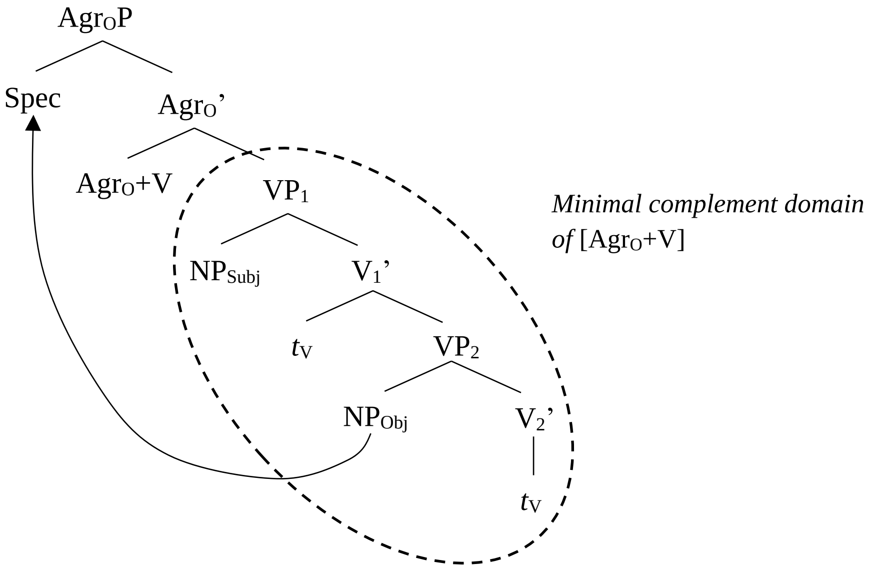
Along the same lines, we may consider (again with Chomsky, 1995) the case in which V raises to Agr$_O$ under Larsonian shells (a configuration like (14) below): both VP shells constitute the *minimal complement domain* (or just *internal domain*) of the complex head [V+Agr$_O$]. Since NP$_{Subj}$ and NP$_{Obj}$

[8] Symbolically,

> *The minimal domain Min(δ(CH)) of CH is the smallest subset K of δ(CH) such that for any γ ∈ δ(CH), some β ∈ K reflexively dominates γ.* (Chomsky, 1995: 274)

are in the same minimal domain, they are equidistant for purposes of operations triggered by the complex head. In this way, NP$_{\text{Obj}}$ can apparently ‘cross over’ NP$_{\text{Subj}}$ in order to check Accusative Case in Spec-Agr$_{\text{O}}$ with the complex [V+Agr$_{\text{O}}$]: the object NP is not *really* ‘crossing over’ the subject and violating a locality condition (Attract Closest / Minimal Link / Shortest Move), since NP$_{\text{Subj}}$ and NP$_{\text{Obj}}$ are *equidistant* from Agr$_{\text{O}}$. Let us diagram the situation:

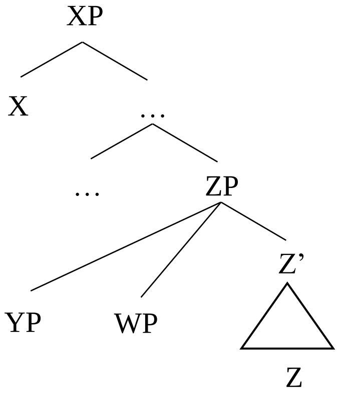
14)



Considering Agr$_{\text{O}}$ *after* V has moved up, we have marked the minimal complement domain of the complex head [Agr$_{\text{O}}$+V]. According to Chomsky’s definition, since NP$_{\text{Subj}}$ and NP$_{\text{Obj}}$ are within the same domain (regardless of the fact that NP$_{\text{Subj}}$ still asymmetrically c-commands NP$_{\text{Obj}}$ and so on), they are equidistant from this complex head and thus either could move to Spec- without violating Shortest Move.

This issue has been taken up in more recent works, which attempt to define a notion of *distance* that is not based on node counting, under the (reasonable) assumption that ‘grammars do not count’. For example, Hornstein (2009: 38) proposes that *distance* is based on comparing the set of nodes involved in a movement path for possible targets, with shorter paths being properly contained in longer paths. Graphs are abandoned in favour of *sets*. The grammar must then contain a mechanism that evaluates the relation between two (or more) sets and is capable of identifying if one of those sets is a subset of the other. If multiple specifiers of Z are equidistant to X since their paths are not proper subsets of each other (ZP need not be counted twice in each set), then (12) is equivalent to (15):

15)



(15) violates the venerable axiom of ‘binary branching all the way down’. This leads us to a further conundrum: extra structure between YP and WP is needed for linearisation purposes (Kayne’s 1994 Linear Correspondence Axiom; see also Uriagereka, 2012) but not for the computation of paths? In this context, both definitions of equidistance (Chomsky’s and Hornstein’s) seem equally stipulative.

What happens in an ultrametric phrase marker? No ‘set comparison’ mechanism needs to be invoked since, by virtue of the ultrametric inequality, all points within the neighbourhood of a head are indeed equidistant. However, even though such a perspective may be useful in some instances (anti-superiority effects, the so-called ‘experiencer paradox’ seen above, and the like), we need to consider the ramifications of the claim that the syntactic workspace is ultrametric. True, it would provide a straightforward characterisation of equidistance (if needed at all) and connect phrase structure with the dynamical frustration between the semantic and morpho-phonological components of language, but it would also pose severe problems. Not the least of which is the fact that things like subject-object asymmetries would become impossible to formulate: under the VP-internal subject hypothesis, both the subject and the object of a transitive construction would be generated in the neighbourhood of V: does that mean that are equidistant from V? The relation between internal arguments and the V is of a different kind, ‘closer’ in a sense, than the relation between the external argument and the V: the Aktionsart of VP is defined at the level of V+Obj, excluding the subject (the well-known alternation between, say, *paint {for two hours / *in two hours}* and *paint the portrait {?for two hours / in two hours}*). Or, even worse: does that mean that all syntactic terms are the center of that local neighbourhood (thus, the identification of the lexical anchor in an elementary tree would become impossible)? Equivalences between infinite single-rooted trees (or finite non-directed trees) and ultrametric spaces notwithstanding, the applicability to syntactic trees is limited: in taxonomical models which make use of ultrametric trees with bottom-up algorithms, the point is to have the distance from the root to any leaf be equal.

In MGG, locality effects are well-documented (Ross, 1967 and much work ever since); the intuition that ‘short’ movement paths are preferred to ‘longer’ ones when both are available (e.g., Rizzi, 1990; Chomsky, 1995) seems to be empirically supported. But, that points towards *metricity* rather than *ultrametricity* as a proper characterisation of the syntactic workspace. So, is *ultrametricity* a useless exercise? Not quite, from our perspective. The properties of ultrametric spaces, while not viable for syntactic structural descriptions, do make thinking about the *lexicon* easier: after all, not only do we have distinguishability in ultrametric spaces, but also the fact that distances do not sum entails that we can think of the lexicon as an associative network in which there are no biases: anything can, in principle, be selected and connected to anything else. Once elements from the lexicon enter into syntactic relations (and only then), we can define variable distances in the context of a phrase marker. Following this reasoning, if the lexicon needs to maintain distinguishability but not have pre-encoded biases (in the form of distances), then an individual’s lexicon is an underlying field of connectivities between unobservable states, which become observable only after Spell-Out. The topology of this field is ultrametric, and thus each element of the lexicon should be connected to every other by a constant distance *d*. When a set of items enter syntactic relations, their distance changes, and this impacts the ultrametricity of the lexicon. With reference to this point, Uriagereka (p.c.) says that

> *The idea is that when you merge, say, “men” into “like arguments” (or some such),* ***you are literally getting “men” to a proximity w.r.t. “arguments” that it would not otherwise have had*** *(as compared to, say, “men” and “boys” or “arguments” and “discussions”, say). As a consequence of the merge, each of the relevant words (understood as information-density peaks within the space) will obtain new conditions.*

What this means, in the present context, is that a linguistic derivation is a *dynamical process,* whose initial state can be described by a certain formalism and which changes in time; the metricity that we can call upon in the definition of local domains as the neighbourhoods of lexical heads, as suggested in the previous section, is not an inherent property of the workspace, but rather the result of perturbations of the initial state of the workspace.

3. Why (and why not) ultrametricity?

The previous section, while objecting to an ultrametric view of syntactic structure, introduced the proposal that the lexicon is best formalised as an *ultrametric space*. Furthermore, we can make the hypothesis stronger by adding the condition that the ultrametric space be *completely Hausdorff* (recall, this means that distinct points have disjoint closed neighbourhoods). The distinction between closed and open neighbourhoods is essential, since bringing points closer together by means of operations over the space can make these points' neighbourhoods intersect if these are open. Metric spaces are typically *Hausdorff*, but not necessarily *completely Hausdorff.* This caveat will be crucial for our conception of what syntax does to the ground state dynamics of the workspace. It is important at this point to give some thought to the question of how these workspaces come to be: here we will propose that the ultrametricity of the ground state of the workspace arises as a consequence of a fundamental *tension* at the core of language.

Uriagereka (2012, 2014), Krivochen (2018a); Krivochen & Lacková (2020), and Saddy (2018) –building on Binder (2008) and Moessner & Ramírez (2006)- worked with the concept of *dynamical frustration*, the resolution of a clash between mutually opposing requirements over a system, as a crucial aspect of the architecture of language in cognition (we need to note, however, that the idea of a *tension* between mutually incompatible tendencies in language appears, however, already in Tesnière, 1959: 21[9]). The concept of *dynamical frustration* appeared in physics in the context of the study of spin glasses (Stein & Newman, 2011). These are 'disordered' magnets, lattices in which electrons are subject to a pairwise antialignment constraint, which makes the system locally frustrated (since an electron has to be changing spin permanently in order to maintain the antialignment with its neighbours). As Rammal et al. (1986: 771, ff.) argue, '*The crucial ingredients in these models* [of spin glasses] *are disorder and frustration*.' Disorder is understood in its usual 'entropy' sense -see Caracciolo and Radicatti (1989) for discussion about the entropy of ultrametric dynamical systems-. Since the system *as a whole* cannot achieve a stable state, instead, multiple locally optimal solutions are found by actants in the dynamical system. However, the concept of *dynamical frustration* has also been fruitfully applied to neural assemblies (Papo *et al.*, 2014) and computability (Binder, 2008).

The idea, when applied to the architecture of language, is that morphophonology and semantics impose orthogonal requirements over syntactic representations: low-dimensional structure

[9] *1. — The possibility of a term in the structural order having, beyond its unique higher connection, two or three lower connections* […] *collides, in its place in a sentence, with the impossibility of a word in the spoken string being immediately in a sequence with more than two adjacent words* […] *In other words, **every structural node is susceptible to the creation of bifurcations, trifurcations, etc…, that are incompatible with linear order**.*
*3. — **There is thus a tension between the structural order, which has several dimensions*** […], ***and the linear order, which has one dimension**. This tension is the squaring the circle of language. Its resolution is the sine qua non condition of speech.*
4. — The tension between the structural order and the linear order can only be resolved by sacrificing at least one linear sequence at the point of placement in the sentence. (Tesnière, 1959: 21. Translation: Susan F. Schmerling. Highlighting ours).

on the phonological side (essentially a Saussurean requirement; see also Scheer, 2013; Idsardi & Raimy, 2013 for finite-state low-dimensional approaches to phonology) and multidimensionality and higher-level relations in the semantic side (see Hinzen, 2009: 31 for a Minimalist perspective on '*multi-dimensional thought*' and how the computational system '*boosts the dimensionality of the human mind*'; also Mori, 2005; Uriagereka, 2008: Chapter 6). The idea that language, as a cognitive capacity, serves as a dynamical resolution of a tension between mutually incompatible requirements is also at the core of the concept of *psychogrammar* in Bever (1975: 64-65), with the difference that in the present view the conflict between the perceptual and production systems never fades away; rather, is at the very core of language. Bever (2008: 479) also calls attention to a 'conundrum' [sic] that appears when considering aspects of adult language behaviour:

> *a. Sentence processing involves complex computation of syntax with whole sentences as domain – **it is vertical***
> *b. Language behavior proceeds serially and incrementally – **it is horizontal.***

Recent research on implicit learning of artificial grammars (Vender et al., 2019; Vender et al., 2020) suggests that humans build on statistical regularities which are superficial in a signal (a series of red and blue dots whose appearance on a screen follows a pattern generated by a grammar) as a way to build generalisations that pertain to the distribution of structurally relevant elements in a way that leads to abstraction over statistical regularities and rule-like behaviour over the course of a modified Simon task. In Krivochen (2020a) we propose a method to construct a model for structures of a simple formal language (thus, 'vertical' restrictions) based exclusively on co-occurrence restrictions (thus, on 'horizontal' restrictions). The point is that linear and hierarchical mechanisms impose distinct (and incompatible) requirements on the computational system in charge of processing external stimuli: a system cannot satisfy these output conditions unless it oscillated between meta-stable states. Instead of resting in its least energy state (an attractor; see Spencer et al., 2009: 109), the system (the 'psychogrammar', in Bever's terms) is in permanent oscillation (Binder, 2008: 322); in the present case, this oscillation takes place between high-dimensional vs. low-dimensional computation, global vs. local requirements, and ultrametric vs. metric spaces.

If, as argued in Uriagereka (2012: Chapter 7); Saddy (2018), and Krivochen (2018a) (based on Tesnière, 1959), the computational properties of 'natural language' are the result of a dynamical frustration between global and local tendencies (corresponding to semantic and morpho-phonological requirements, respectively), then the kind of space delivered by a dynamical frustration should be an adequate description for the initial state of syntactic workspaces. What kind of spaces do dynamical frustrations characterise, then? Murtagh (2004b: 168) puts it this way:

> *"Frustrated optimization problems" are ultrametric, and have been shown as such for spin glass and related special cases.*

If the conditions where syntax operates are those defined by a dynamical frustration, then language may well be one of these 'special cases', from both a computational and a cognitive perspective. However, the proposal in the present paper is that these properties *do not remain constant* throughout the derivational process. A *dynamical frustration* can give rise to an ultrametric space which is progressively transformed (metricised) by means of syntactic operations whereby the atoms of syntactic structure are related.

This process is much better captured if we do not assume from the beginning that the topology of the syntactic workspace is static and immutable. Murtagh (2004a) argues that high-dimensional data processing benefits from an ultrametric treatment (Rammal et al., 1986 provide a method to quantify just ‘how ultrametric’ a dataset is); however, for the definition of local syntactic structures ultrametricity is inadequate. In this context, Saddy (2018) argues that the ground state of neurocognitive dynamics is an ultrametric space which is not only high-dimensional but also unrestricted, lacking the possibility of producing usable outputs. This is so because, if syntactic objects are represented as vectors in the ultrametric space, then the relations between objects needs to be formalised in terms of vector operations; since distances do not sum, there is no way to guarantee that vector operations will yield a cumulative output (Saddy, 2018: 323). The process by means of which creatures map their surroundings can deliver a space in which information is indeed represented, but without a way to relate pieces of information to each other (reinforce connections, bring those points in the space closer), such a system cannot generate usable outputs. The information provided by the sensoria is abundant and complex, and given the usefulness of ultrametric spaces in categorising high-dimensional data, Saddy’s hypothesis receives some support. In syntactic terms, Saddy’s argument is that a formalism defined in an ultrametric space has no conditions that can prevent the generation of local syntactic structure (e.g., a set of lexicalised elementary trees) because there are no heads or edges, given the ultrametric inequality. However, if the space is metricised, then we can define lexical heads and their neighbourhoods, which constitute the atoms of syntax in a lexicalised grammar. Properties of the space where a mathematical construct (including a phrase marker) is defined, then, gives us some hard constraints with respect to the properties that can be ascribed to such construct. In our opinion, this is the most important and interesting aspect that follows from taking the idea of a *workspace* for syntactic operations seriously: the specific kind of space that we assume is the canvas for syntactic operations restricts the class of *adequate* grammars and operations.

As observed above, dynamical frustrations deliver ultrametricity. At the same time, natural language syntax is built on Is it possible to reconcile these two insights (one architectural, the other empirical)? Note that, if there is a dynamical frustration at the core of language (and psychogrammar), then the space defined by that frustration describes the *initial* state of syntactic derivations. At this point, we need to carefully consider the following point: do syntactic operations affect the topological properties of the workspace? In other words: is it possible that the initial state of the workspace is affected by syntactic operations in such a way that its topological properties are changed?

What this suggests is that there may be a way to transform an *ultrametric* space into a *metric* space, which entails introducing a *variable* distance function between points. This has major consequences for the kind of grammatical formalism that can be formulated in each space: on the one hand, we can enjoy the advantages of ultrametric spaces as a characterisation of the initial state of the derivation (distinguishability, constant distances, no biases towards specific relations between elements); on the other hand, when the ultrametric space is metricised, we can define open neighbourhoods of lexical heads: these are elementary trees, the building blocks of syntax.

4. What is, then, ‘syntax’?

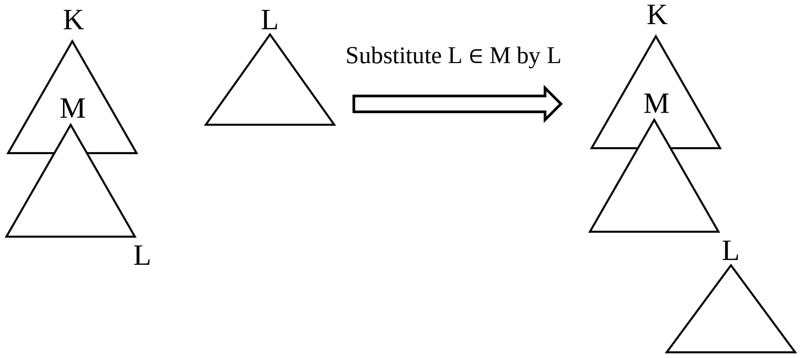
So far we have a characterisation of the workspace where syntactic operations apply and, in that space, we have the elements that appear in structural descriptions defined as points or sets thereof within said space: in this context, elements within the workspace are related when the topology of the space is perturbed, and elements are drawn closer together. The perturbation of an initially ultrametric space disrupts said ultrametricity, yielding a metric space (Saddy, 2018; Krivochen, 2018a). The question now is, what is the role of syntax in this process? If one accepts the proposal that the dynamical frustration at the core of language delivers an ultrametric space, and also the argument that syntax requires metricity, then the role of syntax can be defined to be the metricisation of the space:

syntactic operations partition the space, bringing some elements closer together and making their neighbourhoods intersect. If these are open, then elements within one neighbourhood become accessible from the other neighbourhood. In this context, consider the following condition on a strongly cyclic syntax, from Uriagereka (2012: 75):

> *Whenever a phrase-marker K is divided into complex sub-components L and M* […], *the daughter phrase-marker M that spells-out separately must correspond to an identical term M within K.*

Uriagereka proposes a solution to the problem with which all models of structure interpretation struggle (be it narrowly syntactic or much more general cognitive mechanisms, see e.g. Rabinovich et al., 2014; also Feigenson, 2011 for a flexible approach to chunking and grouping): once an input has been chunked and each part has been subject to an arbitrary set of operations, how to we put everything back together? From a generative-derivational viewpoint, in which structure is built step-by-step by means of discrete recursive combinatorics (e.g., the operation Merge), the question can be phrased as: how can separate *command units* (local monotonically derived phrase markers) be linked? The problem, when asked specifically about linguistic structures, pertains to the relation between strict locality and compositionality: if we consider the cases analysed in **Section 2**, we need both to define local elementary trees, assign local interpretations, and account for how these local units are combined. However, given the ubiquity of chunking operations in cognition, the issue is much more general. What we want to do is provide a way to capture compositionality in both local and long-distance dependencies *without having to invoke additional structure* in the form of non-terminal nodes (see Lasnik, 2011; Krivochen, 2015, 2020b for further discussion on the issue of 'too much structure' that arises in MGG). A simple case we can deal with is that of *substitution* (Chomsky, 1955; Joshi & Kroch, 1985). In traditional phrase structure terms, let K be a term, and let M be a term within K, with a node L in its frontier. Furthermore, let L be a distinct term (not a part of K). Then, we can substitute the node L in M with the sub-tree (term) L:

16)



We have been deliberately imprecise with respect to which L we are referring to in each instance, the reason being that *substitution* works if and only if L ∈ M is *identical* to the root of the separate term L (see Frank, 2002: 17, ff. for discussion). This is a simple case, which can correspond to clausal complementation (see Bresnan, 1971 for an early treatment of clausal complementation in terms very similar to these):

17) John wished [that Mary would go out with him]

(17) contains two clausal domains in a hypotactic relation; each of these clausal domains corresponds to an elementary tree. The bracketed clause, whose lexical anchor is the verb *go out* (call it L) is

subordinated to the elementary tree headed by the verb *wish* (call it M). Then, a derivation of (17) using cyclic *substitution* goes along the following lines (cf. Uriagereka's citation above):

18) a. [$_{\text{M}}$ John wished [L]]]
b. [$_{\text{L}}$ that Mary would go out with him]
c. [$_{\text{K}}$ [$_{\text{M}}$ John wished [$_{\text{L}}$ that Mary would go out with him]]] (via *substitution* targeting L)

In this case, what we have done is link the neighbourhoods of two points, *go out* and *wish*, by identifying the same node in both syntactic terms. But we still have a problem: how do we get the relation *John-him* to hold? In an MGG-style phrase structure grammar it is necessary to invoke additional devices (e.g., indexing, plus a level of representation where indices are identified and interpreted) which allows the grammar to identify *John* and *him* as NPs which are assigned a unique referential index. Here we find one of the big payoffs of taking topological spaces seriously: we can define elements in a structure by means of coordinates, or addresses. If we do not pay attention to their phonological form (*John* vs. *him*), then we can simply assign each syntactic node an address that corresponds to the *interpretation*[10] of that node; the interpretation of neighbourhoods proceeds in a directly compositional manner. Let Δ stand for the address which points towards the interpretation of 'John' (*John* refers to the word, 'John' to the entity). Then we need to revise (18) as in (19):

19) a. [$_{\text{M}}$ Δ wished [L]]]
b. [$_{\text{L}}$ that Mary would go out with Δ]

If syntactic nodes are Gorn addresses (as in Sarkar & Joshi, 1997; see Gorn, 1967), then once L substitutes for L in M, we simply have *a single instance of the address* Δ *in two distinct syntactic contexts*. Note that substitution itself is allowed because the grammar is capable of identifying identical labels; what we call L is also an address, which corresponds to the neighbourhood of *go out*. The process of substitution and adjunction depend on the grammar being able to read addresses, and addresses make sense if they point *somewhere*: that somewhere is a location in the workspace. These addresses serve to identify corresponding expressions: if nodes on distinct elementary trees T and T' are assigned the same address, then a derived tree that contains T and T' will collapse those nodes into one. In a model like Saddy's, where syntactic objects are vectors in a topological space, then the vector that defines the syntactic objects is the same, only its context of occurrence changes. Identity is not defined in terms of indexes or external elements added to the representation, but simply in terms of where the addresses point towards. It is crucial to note that, if syntactic operations are required to yield tree-like structures in which an element cannot be dominated by more than a single node (the so-called Single Mother Condition; Sampson, 1975) then we are *required* to multiply the entities in the structural description: because syntactic context is defined in terms of dominance (and possibly also precedence; see McCawley, 1968), Δ dominated by K and Δ dominated by *with* in L already have a mother node and cannot have another; in MGG each needs to be a distinct object for purposes of syntactic operations. However, this multiplication of entities does not arise in the present proposal; topologically, the identification of Δ in K and Δ in L (which we will refer to as Δ_{K} and Δ_{L} for concreteness) amounts to having $d(\Delta_{\text{K}}, \Delta_{\text{L}}) = 0$. A proper formalisation of the notion of *workspace* allows us to simplify the mechanisms of the grammar, in this case dispensing with independent indexing mechanisms.

[10] This *interpretation* can be defined, as in Krivochen (2018b), as the translation of the NP into intensional logic (in the sense of Montague, 1973). For reasons of space and scope we will not go deeper into this here.

More interesting issues arise when we consider complex cases. For instance, the annotated examples in (20):

20) a. [Which picture of himself$_j$]$_i$ did John$_j$ say Mary likes __$_i$
b. John$_i$ wondered [which picture of himself$_{i/j}$]$_k$ Bill$_j$ saw __$_k$ (Chomsky, 1995: 205, ex. 36 a)

The approach to chain formation in Martin & Uriagereka (2014: 174, ff.) allows for a phrase marker to 'fold', creating a torus of sorts. In their view,

> *Chains are best represented as being comprised of several simultaneous derivational stages, so that in principle they exist in one or the other stage (say, the 'foot' or the 'head' of the chain, in these instances). To interpret a chain in a particular chain-state* ***ρ*** *is to collapse the chain in* ***ρ****.* (bolds in the original)

But, under present assumptions, there is no 'chain collapse' mechanism, because there are no chains *stricto sensu*: we have a set of local neighbourhoods of lexical heads, each of which is in turn a set of addresses and relations. In graph-theoretic terms, these relations can be identified with annotated arcs (Johnson & Postal, 1980; Postal, 2010), or by defining a total order relation between addresses which defines a unique walk through the graph (Krivochen, 2018b). In either case, when neighbourhoods intersect via *substitution* or *adjunction*, identical addresses in distinct neighbourhoods are treated as a single element, since it is just an instruction to retrieve an interpretation (or a portion of the lexical space; see e.g. Manzini & Savoia, 2011: Introduction for related discussion).

While a 'head-foot' gluing or folding (as proposed in Martin & Uriagereka, 2014) could work for (12), making ends meet and identifying Δ_K and Δ_L, the same mechanism cannot work in (20) because there are embedded syntactic terms which contain addresses that coincide with addresses in the matrix clause. Let us unpack this: let Greek letters stand for the addresses that correspond to specific syntactic terms. In this context, let us consider the structure of (20a), indicating only the variables that correspond to *John* (whose address will be Δ) and *Mary* (whose address will be Φ):

21) [Which picture of Δ] did Δ say Φ likes

But this cannot be right, because there is a 'gap' licensed by the transitive verb *like*. Regardless of how we represent filler-gap relations (see, e.g., Chomsky & Lasnik, 1977; Gazdar, 1982; Joshi & Kroch, 1985; Sag, 2010), there has to be a way to indicate that the term [which picture of Δ] satisfies the valency of *like*, but it also receives an operator interpretation:

22) For which *x*, *x* a picture of John, John said Mary likes *x*

The same procedure as in (17), *substitution* at the frontier of the tree, would not work. The reason is that we need the structural description not only to *fold*, but also to *self-intersect*: there is a Δ in the subject position of the matrix clause, and a Δ within the operator complex [which picture of Δ]. Furthermore, this operator complex also appears in two contexts, as evidenced informally in (22). How would this be solved in a transformational, combinatory-based syntax? By multiplying the nodes and incorporating a notion of *indexing* that takes care of identification whenever relevant:

23) [Which picture of himself$_j$]$_i$ did John$_j$ say Mary likes t_i

The derivation of (23) along classical generative lines requires, at least, the following:

- A set of operations (phrase structure rules, IM / EM, MERGE, etc.) to generate the string *John said Mary likes which picture of himself*
- A movement rule that displaces the syntactic term [which picture of himself] from its base position as the complement of *like* to the 'left periphery' (adjoined to the root), call it *Wh-movement* (Chomsky & Lasnik, 1977: 434)
- An indexing rule that keeps track of occurrences of syntactic terms. It needs to be able to assign the same index to *John* and *himself*, but also to *which picture of himself* and *t*.
- A rule that inserts the dummy auxiliary *do* to spell-out tense and agreement features

Interestingly, much of this complication emerges because interrogatives are assumed to derive from declaratives by means of application of further rules (in other words, declaratives are assumed to be derivationally more basic than interrogatives, requiring less rules to be generated). But what happens if we take interrogatives and declaratives as equally 'basic' structures, and only care about providing a map of dependencies between syntactic objects in terms of the distance between those objects, which are points in the syntactic workspace? In that case, we can summarise what we need to capture:

- *John* and *himself* denote the same sortal entity
- The syntactic object *Which picture of himself* occurs[11] in two syntactic contexts

We can express the above two points as follows:

24) $d(John, himself) = 0$

But we know that (24) can only be the case if $John = himself$ by (8b), which is equivalent to saying that *John* and *himself* have the same address in the workspace. Consequently, we do not need to incorporate any additional terminal node or indexing mechanism (or reconstruction procedure), provided that we have the dependency in (24) and the *identity property* holds for the space where the dependency in (24) holds (see (7b)). In a lexicalised grammar, *picture* would also head its own elementary tree, which means that we need to consider, on the one hand, the interpretation of the address corresponding to 'John'; on the other, the interpretation of the elementary tree where that address occurs. We can present the procedure as follows:

25) 1. Define the neighbourhood of *picture*; assign a (directly compositional) interpretation
2. Define the neighbourhood of *like*; this requires us to define the neighbourhood of *picture* as a proper subset in the neighbourhood of *like*
3. We have the interpretation of the neighbourhood of *picture*, use that as part of the input for the (directly compositional) interpretation of the neighbourhood of *like*

[11] The notion of 'occurrence' used in the context of Chomskyan Minimalism is far from clear. The terms 'occurrence', 'copies', 'repetitions' have been used in a transformational framework (Collins & Groat, 2018; Chomsky, 1995, 2019), but they correspond to intra-theoretical entities which depend on there being copying and chopping transformations (in the sense of Ross, 1967) and an indexing mechanism over elements in a Numeration and in the derivation. Many of the problems identified in Collins & Groat (2018); Collins & Stabler (2016); Chomsky (2019) arise because derivations operate over sets of lexical items and sets of sets of lexical items, with workspaces being defined over these (Collins & Stabler, 2016: Definition 10). Chomsky (2013: 40) uses *copies* and *repetitions* as types of *occurrences*, without defining any of these terms. In (2000: 115) he says that '*an occurrence of α in K to be the full context of α in K.*', where K is a syntactic object. Presumably, the 'context' refers to the mother-daughter nodes of α. There is a confusion, we think, between phrase markers as *sets* and diagrams of phrase markers (trees) (see also Postal, 2010: 7; McCawley, 1998: 47-48).
The approach adopted here is in this sense diametrically opposite to the Minimalist one: we are concerned with workspaces as topological spaces; these are indeed sets of points but we are concerned about the *distance* function between points in these spaces.

4. Define the neighbourhood of *say*; this contains an element whose address is the same as an address in the neighbourhood of *like*
5. Link the elementary trees by *substitution*
6. Unify all nodes with the same address

These need not be interpreted as sequential steps Von Neumann-style; because there is no separation between lexicon and syntax, the operations glossed in (25) disrupt the ultrametricity of the lexicon (which gives us distinguishability, organisation of the data, and absence of prior biases) and yields a metric space by bringing some points in that space closer together, defining the neighbourhood of lexical heads. Thus, steps (25.1), (25.2), and (25.4) proceed in parallel.

In the end, what we have is a *self-intersecting* oriented structure (i.e., a graph with *loops*) defined in metric space: we have nodes which correspond to (Intensional Logic translations of) basic expressions of the language and which are connected; these *connections* are represented by *edges* between those nodes which encode the *distance* that separates any given nodes within a space. Note that the process of metricisation we assume here only needs *one* space, because (unlike *Y-model* based architectures of the grammar) there is no separation between a *lexicon* component and a *syntactic* component: 'syntax' is something that 'happens to the (topology of the) lexicon', informally put (see also Stroik & Putnam, 2013 for a related perspective). Also, it is important to point out that the operations *do not extend* the space, if anything, the portion of the lexical space that we care about gets literally smaller as points are drawn closer together (consequently, nothing like the *Extension Condition* applies). The system sketched here is thus (at least partially) compatible with Chomsky's (2019) desiderata for a theory of operations in the syntactic workspace.

5. Conclusions

We can now summarise some aspects of the theoretical proposal as has been presented so far. We have proposed an explicit definition of *workspace*, and related the mathematical properties of the syntactic workspace to the kinds of operations and relations that can hold between terms in that space. The ground state of the syntactic workspace is defined by the dynamical frustration between high-dimensional and low-dimensional structures: semantics and linearisation requirements. This dynamical frustration defines an *ultrametric space* with a strong separation axiom (specifying *closed* neighbourhoods for all points). But in such a space it is not possible to define syntactic dependencies, which require a notion of variable *distance* in order to restrict the application of structure mapping rules (consider Relativised Minimality, Minimal Link, Attract Closest, etc.). The selection of a lexical array disrupts the ultrametricity of the initial state of the syntactic workspace, with major consequences: if in this ultrametric space distinct points have closed disjoint neighborhoods then disrupting the ultrametricity of a space by defining neighbourhoods and bringing points closer together also impacts on this characteristic: closed neighbourhoods become open ones. Syntactic operations create intersections between the neighborhoods of lexical heads; these are elementary trees in the LTAG sense and allow for non-monotonic derivations (for which we have provided empirical motivation from Spanish auxiliary chains). The resulting metric space displays topological distinguishability as well as variable distances between points (Willard, 2004; Munkres, 2000)[12], which is what we want for syntactic structures.

[12] Saddy (2018: 323) conceives of the neighbourhoods of lexical heads as *manifolds* (high-dimensional topological structures which are only locally Euclidean) in an ultrametric space. The disruption of the initial state ultrametricity is a consequence of syntactic operations, as here. Saddy explains the process as follows:

This paper started as an effort to explore the concept of *workspace*, which plays an important role in current MGG. In what pertains to the problem of making explicit the properties of spaces where syntactic operations take place (call them Merge, Copy, Transfer –from somewhere to somewhere else-; see Chomsky, 2019; Chomsky et al., 2019), we argued that syntactic derivations involve the metricisation of an initially ultrametric space; and that structure is built when the topology of that ultrametric space is disrupted such that points within that space are brought closer together. Conceiving of syntactic terms as addresses to points in a space allows us to dispense with inaudible additional structure (the multiplication of nodes in *chains*). Furthermore, we eliminate the distinction between a lexical space and a syntactic space by having 'syntax' be a set of local topological transformations of the lexical space. Problems related to the multiplication of workspaces in copying operations (External Merge as *copy from the lexicon to the syntactic workspace* and Internal Merge as *copy from and to the syntactic workspace*) thus do not arise.

6. References

Artin, Emil. 1967. *Algebraic numbers and algebraic functions*. New York: Gordon and Breach.

Bach, Emmon. 1964. *An Introduction to Transformational Grammars*. New York: Holt Rinehart & Winston.

Beim Graben, Peter & Sabrina Gerth. 2012. Geometric Representations for Minimalist Grammars. *Journal of Logic, Language, and Information* 21(4). 393-432.

Bever, Thomas G. 1975. Psychologically real grammar emerges because of its role in language acquisition. In Daniel P. Dato (ed.) *Developmental psycholinguistics: Theory and applications*. Georgetown University Round Table on Languages and Linguistics. 63‑75.

Bever, Thomas G. 2008. The canonical form constraint: language acquisition via a general theory of learning. In Susan M. Ervin-Tripp, Nancy Budwig, Jiansheng Guo et al. (eds.), *Cross-linguistic approaches to the psychology of language.* New York: Psychology Press. 475-492.

Binder, Philipe. 2008. Frustration in Complexity. *Science* 320. 322-323.

Boeckx, Cedric. 2008. *Aspects of the syntax of agreement.* London: Routledge.

Bošković, Željko. 2016. What is sent to spell-out is phases, not phasal complements. *Linguistica* 56, 25-56.

Bošković, Željko. 2020. On the Coordinate Structure Constraint, Across-The-Board movement, phases, and labelling. In Jeroen van Craenenbroeck, Cora Pots and Tanja Temmerman (eds.) *Recent Developments in Phase Theory*. Berlin: De Gruyter.

Bresnan, Joan. 1971. Sentence Stress and Syntactic Transformations. *Language* 47(2). 257–81.

The extension of the manifolds into metric space comes about due to the fact that the distance function between any point x in manifold X and any point y in manifold Y varies as the manifolds get closer and finally intersect. This is a state of affairs that is impossible in ultrametric space […], *for* $x \in X$, $y \in Y$, $d(x, y) = k$ *(some constant) in the ultrametric space* […]; *but* $k \to 0$ *as the space in which the manifolds exist is deformed. This is a process of metrization. Crucially, the metrization of the initial space is a consequence of the intersection of manifolds.*

Caracciolo, Sergio & Luigi Radicati. 1989. Entropy Ultrametric for Dynamical and Disordered Systems. *Journal de Physique* 50(19). 2919-2930.

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

Chomsky, Noam. 1956. Three models for the description of language. *IRE Transactions on Information Theory* 2(3). 113-124.

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

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

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

Chomsky, Noam. 2012. Foreword. In Angel Gallego (ed.) *Phases: Developing the Framework*. Berlin: Mouton de Gruyter. 1-7.

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

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

Chomsky, Noam. 2019. Some Puzzling Foundational Issues: The Reading Program. *Catalan Journal of Linguistics*. 263-285 (transcribed and annotated version of the lecture Generative Linguistics in the 21st Century: the evidence and the rhetoric. University of Reading. Available at https://www.facebook.com/theuniversityofreading/videos/1115026661937460/)

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

Chomsky, Noam & Howard Lasnik. 1977. Filters and Control. *Linguistic Inquiry*, 8(3). 425-504.

Cinque, Guglielmo. 2004. 'Restructuring' and Functional Structure. In Guglielmo Cinque (ed.), *Restructuring and Functional Heads. The Cartography of Syntactic Structures*, Volume 4. Oxford: OUP. 132–192.

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

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

Dovgoshey, Oleksiy & Evgeniy Petrov. 2019. Properties and Morphisms of Finite Ultrametric Spaces and Their Representing Trees. *p-Adic Numbers, Ultrametric Analysis and Applications* 11. 1–20.

Epstein, Samuel; Husatsugu Kitahara, & T. Daniel Seely. 2015. Labeling by Minimal Search: Implications for Successive-Cyclic A-Movement and the Conception of the Postulate "Phase". In *Explorations in Maximizing Syntactic Minimization*. London: Routledge. 201-221.

Feigenson, Lisa. 2011. Objects, Sets, and Ensembles. In Stanislas Dehaene and Elizabeth Brannon (eds.) *Space, Time and Number in the Brain: Searching for the Foundations of Mathematical Thought*. London: Elsevier. 13-22.

Fiengo, Robert. 1977. On Trace Theory. *Linguistic Inquiry*, 8(1). 35-61.

Frank, Robert. 2002. *Phrase Structure Composition and Syntactic Dependencies*. Cambridge, Mass.: MIT Press.

Frank, Robert. 2006. Phase theory and Tree Adjoining Grammar. *Lingua*, 116(2). 145–202.

Frank, Robert. 2013. Tree adjoining grammar. In den Dikken, Marcel (ed.) *The Cambridge Handbook of Generative Syntax*. Cambridge: CUP. 226-261.

Gajić, Ljiljana. 2001. On ultrametric space. *Novi Sad J. Math*. 31(2). 69-71

Gavryushkin, Alex & Alexei J. Drummond. 2016. The space of ultrametric phylogenetic trees. *Journal of Theoretical Biology* 403. 197-208

Gazdar, Gerald. 1982. Phrase Structure Grammar. In Pauline Jacobson & Geoffrey Pullum (eds.) The Nature of Syntactic Representation. Dordrecht: Reidel. 131-186.

Gorn, Saul. 1967. Handling the growth by definition of mechanical languages. *Proceedings of the April 18-20, 1967, spring joint computer conference*. New York: Association for Computing Machinery. 213–224.

Hazewinkel, Michiel (ed.) 2001. Topological space. *Encyclopaedia of Mathematics*. URL: http://www.encyclopediaofmath.org/index.php?title=Topological_space&oldid=40046

Hefferon, Jim. 2014. *Linear Algebra*. Available online at http://joshua.smcvt.edu/linearalgebra/book.pdf [Retrieved on 5/1/2015]

Hinzen, Wolfram. 2009. The successor function + LEX = Human Language. In Kleanthes K. Grohmann (ed.) *InterPhases: Phase-theoretic investigations of linguistic interfaces*. Oxford University Press. 25–47.

Hopcroft, John & Jeffrey Ullman. 1969. *Formal Languages and their relation to Automata*. London: Addison-Wesley.

Hughes, Bruce. 2003. Trees and ultrametric spaces: a categorical equivalence. *Advances in Mathematics* 189. 148–191.

Idsardi, William J. & Eric Raimy. 2013. Three types of linearization and the temporal aspects of speech. In Theresa Biberauer and Ian Roberts (eds.) *Challenges to linearization.* Berlin: Mouton de Gruyter. 31-56.

Jayseelan, K. A. 2017. Parallel Work Spaces in Syntax and the Inexistence of Internal Merge. In Gautam Sengupta, Shruti Sicar, Madhavi Gayathri Raman, and Rahul Balusu (eds) *Perspectives on the architecture and the acquisition of syntax*. Singapore: Springer. 115-136.

Johnson, David & Paul M. Postal. 1980. *Arc Pair Grammar*. Princeton, NJ.: Princeton University Press.

Johnson, Kyle. 2016. Toward a Multidominant Theory of Movement. Lectures presented at ACTL, University College, June 2016. Available at https://people.umass.edu/kbj/homepage/Content/Multi_Movement.pdf

Joshi, Aravind K. 1985. Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions? In David Dowty, Lauri Karttunen, and Arnold Zwicky (eds.) *Natural Language Parsing*. Cambridge, Mass.: CUP. 206-250.

Joshi, Aravind K. & Anthony Kroch. 1985. Linguistic significance of Tree Adjoining Grammar. Ms. University of Pennsylvania. ftp://babel.ling.upenn.edu/facpapers/tony_kroch/papers/relevance3.pdf

Joshi, Aravind K. & Takashi Yokomori. 1983. Parsing of tree adjoining grammars. Technical Report. Department of Computer and Information Science, University of Pennsylvania.

Kaplansky, Irving. 1977. *Set theory and metric spaces*. Boston: Allyn and Bacon.

Kato, Takaomi; Hiroki Narita, Hironobu Kasai, Mihoko Zushi and Naoki Fukui. 2016. On the primitive operations of syntax. In Koji Fujita & Cedric Boeckx (eds.) Advances in Biolinguistics. The human language faculty and its biological basis. London: Routledge. 29-45.

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

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

Kornai, Andras. 1985. Natural Languages and the Chomsky Hierarchy. In M. King (ed): *Proceedings of the 2nd European Conference of the Association for Computational Linguistics 1985*. 1-7.

Kracht, Marcus. 2001. Syntax in chains. *Linguistics and Philosophy* 24. 467–529.

Krasner, Marc. 1944. Nombres semi-réels et espaces ultramétriques. *Comptes-Rendus de l'Academie des Sciences*, 219. 433-435.

Krivochen, Diego Gabriel. 2015. On Phrase Structure Building and Labeling Algorithms: Towards a Non-Uniform Theory of Syntactic Structures. *The Linguistic Review* 32(3). 515-572.

Krivochen, Diego Gabriel. 2018a. *Aspects of emergent cyclicity in language and computation*. PhD thesis, University of Reading.

Krivochen, Diego Gabriel. 2018b. *Syntax as graph theory*. Ms. University of Reading. https://ling.auf.net/lingbuzz/003842

Krivochen, Diego Gabriel. 2020a. From *n*-grams to trees in Lindenmayer systems. Ms. https://www.researchgate.net/publication/341407852_From_n-grams_to_trees_in_Lindenmayer_systems_v2_additions_corrections_now_it_looks_more_like_a_paper

Krivochen, Diego Gabriel. 2020b. Some notes on the locally variable complexity of natural language strings. To appear.

Krivochen, Diego Gabriel & Luis García Fernández. 2019. On the position of subjects in Spanish periphrases: Subjecthood left and right. *Borealis: An international journal of Hispanic linguistics*, *8*(1), 1-33. https://doi.org/10.7557/1.8.1.4687

Krivochen, Diego Gabriel & Luis García Fernández. 2020. Variability in syntactic-semantic cycles: evidence from auxiliary chains. In Melvin González-Rivera & Sandro Sessarego (eds.) *Interface-Driven Phenomena in Spanish: Essays in honor of Javier Gutiérrez-Rexach*. London: Routledge. 145-168.

Krivochen, Diego Gabriel & Ludmila Lacková. 2020. Iconicity in syntax and the architecture of linguistic theory. *Studies in Language* 44(1). 95-131.

Krivochen, Diego Gabriel & Douglas Saddy. 2016. Structure mapping: Uniformity and mixture in displacement. *Czech and Slovak Linguistic Review* 2016/2. 16-45.

Kuroda, Sige-Yuki. 1976. A Topological Study of Phrase-Structure Languages. *Information and Control*, 30. 307-379.

Ladusaw, William. 1980. *Polarity sensitivity as inherent scope relations*. Bloomington, Indiana: University of Iowa, Indiana University Linguistics Club.

Lasnik, Howard. 2009. Shortest Move and Equidistance. Linguistics 611 class notes, University of Maryland.

Lasnik, Howard. 2011. What kind of computing device is the human language faculty? In Anna-Maria di Sciullo and Cedric Boeckx (eds.) *The biolinguistic enterprise: New perspectives on the evolution and nature of the human language faculty*. 354–365. Oxford University Press.

Manzini, Maria Rita & Leonardo Savoia. 2011. *Grammatical categories: Variation in Romance languages.* Cambridge: CUP.

Martin, Roger & Juan Uriagereka. 2008. Competence for Preferences. In Xabier Artiagoitiak and Joseba Lakarrak (eds/) *Gramatika Jaietan: Patxi Goenagaren Omenez (Grammar Festival: A Tribute to Patxi Goenaga)*. Bilbao: University of the Basque Country. 561–572.

Martin, Roger & Juan Uriagereka. 2014. Chains in Minimalism. In Peter Kosta, Steven L. Franks, Teodora Radeva-Bork and Lilia Schürcks (eds.), *Minimalism and Beyond: Radicalizing the interfaces*. Amsterdam: John Benjamins. 169-194.

Martín-Vide, Carlos, Gheorghe Păun & Arto Salomaa. 1998. A characterization of recursively enumerable languages by means of insertion grammars. *Theoretical Computer Science,* 205 (1–2). 195-205.

May, Robert. 1985. *Logical Form: Its Structure and Derivation*. Cambridge, Mass.: MIT Press.

McCawley, James D. 1968. Concerning the base component of a transformational grammar. *Foundations of Language* 4. 243-269.

McCawley, James D. 1998. *The Syntactic Phenomena of English*. 2 Vols. Chicago: University of Chicago Press.

Medeiros, David P. 2018. ULTRA: Universal Grammar as a Universal Parser. Invited paper for *Frontiers in Psychology: Language Sciences* 9. Research Topic Approaches to Language: Data, Theory, and Explanation. https://doi.org/10.3389/fpsyg.2018.00155

Medeiros, David P. & Massimo Piattelli-Palmarini. 2018. The Golden Phrase: Steps to the Physics of Language. In Angel Gallego & Roger Martin (eds.) *Language, Syntax, and the Natural Sciences*. Cambridge: CUP. 333–350.

Moessner, Roderich & Art Ramírez (2006). Geometrical frustration. *Physics Today* (February): 24–29.

Montague, Richard. 1973. The proper treatment of quantification in ordinary English. In Hintikka, Jaakko & Moravcsik, Julius & Suppes, Patrick (eds.), *Approaches to natural language*. Dordrecht: Reidel. 221–242.

Mori, Nobue. 2005. *A Syntactic Structure of Lexical Verbs.* PhD Thesis, University of Maryland.

Morin, Yves-Charles & Michael O'Malley. 1969. Multi-rooted vines in semantic representation. In Robert Binnick et al. (eds.) *Papers from the Fifth Regional Meeting of the Chicago Linguistic Society*. University of Chicago. 178-185.

Munkres, James R. 2000. *Topology*. 2nd Edition. Prentice Hall.

Murtagh, Fionn. 2004a. On Ultrametricity, Data Coding, and Computation. *Journal of Classification* 21. 167-184.

Murtagh, Fionn. 2004b. Thinking ultrametrically. In Banks D., McMorris F.R., Arabie P., Gaul W. (eds.) Classification, Clustering, and Data Mining Applications. Berlin: Springer. 3-14.

Müller, Gereon. 2004. Phrase Impenetrability and Wh-Intervention. In Arthur Stepanov, Gisbert Fanselow, & Ralf Vogel (eds.), *Minimality Effects in Syntax*. Berlin: Mouton/de Gruyter. 289-325.

Müller, Gereon. 2011. *Constraints on Displacement. A Phase-Based Approach*. Amsterdam: John Benjamins.

Nunes, Jairo. 2004. *Linearization of Chains and Sidewards Movement*. Cambridge, Mass.: MIT Press.

Ouali, Hamid. 2010. Computation efficiency and feature inheritance in crash-proof syntax. In Michael T. Putnam (ed.) *Esploring Crash-Proof Grammars*. Amsterdam: John Benjamins. 15-30.

Papo, David, Javier Buldú, Stefano Boccaletti & Edward Bullmore. 2014. Complex network theory and the brain. *Philosophical Transactions of the Royal Society*. B 369 (1653).

Postal, Paul M. 1998. *Three Investigations on Extraction*. Cambridge, Mass.: MIT Press.

Postal, Paul M. 2010. *Edge-Based Clausal Syntax*. Cambridge, Mass.: MIT Press.

Putnam, Michael & Rui Chaves (to appear) *Unbounded dependency constructions: Theoretical and experimental perspectives*. Oxford: Oxford University Press.

Putnam, Michael & Tom Stroik. 2010. Syntactic relations in Survive-minimalism. In Michael Putnam (ed.) *Exploring crash-proof grammars*. Amsterdam: John Benjamins, 143–166.

Rabinovich, Mikhail I., Pablo Varona, Irma Tristan, and Valentin Afraimovich. 2014. Chunking dynamics: heteroclinics in mind. *Frontiers in Computational Neuroscience* 8(22).

Rammal, R, G. Toulouse, and M. A. Virasoro. 1986. Ultrametricity for Physicists. *Reviews of Modern Physics* 58(3). 765-788.

Reid, Miles & Balász Szendröi. 2005. *Geometry and Topology.* Cambridge: CUP.

Rizzi, Luigi. 1990. *Relativized Minimality*. Cambridge, Mass.: MIT Press.

Rizzi, Luigi. 2016. Labeling, maximality and the head – phrase distinction. *The Linguistic Review* 33(1). 103-127.

Roberts, Mark. 2015. Ultrametric Distance in Syntax. *Prague Bulletin of Mathematical Linguistics* 103. 111-130.

Ross, John Robert. 1967. *Constraints on Variables in Syntax*. PhD dissertation, MIT.

Saddy, Douglas. 2018. Syntax and Uncertainty. In Angel Gallego & Roger Martin (eds.) *Language, Syntax, and the Natural Sciences*. Cambridge: CUP. 316-332.

Sag, Ivan. 2010. English filler-gap constructions. *Language*, 86(3). 486-545.

Sakai, Katsuro. 2013. *Geometric Aspects of General Topology*. Tokyo: Springer Japan.

Sampson, Geoffrey. 1975. The Single Mother Condition. *Journal of Linguistics* 11(1). 1-11.

Sarkar, Anoop & Aravind K. Joshi. 1997. Handling coordination in a tree adjoining grammar. Technical report, University of Pennsylvania. https://www2.cs.sfu.ca/~anoop/papers/pdf/tag-coordination.pdf

Scheer, Tobias. 2013. Why phonology is flat: the role of concatenation and linearity. *Language Sciences* 39. 54-74.

Schmerling, Susan F. 1982. How imperatives are special, and how they aren't. In Robinson Schneider, Kevin Tuite, and Robert Chametzky (eds.) *Papers from the parasession on nondeclaratives*. Chicago: Chicago Linguistic Society. 202–218.

Schmerling, Susan F. 2018. *Sound and Meaning: A Neo-Sapirian Theory of Language*. London: Brill.

Spencer, John, Sammy Perone, and Jeffrey Johnson (2009) Dynamic Field Theory and Embodied Cognitive Dynamics. In John Spencer et al. (eds.) *Toward a Unified Theory of Development Connectionism and Dynamic System Theory Re-Considered*. Oxford: OUP. Retrieved from Oxford Scholarship Online. DOI:10.1093/acprof:oso/9780195300598.003.0005

Stein, Daniel & Charles Newman. 2011. Spin glasses: Old and new complexity. *Complex Systems* 20: 115–126.

Stroik, Tom. 2009. *Locality in minimalist syntax*. Cambridge, MA: MIT Press.

Stroik, Tom & Michael T. Putnam. 2013. *The Structural Design of Language*. Oxford: OUP.

Sutherland, Wilson. 2009. *Introduction to metric and topological spaces*. [2nd Edition]. Oxford: OUP.

Tesnière, Lucien. 1959. *Éléments de syntaxe structural*. Paris: Klincksieck.

Uriagereka, Juan. 2002. Multiple Spell-Out. In Uriagereka, Juan, *Derivations: Exploring the Dynamics of Syntax*. London: Routledge. 45-65.

Uriagereka, Juan. 2008. *Syntactic Anchors: On Semantic Restructuring*. Cambridge: CUP.

Uriagereka, Juan. 2012. *Spell-Out and the Minimalist Program*. Oxford: OUP.

Uriagereka, Juan. 2014. Regarding the Third Factor: Arguments for a CLASH model. In Peter Kosta, Steven L. Franks, Teodora Radeva-Bork and Lilia Schürcks (eds.), *Minimalism and Beyond: Radicalizing the interfaces*. Amsterdam: John Benjamins. 363–391.

Uriagereka, Juan. 2018. The formal language hierarchy. In Uriagereka, Juan, *Biolinguistic Investigations and the Formal Language Hierarchy*. London: Routledge. 1-43.

Vender, Maria, Diego Gabriel Krivochen, Beth Phillips, Denis Delfitto & Douglas Saddy. 2019. Implicit learning, bilingualism and dyslexia: Insights from a study assessing AGL with a modified Simon Task. *Frontiers in Psychology: Language Sciences* 2019(10): 1647.

Vender, Maria, Diego Gabriel Krivochen, Arianna Compostella, Beth Phillips, Denis Delfitto & Douglas Saddy. 2020. Disentangling sequential from hierarchical learning in Artificial Grammar Learning: Evidence from a modified Simon Task. *PLOS ONE 15(5): e0232687*

Von Neumann, John. 1945. First Draft of a Report on the EDVAC. Ms. Moore School of Electrical Engineering, University of Pennsylvania.

Wells, Rulon S. 1947. Immediate Constituents. *Language* 23. 81-117.

Willard, Stephen. 2004. *General Topology*. New York: Dover.

Wilson, Robin. 1996. *Introduction to Graph Theory*. [4th edition]. London: Adison Wesley.

Zeijlstra, Hedde. 2020. Labeling, selection, and feature checking. In P. Smith, J. Mursell & K. Hartmann (eds.), *Agree to Agree: Agreement in the Minimalist Programme*. Berlin: Language Science Press. 137-174