DE GRUYTER MOUTON The Linguistic Review 2023; 40(2): 311–360

Diego Gabriel Krivochen*

Towards a theory of syntactic workspaces: neighbourhoods and distances in a lexicalised grammar

https://doi.org/10.1515/tlr-2023-2004
Published online April 20, 2023

Abstract: Recent work on structure building and mapping in Minimalist syntax makes explicit reference to *workspaces*; however, it is still an underexplored area. This paper is an attempt to (a) analyse the notion of 'workspace' as used in current Minimalist syntax and (b) provide a definition of 'syntactic workspace' that can help us capture interesting empirical phenomena. In doing this, we confront set-theoretic and graph-theoretic approaches to syntactic structure in terms of the operations that can affect syntactic objects and how their properties are related to the definition of workspace. We analyse the consequences of conceptualising 'syntax' as a set of operations that affect local regions of the workspace, defining directed graphs.

Keywords: graph theory; lexicalised grammar; Merge; topological space; workspace

1 Introduction: setting the scene

A recurrent topic in generative grammar has been the need to hold on to structure, either because it needs to be kept within probing memory for further operations or because an object has been subjected to a reordering rule, as in (1a–b) (where strikethrough indicates unpronounced but syntactically active material):

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~~]]?

I want to thank Harry van der Hulst for his interest in this paper and his editorial work, and an anonymous TLR reviewer for their detailed comments. I have greatly benefitted from discussing different aspects of the theory exposed here with Doug Saddy since at least 2016. Thanks are also due to: Elly van Gelderen and the Arizona State University Syntax reading group, participants in LingBaW2022 (especially Anna Bondaruk and Adam Przepiórkowski), Mostafa Barzegar and the syntax group at Allameh Tabatabae'i University, Dave Medeiros, Andrea Padovan, and Juan Uriagereka. The usual disclaimers apply.

***Corresponding author: Diego Gabriel Krivochen**, University of Oxford, Oxford, UK,
E-mail: diego.krivochen@ling-phil.ox.ac.uk. https://orcid.org/0000-0001-6145-1057

Open Access. © 2023 the author(s), published by De Gruyter. This work is licensed under the Creative Commons Attribution 4.0 International License.

The non-local dependencies in (1), one base-generated co-indexing and the other a product of movement, illustrate the necessity of maintaining structure active and syntactic objects accessible for non-subsequent derivational steps. Operations that refer to non-subsequent derivational steps, or to chunks of structure, require syntactic objects of variable complexity to be *stored* somewhere, where they can be accessed and retrieved. In connection to this, recent generative theory has appealed to the idea of *workspaces*, and it is playing an increasingly important role in the definition of Merge, Agree, and Minimal Search (e.g., Chomsky 2019, 2020a, 2020b, 2021; Chomsky et al. 2019; Collins and Stabler 2016; Epstein et al. 2020; Jayaseelan 2017; Kato et al. 2016; Kitahara 2021; Komachi et al. 2019; Milway 2022; Müller 2004: 298; Reuland 2011: 20). A careful look at the properties of workspaces and how they determine conditions on allowable structures and relations is therefore essential to advance Minimalist inquiry. In this paper we propose a topological theory of workspaces, and explore some of its architectural and empirical consequences.

Let us further illustrate the kind of problem that requires reference to derivational spaces. Assume that we have objects X and Y in a local relation, as represented in the unordered set (2):

2) Merge(X, Y) = {X, Y} (e.g., {die, John})

Now suppose that there is some relation *R* between X and Y: for instance, 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 operation. Let us now introduce a further element in the derivation, W, which requires a local relation with Y to satisfy some requirement (which one in particular is not relevant for the present argument). W is external to {X, Y}:

3) {W, {X, Y}} (e.g., {T, {die, John}})

What happens if a local configuration between W and Y is required which excludes X? A movement approach under set-theoretic assumptions can either (a) move Y to a higher position in the checking domain of W, 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 *Copy Theory of Movement*, Chomsky 2000; Nunes 2004; Uriagereka 2002 and much related work). Both options are diagrammed below:

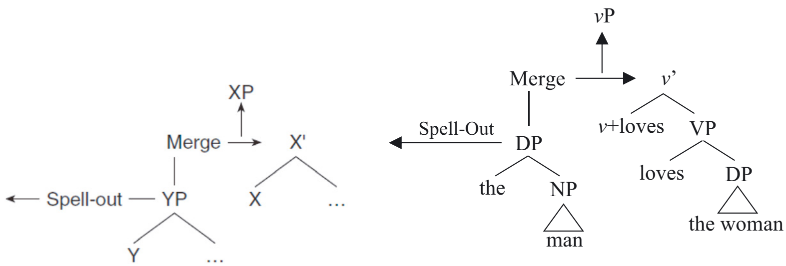
4) a. $\{Y_i, \{W, \{X, t_i\}\}\}$
   b. {Y, {W, {X, Y}}} (e.g., {John, {T, {die, John}}})

In both cases, the structure is extended by means of new symbols: in (4a), we add a *trace* of Y, an index to Y and *t*, and expand the structure; in (4b) we add a copy of Y and similarly expand the structure. In both cases there is a local relation between W and Y, as required, but at the cost of introducing additional terms (a trace or a copy).

Moreover, the very idea of copying requires not only an operation that takes Y and somehow produces another Y (Chomsky 2020a: 44), but also somewhere where the copy is stored and where it can be retrieved from: this ‘somewhere’ has usually been referred to as a *workspace*.

Consider, alternatively, the case of Merging two complex syntactic objects: when Merge applies to {XP, YP}, parallel workspaces where each of these objects is derived independently from the other are frequently invoked. Uriagereka (2002) applies such an analysis to the derivation of complex (i.e., ‘phrasal’) specifiers under Multiple Spell-Out: a phrasal specifier is a unit within which asymmetric c-command relations -and thus precedence- can be defined unambiguously (a ‘command unit’), and which in order to comply with the Linear Correspondence Axiom must be Spelled-Out separately from the derivational current to which it attaches (see also Jayaseelan 2017: 121, fn. 10, who needs parallel workspaces even for arguments, if phrasal):

5)



Nunes (2004) and Milway (2022) multiply workspaces in their analyses of adjunction, with Nunes (2004: 139) defining sidewards movement in parasitic gaps as movement ‘from a “subordinated” to a “subordinating” derivational space’. To summarise, an explicit workout of the concept of *workspace* seems unavoidable, yet there are few explicit characterisations of what *workspaces* are, how they interact with the generative procedure in Minimalist syntax, and, most importantly, how properties of operations and objects are related to (or depend on) properties of the workspace.

The paper is structured as follows: after a brief literature review in Sections 1.1 and 1.2, Section 2 defines topological spaces and explores how we can use topological characterisations of spaces to our advantage in syntactic theory. In addition to providing a general definition of topological space, we argue for a specific technical implementation of workspace-based syntax, through a graph-theoretic lexicalised grammar. Section 3 compares two kinds of topological spaces (ultrametric and metric) as models for the syntactic workspace, and argues that derivations may be conceptualised as mapping one to the other, locally: syntactic relations change the topological properties of the workspace. Some empirical consequences of this view are explored in Section 4.

1.1 Workspaces in the Minimalist literature: questions and problems

The generative procedure in Minimalist syntax takes the form of an operation of 'unordered binary set formation' (Chomsky 2019, 2020a: 22, 2021; Collins 2017; Epstein et al. 2015a, 2015b; Kitahara 2020, 2021): Merge(X, Y) = {X, Y}. Recently, Merge has been recast as MERGE, giving workspaces a central role. Chomsky et al. (2019: 236, 245) say:

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

> All syntactic objects in the lexicon and in the workspace WS are accessible to MERGE; there is no need for a SELECT operation […]. WS represents the stage of the derivation at any given point.

where MERGE is defined as follows (Chomsky 2020a: 34, 42, 2021: 20; Komachi et al. 2019: 275):

> MERGE(P, Q, WS) = [{P, Q}, X_1, …, X_n] = WS′, where if Y is a term of WS, it is a term of WS′.

A derivation, then, is a sequence of workspaces <WS_1, WS_2, …, WS_n> where WS_n is the result of MERGE(WS_{n-1}). In this context, an important question arises: what does the stage of the derivation at any given point include? Are objects in the lexicon part of WS?[1] In this respect, Chomsky (2021: 16) claims that

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

Chomsky here clearly separates WS from LEX: syntactic computation involves *two* distinct spaces, such that lexical elements are taken from LEX and combined in WS. This begs the question of how a derivation would begin (the problem of determining

---

1 The answer is *yes* in Kato et al. (2016: 35): "We assume that WS is the set consisting of SOs already constructed and LIs in the Lexicon, that is, WS = {Σ_1, …, Σ_n} ∪ Lexicon = {Σ_1, …, Σ_n, LI_1, …, LI_m}". Also Chomsky (2020a: 45): "for a given I-language, the set of workspaces—the set notice, not the least set—is the set containing the lexicon and containing MERGE (P, Q, WS) for any P, Q and WS that has already been generated" (our highlighting); cf. (2020a: 56). Chomsky (2020b) defines the workspace as "the set of objects that are available for further computation […] the workspace specifies the current state of computation"; in this sense, the workspace is analogous to Turing's (1936: 231) definition of 'configuration'. If External Merge involves access to the lexicon, then elements in the lexicon must be available for further computation (since in Chomsky et al.'s (2019) system there is no Select operation, and therefore no Lexical Array), thus belonging to the workspace.

the 'ground zero' for a derivation; Putnam and Stroik 2011): in the definition of MERGE(P, Q, WS) (e.g., Chomsky 2020a: 48; Kitahara 2021; Komachi et al. 2019: 275), how are P and Q selected, considering that Chomsky et al. (2019: 245) reject the existence of an operation SELECT (but see e.g. Chomsky 2021: 36)? The only thing that is specified is that P and Q are accessible in the workspace, which does not answer this particular question if LEX and WS are disjoint spaces. Accessibility also requires a special stipulation, in that either only a subset of LEX is 'accessible' (possibly through a lexical array or numeration, the selection of which remains mysterious) or all of LEX is. We will see that one of the advantages of our proposal is that it allows us to maintain the idea that LEX is always (locally) available without multiplying the spaces.

The absence of detailed discussion about the nature and properties of WS and its relation to LEX is also concerning because the properties of WS may impose conditions on the operations that apply and the relations that may hold: the very notions of local and non-local dependency need to be reconceptualised in terms of what is accessible at each derivational step.[2] For definitions, Chomsky et al. (2019: 236) refer the reader to Collins and Stabler (2016: 46), who say:

> **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. [highlighting ours]

In this context we can consider Chomsky's view alongside Collins and Stabler's, as they represent what can be found elsewhere in the literature (e.g., Collins 2017: 63; Milway 2022; Müller 2004: 298).

Several questions arise. If the workspace of a derivation is the structure already built, how does it help in formulating a theory of copies (vs. repetitions)? Do properties of the workspace play any role in defining locality relations? In sum: what exactly does proposing the existence of a workspace buy us in theoretical and empirical terms under current Minimalist assumptions?

1.2 Copies and workspaces

Let us briefly focus on copying. Suppose that WS is the lexical array + the syntactic object being built. This does not help in solving the issue of determining 'where

2 In Chomsky's recent works (e.g. 2020a, 2020c) phases are maintained as entities in the theory, defined in terms of designated 'endmarkers' for structural probing within sets: *v** and C. Crucially, in that approach locality is not defined in terms of properties of the workspace.

copies are stored temporarily' or, indeed, what copies are. Consider the following derivation (Chomsky 2020a: 48):

6) a. $\{b\}$
   b. $\{a, b\}$
   c. $\{b_1, \{a, b_2\}\}$

In (6), b_1 is a copy of b_2, internally Merged to the set $\{a, b\}$. As Gärtner (2022) observes, in order to avoid issues with the set-theoretic Axiom of Extensionality and represent the intended use of copies in syntax, (6c) needs to either be formalised as a multiset (a collection which allows for multiple instances of an element which would otherwise be counted only once, by Extensionality) or include diacritics (such that b_1 and b_2 are 'copies' in name only, not identical objects): if IM 'produced' two copies of *b* (the wording is Chomsky's 2020a: 44, Chomsky 2021: 16 proposes an operation FormCopy which -informally- annotates 'certain identical inscriptions' in a set with the relation *copy-of*; see also Chomsky 2001: 39), then (6c) would contain two distinct objects assigned to the indexed category *b*. Chomsky et al. (2019: 237) offer no set-theoretic basis for their claim that in an object like (6c) there is a single 'discontinuous' [sic] object *b*. Where are copies produced and stored? How is the object containing b_1 and b_2 interpreted? How many spaces do we need to make this system work?

As observed above, the Copy-based approach has some problems pertaining to the lack of explicitness about the specific mechanisms involved in copying syntactic objects and linking them such that in (6c) there would somehow be only one 'discontinuous' object and not two distinct ones related by a diacritic. Stroik and Putnam (2013: 20) formulate the issue clearly (cf. Jayaseelan 2017: 115–116):

> 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 computer file, one must reproduce the file somewhere in the computer (at least in temporary memory).

It is important to point out that the problem of specifying *where* operations take place arises for both External Merge (EM) and Internal Merge (IM). Stroik and Putnam (2013: 21) propose 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 (EM would be 'Copy from lexicon-to-phrase marker', and IM 'Copy from phrase marker-to-phrase marker'): this reworking of IM and EM makes it explicit that they 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- (cf. Stroik 2009). Current Minimalism adopts a different perspective. Chomsky (2020a: 23) contends that EM is more complex than IM in that it involves a relation between two distinct spaces, and possibly a further operation 'Select' such that only some elements of LEX are used in any given derivation (e.g., Chomsky 2000: 101, 2012: 3; but see Chomsky et al. 2019: 245).[3] This is a direct and (so far as we can see) unavoidable consequence of dissociating *lexicon* from *syntax* and these two from the sound and meaning systems. And it is one of the issues that a theory of workspaces should address.

In the sections that follow we make a concrete claim: the syntactic workspace will be defined as a *topological space*, and syntactic objects are either (i) points in the workspace or (ii) graphs whose nodes are points in the workspace. In a departure from set-theoretic syntax, we will argue that syntactic operations relate expressions in the workspace by delivering directed graphs: anticipating discussion from Section 2.1, Merge(X, Y) = {X, Y} will be replaced by Merge(X, Y) = *e*<X, Y> (a directed edge from X to Y, capitalising on the asymmetry of Merge in the original definitions), where X and Y are points in the workspace corresponding to basic expressions of the language indexed by a set of uniquely identifying addresses.

We are primarily concerned with two aspects in the analysis of syntactic dependencies: (i) the relevance of the notion of distance between any X and Y in the definition of syntactic dependencies, and (ii) the properties of the spaces where these dependencies are defined and how they constrain allowable relations: how we can make properties of structural descriptions fall out from properties of the workspace. 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, along Minimalist lines, that syntactic operations (in particular, Copy and Re-Merge, but also indexing and the determination of structural contexts for occurrences of syntactic objects) occur somewhere. We depart from current Minimalism in the characterisation of the operations themselves and the objects generated by these operations, and show that this departure results in simplifications of the theory of syntactic dependencies (and the *de facto* elimination of copies). In addition to providing an explicit definition of *workspace* and analysing the consequences of adopting it, the main innovation of our work consists of the proposal that computational operations affect the properties of WS itself.

---

3 Blümel (2017: 15) makes a curious argument against multidominance related to this point:

> [having EM deliver Multidominance] means that a verb like *loves* introduced into the derivation is simultaneously part of the lexicon and part of the derivation, i.e. it is "multiply dominated" in a structure of the current derivation and by the lexicon – clearly, an absurd result.

This argument relies on the strange idea (never actually put forth by anyone so far as we know) that dominance is a relation that holds between elements in the lexicon, plus the usual assumption that the lexicon and the workspace are disjoint. The conclusion that Blümel wants to avoid, and which requires the introduction of an asymmetry between IM and EM, is only reached under a view where the lexicon works like a phrase marker.

2 Basic properties of spaces

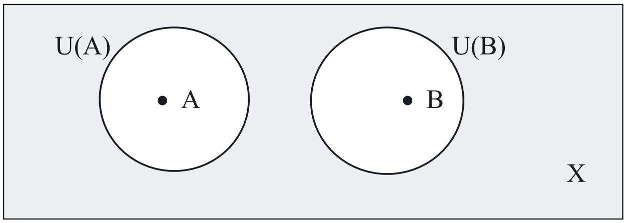
If we want the notion of workspace to work for us, theoretically and empirically, we need to provide an explicit characterisation of what a workspace is, what properties it has, and how it determines the kinds of operations that can apply in it and the properties of objects that are inputs and outputs of those operations. To fully understand what assuming workspaces in syntax commit us to, some definitions are in order: as anticipated above, we will explore the possibility that syntactic workspaces are topological spaces, primarily because this makes available to us a set of mathematical tools that we can use in our inquiry. This approach implies a departure from the literature reviewed in the previous section; we will argue that it is justified in that solves or avoids the problems and ambiguities identified above.

First, then, 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, so we must provide further details.

Let A and B be points in a topological space X. Then, we need to define the *neighbourhoods* of A and B, call them U(A) and U(B). If 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 and Szendröi 2005: 108). In plain words, the neighbourhood of A in X is a set properly containing A where one can move some amount without leaving the set: there exists a non-zero number ε such that one is in the neighbourhood of A if one's distance to A is equal to or smaller than ε. 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 set is *open* iff it is a neighbourhood for every one of its points, and *closed* otherwise (if there is a boundary set, then the union of an open set and its boundary set defines a closed set). 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.[4] We can provide a graphical representation of disjoint closed neighbourhoods for A and B in a topological space X:

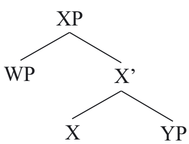
4 We can choose between different axioms which relate points and their neighbourhoods, and the specific axioms that we choose gives us a classification of spaces. Furthermore, we can define functions that take us from one kind of space to another: the possibility to transition from one kind of space to another will become relevant below (for technical details Willard 1970).

7)



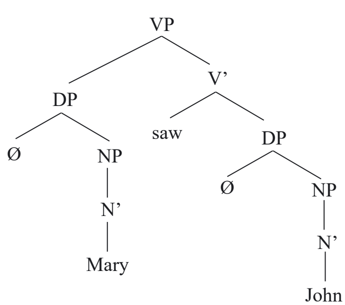
The notions of open and closed neighbourhoods should resonate with the syntactician. Consider a tree diagram like (8):

8)

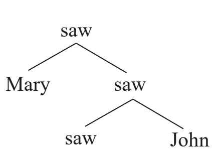


In X-bar syntax, X 'projects' an XP: a derived object whose internal composition, distribution, and semantic interpretation are determined by the properties of X, the 'head' of the phrase. As has been observed repeatedly in the literature there are problematic issues with the idea of 'intermediate' and 'maximal' projections that have motivated a departure from the XP-X′-X template used in X-bar theory and the adoption of so-called bare phrase structure (BPS) under Merge: essentially, trees without extrinsic labels (like NP, VP, etc.), bar levels, or unary-branching. Under BPS assumptions, (9a) becomes (9b) (e.g., Chomsky 1995a: 63–64, 1995b: 226–228):

9) a.



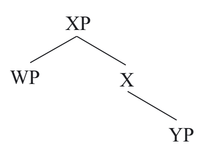
b.



Note that in (9b) we can use ‘VP’ as a shorthand for the set {*saw*, {*saw*, *John*}}, a ‘proxy’ for a set, as it were (as in Seely 2015: 124). It is, however, no longer a part of the syntactic representation strictly speaking (Chomsky 1995b: 247).

If we adopt the idea that WS is a topological space (and that phrase markers built in WS are topological graphs), we can ask what it means for XP to be the ‘projection’ of X from a different perspective which allows us to simplify the theoretical machinery and, as we will see in Section 2.2, has independent empirical advantages: using the definitions provided above, the ‘projection’ of a head H (specifically, its c-projection) can be characterised, in terms of relations in the workspace, as the neighbourhood of H. In other words: for X, Y, and W points in the workspace, the neighbourhood of X in (7) contains WP and YP (which in turn are the neighbourhoods of W and Y respectively). Crucially, the syntactic objects that belong to the neighbourhood of a head are selected by that head (the traditional notion of ‘subcategorisation’): we advance the claim that neighbourhoods are units of argument structure, in that selectional properties of this head must be satisfied (and are checked) within its neighbourhood (see also Frank 2002: 55, 2013: 240; Hale and Keyser 2005: 11; Kroch and Joshi 1985: 26). As we will see in Section 2.1, this means neighbourhoods are *lexicalised*. (10) is a first approximation to what such a simplification of phrase structure would deliver:

10)



Assuming for now that XP is a node in the structure independent of X, XP and YP are the graph-theoretic neighbours of X. In classical X-bar trees WP is further away from X than YP (counting number of nodes and edges in the walk between X and WP and X and YP), but still within the open neighbourhood of X. However, if labels are simply proxies for structures then what we call XP in BPS is a segment of the category X which acts as a boundary for the neighbourhood of X. A ‘head’, in this view, must be directly connected to all its dependant nodes: in BPS something along these lines is accomplished with multi-segment categories (as in (9b)), but in that framework these segments are category tokens, not category types. This initial simplification of phrase structure allows for the elimination of intermediate projections (on the syntactic inertness of which see e.g. Chomsky 1995a; Collins 2002; Hornstein and Nunes 2008; Seely 2015) without additional stipulations. Pursuing this idea will entail a departure from the set-theoretic foundations of current Minimalist syntax.

We can push this idea further. The core of it is that syntactic operations create local neighbourhoods in the workspace. We can now recast (8) in different terms. Let

WS be our workspace, with X, Y, and W points in WS. Then, the syntactic object we called XP (a proxy for a set of points and specific relations between them) is neither a segment nor a category, but the 'name' of a subset of WS (since every point in XP belongs to WS). YP is an *interior set* of XP, and WP is a *boundary set*: their union is the closure of the set XP. That is: W and its neighbourhood U(W), and Y and its neighbourhood U(Y), are subsets of XP (which, in turn, is just U(X)): we will see that Y and W differ in terms of how their neighbourhoods are derivationally connected to X's and whether they are accessible for probing from outside XP. This allows us to make the notion of 'projection of X' (and the definition of the building blocks of syntax) into a concept that, configurationally, does not need independent definitions other than 'point' and 'neighbourhood', which are part of the definition of topological workspace. This is a desirable consequence of the present view, as properties of local derivational units can now be defined in terms of WS.

2.1 Two substantive commitments

The approach to workspaces adopted here is compatible with different ways to define the atomic units of syntax (adopted by different theories): lexical items, features, roots and functional morphemes, basic expressions … A space is a set of points and a metric defined on them; there are no constraints on what 'points' correspond to delivered by the formalism: they are provided by a linguistic theory and the choice between the aforementioned options (and others), so far as we can see, is eminently empirical. However, a choice must be made to apply the formalism. We propose that points in the workspace are *basic expressions* of the language (Oehrle et al. 1988; Schmerling 2018).[5] These need not coincide with orthographical words: the lexicon contains multi-word basic expressions such as *heat up* or *would rather* (see Schmerling 1983 for an example of syntactic analysis making use of multi-word basic expressions in a categorial algebra). Basic expressions can be either *categorematic* or *syncategorematic*: categorematic expressions have semantic interpretations which vary (or may vary) across models. Ns, Vs, As, Advs and contentful Ps are

5 Schmerling (2018: 16) defines that

> The basic expressions of L are simply the expressions that are not derived by any operation – they can be thought of as comprising the language's lexicon. If L has a non-empty set of operations (as any natural language does), the outputs of those operations are derived expressions.

The set of basic expressions is indexed by a set of categories Cat = {N, V, A, P, …}. For convenience, we will use Cat_X where X is a variable over members of Cat. These categories can be seen as abbreviations of a more articulated system (see fn. 9).

categorematic. Expressions like copulas, expletive pronouns, infinitival *to*, or case-marking prepositions (e.g. Spanish DOM), have *constant* interpretations across models. These expressions do not have meanings of their own, but every expression in which they occur does have a meaning: following the logical tradition (MacFarlane 2017; see also Schmerling 2018 for a linguistic perspective), we will refer to them as *syncategorematic*.

Crucially, we argue that the theory of workspaces is a theory of the lexicon as much as of syntax: we contend that syntactic operations apply *to* (not *in*) the workspace, the initial state of which we call the 'lexicon' (more on this in Section 3). This is so because, in our view, the initial state of the workspace is an associative network of basic expressions:[6] a lexical space. These expressions enter relations in syntactic structures; we need to define a format for structural descriptions that allows for this to happen.

A crucial aspect of our proposal is that, in addition to a set of basic expressions, there is a set of addresses, such the set of addresses indexes the set of categorematic basic expressions. Let E be a basic categorematic expression of the language (see Schmerling 2018: 151, ff.), then ⦃E⦄ is the uniquely identifying address of E. Essentially, addresses allow us to uniquely identify expressions regardless of syntactic context, not unlike a direct access icon we may create in an operative system that uniquely identifies a file and always accesses it regardless of where the icon is located. In our analogy, the content of an address would be a file; here, the content of the address assigned to a basic expression is the semantic value of that expression (in the sense of Dowty et al. 1981; Heim and Kratzer 1998 and others): ⟦E⟧ is the semantic value of E. This means that when a node appears in a complex object, that node is assigned a unique identifier that is independent of the syntactic context in which that node appears. In this sense, our addresses differ from the Gorn addressing scheme used e.g. in Sarkar and Joshi (1997) (see Gorn 1967a, 1967b). Gorn addresses are assigned to nodes in a data structure in a way that depends on their context: if the root of a strictly binary-branching tree has address 0, its daughters will have addresses 1 and 2. The daughters of these two nodes will in turn be 1.1 and 1.2 for the daughters of 1, and 2.1, and 2.2 for the daughters of 2, and so on. Because we want to be able to locate a semantic value even when called in distinct (immediate dominance) contexts, our addresses are not dependent on a node's position in a structure.

There is a single address for each elementary semantic value and a single elementary semantic value for each address (recall that syncategorematic expressions have no semantic values, and are not assigned addresses). This entails that, in

6 In the present view, the theory of syntax is, explicitly, a theory of a system of expressions and relations. See e.g. Epstein (1999: 317–318), Postal (2010: 4–5), and Schmerling (2018: 27, ff.) for a variety of views on this vein.

the intended interpretation of [*b*, [*a*, *b*]] (leaving aside set-theoretic notation), we use the address ⦃b⦄ twice: there are two calls for the same semantic value in distinct syntactic contexts. Given a one-to-one relation between addresses and nodes (and between addresses and semantic values), the structure contains only one expression *b* with address ⦃*b*⦄ accessed twice. Importantly, there is no direct relation between morpho-phonological exponent and address: different basic expressions may receive the same PF exponent (an issue we will come back to in Section 4), and a single addressed expression may receive multiple PF exponents in distinct local domains. There is no need to multiply elements in the representation by having two *b*'s, no operation such as FormCopy or condition such as Stability required to achieve this (cf. Chomsky 2021: 17): these are artefacts of the set-theoretic foundations of Minimalism. We want to define structures where syntactic dependencies between addressed expressions can be adequately characterised. What formal device, if not sets, delivers the desired result?

This takes us to our second substantive commitment, also anticipated above: derived structures in the workspace are graphs, not sets. A graph is a set G = (V, E), where V is a set of vertices (or 'nodes') and E is a set of edges. $v \in$ V is a node, and $e \in$ E is an edge. An edge *e* joining nodes *a* and *b* is notated *e* = <*a*, *b*>, and *a* and *b* are said to be *adjacent* nodes. If edges in G are 'one-way roads' connecting a *head* and a *tail*, they are referred to as *arcs*, and G is a *directed* graph (or *digraph*) (van Steen 2010; Wilson 1996).[7] In this view, syntactic operations establish dependencies between nodes corresponding to basic expressions and define formal objects known as topological graphs (representations of graphs in a plane). For reasons that will become clear shortly, let us refer to the neighbourhood of a lexical predicate as an *elementary graph*.

The aforementioned 'syntactic contexts' can now be defined as the set of nodes that a node is directly connected to (either as the head or tail of an arc): in an irreducible elementary graph, a predicate will be directly connected to its arguments (for context, BPS-style trees are not irreducible, in that they contain multiple tokens of a category in the form of multi-segment categories). Viewing syntactic structures as digraphs in the workspace allows us to recover the asymmetry of structure building and selection: elementary graphs are minimal units of argument structure. Explicitly, we propose to replace

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

7 Exponents of graph-theoretic approaches to syntax include Zwicky and Isard (1963), Peters and Ritchie (1981), McCawley (1968, 1982a, 1982b), Huck (1984), Morin and O'Malley (1969), Kuroda (1976), Johnson and Postal (1980), Postal (2010), Gärtner (2002, 2014), Kural (2005), McKinney-Bock and Vergnaud (2014), and among others.

with

12) Merge(X, Y) = *e*<X, Y>

where X, Y are uniquely indexed points in the workspace, and either (i) X selects Y (e.g., $X \in Cat_V$, $Y \in Cat_N$)[8] or (ii) X modifies Y (e.g., $X \in Cat_{Adv}$, $Y \in Cat_V$).[9] Unlike (11), (12) defines an asymmetric order between X (the head of an arc) and Y (the tail of an arc), which allows us to capture (among other things) subcategorisation and argument structure: in an elementary graph, a functor will always dominate its arguments, and an *n*-place predicate will have outdegree *n*. Note, however, that the order imposed over nodes in (12) is not precedence: the output of Merge in the present view is ordered inasmuch as subcategorisation creates an asymmetry between selector and selectee (asymmetry that is represented in various ways in theoretical syntax, including feature descriptions, e.g. Ermolaeva and Kobele 2022; Stabler 2011; Stroik and Putnam 2013; Wurmbrand 2014; Zyman 2023).

Let us give an example. Suppose that we have an intransitive verb, such as *run*, and a nominal expression, say, *John*. As the result of Merge(*run, John*) we define the irreducible graph (13), an *arc* from a predicate to its argument (cf. McKinney-Bock and Vergnaud 2014):

13) ⦃run⦄ ⟶ ⦃John⦄

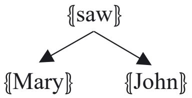
The arc *e*<run, John> is not equivalent to the unordered set {run, John}, which gives no information about selection or predicate-argument relations: in (13), order encodes selection (but not precedence). Consider now a transitive construction. If we go back to the BPS structure for *Mary saw John*, we see that both *Mary* and *John* are dominated by a node labelled *saw*. As Osborne et al. (2011) observe, the BPS representation (and even more so its radical label-less variants) takes Minimalism in the direction of Dependency Grammars, something we capitalise on. Under present assumptions, a diagram for the maximally simplified structure we can assign to the

8 See also Zyman (2023) for a graph-theoretic definition of Merge that also depends heavily on the satisfaction of selectional properties. Zyman's Merge, like classical phrase markers, does not defining a direct relation between X and Y (but rather between Y and a 'reflection' of X: X's reflection is distinct from X in a graph but everything that happens to X also happens to its 'reflection' in every derivational stage at which both are accessible, such as the satisfaction of a selectional feature).

9 These categories could be defined recursively, as in Categorial Grammar. In that case, *selection* always relates an expression of category X/Y with an expression of category Y (or X/(X/Y), depending on what the functor is), and *modification* always relates an expression of category X/Y with an expression of category (X/Y)/(X/Y) (i.e., modification does not change the category of its input). See Dowty (2003).

string *Mary saw John* would be (omitting functional structure for now) the directed graph in (14a), generated by means of the derivation in (14b):

14) a.



b. Merge(saw, John) = *e*<saw, John>
Merge(saw, Mary) = *e*<saw, Mary>, *e*<saw, John>

Crucially, because Merge applies to points in WS, and these are basic expressions, at any point only two basic expressions are Merged: by definition, arcs connect two nodes, which under present assumptions are necessarily basic expressions. There is, then, no derivational step that merges a basic expression with a complex structure (i.e., there is no ‘Merge(John, <saw, Mary>)’), since complex structures are graphs, not points in WS. As such, a graph cannot be the head or tail of an arc: only a basic expression can (which may be the root of a subgraph).

As in the BPS representation, in (14b) *John* and *Mary* are immediately dominated by a node indexed *saw*, but unlike in BPS there is only one such node in (14b): departing from set theory to embrace irreducible graphs defined in the workspace allows us to eliminate multi-segment categories as well as labels and projection (see Collins 2002; Seely 2015 for discussion).[10,11] But it is not just nodes that are ordered (by means of arcs): we contend that arcs themselves are ordered (Krivochen 2023b: Chapter 5). Assuming for concreteness standard bottom-up sequential structure building and composition (see also Larson 2014: 2, ff.), this order follows the derivational order of Merge, such that we can define the set of ordered arcs in (14a–b) thus:

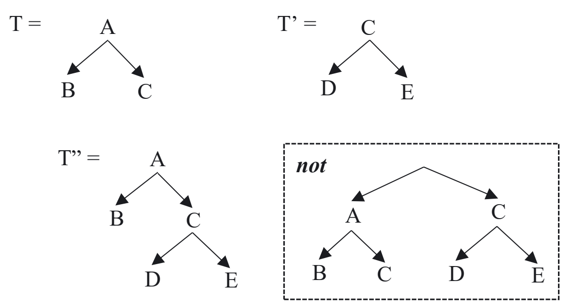
14) c. G = <*e*<saw, Mary>, *e*<saw, John>>

10 In this arc ordering, it is possible to encode further information, such as grammatical function (Krivochen 2023b: Chapter 5). Specifically, we can map the order between arcs to the grammatical function hierarchy (Bell 1983: 148; Keenan and Comrie 1977), of wide application in Relational Grammar and its developments (Arc Pair Grammar and Metagraph Grammar), as well as HPSG (Müller 2015: §2) and LFG (Dalrymple et al. 2019: Chapter 2): here, <*e*<saw, John>, *e*<saw, Mary>> would be mapped to <1/Subject, 2/Object>.

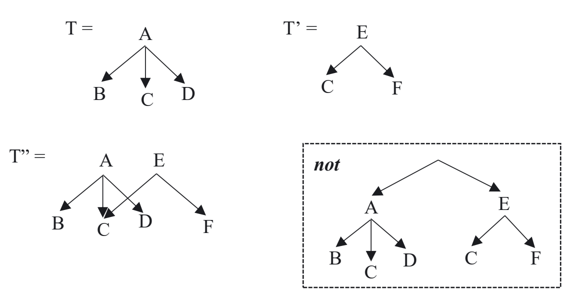
11 As observed in Krivochen (2023a, 2023b: Appendix), the BPS description (9b) can be reduced to (14) via *edge contraction*, an operation made available by the formalism. An edge is deleted, and the nodes it connected are contracted into a single node (Bondy and Murty 2008: 55). Here, edge contraction applies under index identity of head and tail: *e*<*saw, saw*> is therefore contracted (as these are two segments of the same category).

This format for syntactic structure allows us to compose local neighbourhoods straightforwardly: if nodes v_1 and v_2 on distinct neighbourhoods T and T′ ($v_1 \in$ T and $v_2 \in$ T′) are assigned the same address, then the composition of T and T’ (call it T″) will ‘collapse’ or ‘contract’ those nodes into one, call it v_3: $v_3 \in$ T″, iff $\{\!\!\{v_1\}\!\!\} = \{\!\!\{v_2\}\!\!\} = \{\!\!\{v_3\}\!\!\}$. The condition that must be met to contract nodes is identity of addresses. In other words: the composition of local neighbourhoods takes the form of graph-theoretic union. Let V_1 and E_1 be the set of nodes and edges in a graph G, and V_2 and E_2 the set of nodes and edges in G′. Then, G ∪ G′ = ($V_1 \cup V_2$, $E_1 \cup E_2$) (van Steen 2010: 29; Wilson 1996: 10). Node sharing in structure composition has a pedigree in TAGs (Han and Sarkar 2017; Kallemeyer 2004; Sarkar and Joshi 1997). Karttunen and Kay (1985) use the term ‘structure sharing’ for a process very similar to this one, but restricted to binary trees representing feature structures; structure sharing is also used extensively in HPSG (Müller 2015; Pollard and Sag 1994: 19) and LFG (Alsina 2008; Börjars et al. 2019: 103). (15) illustrates the results of the composition of local graphs as structure sharing, applying at the root (15a) or at non-root nodes (15b) (in what follows we will omit address notation, presupposing it: for any expression E, we will use E instead of $\{\!\!\{E\}\!\!\}$):

15) a.



b.



Anticipating discussion from Section 4, the syntactic interpretation of relations in a derived graph such as T″ in (15b) is straightforward: a single expression C establishes grammatical relations with predicates A and E, each defined within an elementary graph and preserved after composition. Note, incidentally, that in defining T and T′ there is no need to resort to parallel workspaces: each single-rooted structure defines a local domain within WS, with the root being the highest-order functor in that domain.[12] Under these assumptions about structure composition, non-monotonic applications of Merge do not differ substantially from monotonic applications in their formal properties: in the non-monotonic case, Merge applies to expressions which are themselves part of distinct neighbourhoods (as in (15b)). The opacity effects that seem to emerge from non-monotonicity (Uriagereka 2002, 2012) would be a consequence, in the present view, of the requirement that all syntactic dependencies be defined at the level of local neighbourhoods of lexical heads, as in lexicalised Tree Adjoining Grammars (LTAGs):

> Fundamental TAG Hypothesis (Frank 2002: 22)
>
> Every syntactic dependency is expressed locally within a single elementary tree

What LTAGs call ‘elementary trees’ are defined as the extended projections of lexical heads (Frank 2002, 2013), which is exactly what we intend to have in our ‘elementary graphs’: these are *minimal* in that they are strictly large enough to express all syntactic dependencies required by a lexical predicate which we will refer to as the *anchor* of the elementary graph (cf. Frank 2006: 150–152).[13] These requirements include subcategorisation and theta grids, both lexical properties of the anchor of the graph.

An elementary graph contains: (a) a single lexical predicate (the anchor of the graph), (b) all functional heads modifying that predicate, and (c) all non-clausal arguments subcategorised by that predicate (Krivochen 2023b; Krivochen and García Fernández 2019). These structures will be the domains where all syntactic relations are defined. Adopting the Fundamental TAG Hypothesis as a condition on the well-formedness of local structures means that (i) if the same address occurs in distinct elementary graphs, it must enter syntactic dependencies in each of these (and, as we shall see, a single addressed expression may even establish more than one relation with a predicate within an elementary graph), and (ii) syntactic relations, once

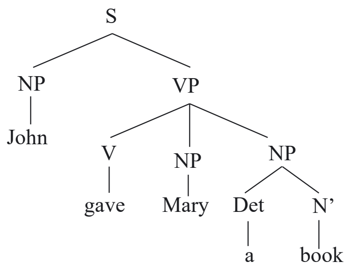
12 A related approach can be found in Uriagereka (2008: 257), Krivochen (2015a: 560; 2018), Mori (2005: 3), and especially Saddy (2018). In these works (in one way or another) each application of Merge increases the dimensionality of the structure, defining manifolds of increasing complexity in the workspace. Those approaches are, so far as we can see, compatible with our current characterisation of the workspace. In the present implementation, however, the dimensional complexity of the structure itself does not change, as all we are doing is connecting points.

13 The concept of ‘elementary graph’ could also be related to Chomsky and Lasnik’s (1995: 102) definition of ‘complete functional complex’ (CFC), ‘where a CFC is a projection containing all grammatical functions compatible with its head’. Chomsky and Lasnik’s ‘head’, in the present context, would be the anchor of an elementary graph.

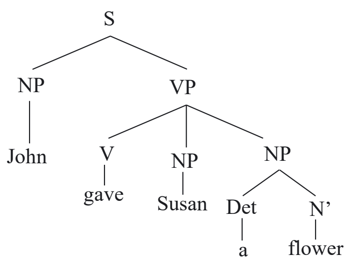
established, cannot be disrupted (but new relations can be built on top of existing ones).[14] In turn, conceiving of the composition of local neighbourhoods as graph union allows us to have derived graphs with multiple calls for a single address: we say that the node assigned that multiply called address is *shared* between elementary graphs. Note, however, that calling for an address more than once in a domain works both for elementary and derived graphs. Distinct configurations subject to distinct conditions arise in each case, as we will see in Section 4.

Let us give an example of the system at work, using traditional X-bar trees: the derivation of *gapping* in TAGs. The addressing system, plus composition at the non-root (TAGs explicitly do not abide by the Extension Condition; see Frank and Hunter 2021), allows Sarkar and Joshi (1997) to compose the trees in (16a, b) with the coordination schema in (16c) (where X stands for any category label) to output the derived structure (17):

16) a.



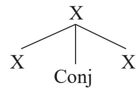
b.

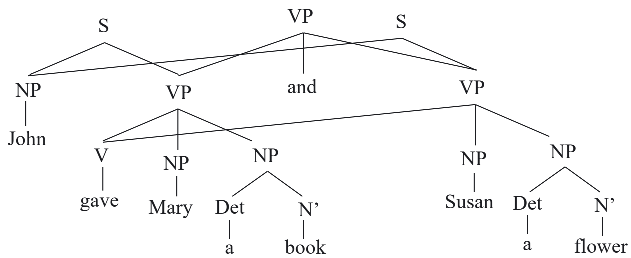


14 Kroch (2001) makes this requirement explicit in TAGs:

> we never want to allow derivations under which thematic roles, once established, are altered by further adjunctions, and we will block such derivations by, in every tree, placing a particular local constraint on every node that is assigned a thematic role by a governor (Kroch 2001: 11).

To the extent that subcategorisation and theta marking are lexical requirements, the idea that lexically governed relations are not disrupted throughout the derivation can be related to Lasnik and Uriagereka's (2005: 53, 112) *Conservation Laws*.

c. 

17) 

As Chomsky, Collins and Groat, and others have emphasised, (17) does not correspond to a legitimate set-theoretic object, but it is a perfectly well-formed graph-theoretic one, and -as we will highlight in Section 4- structure sharing is unavoidable given our addressing system. The idea that points in the workspace are indexed by a set of uniquely identifying addresses also has important consequences for the analysis of long-distance dependencies and binding.

2.2 Some technical consequences

The definition of the 'projection' of a head as the neighbourhood of that head defined in the workspace allows us to clarify several issues: in particular, we can shed some light on the concept of *edge* (of a syntactic domain). In an otherwise impenetrable domain (for example, a phase), the edge is a structural position that is accessible for operations triggered by a head external to that domain. What exactly constitutes the edge of a domain and is therefore (i) accessible for external operations and (ii) excluded from Transfer has been a problematic issue. In this context, it is worth asking whether there are properties of the workspace which can deliver locality and accessibility, as well as a natural definition of distance, or whether these must be defined and characterised in different terms. Given that all syntactic dependencies must be defined within elementary graphs and that these are lexically anchored, it is lexicalisation that delivers not only a workable concept of locality, but also determines the size of grammatical building blocks. *Accessibility*, although related to *lexicalisation*, is a different matter, as is *distance*.

In what pertains to accessibility, this is another point where the definition of local syntactic domains in terms of neighbourhoods of heads, as defined above, pays out theoretically and empirically. Recall that the neighbourhood of X may be open or closed. In this context, we can propose the following condition: if WP is accessible from outside XP, and WP is a syntactic object dependent of X, then we can define the neighbourhood of X, U(X), to be open and WP to be the edge of U(X). Crucially, it is possible for interior points of a set to be limit points as well: these are what phase theory refers to as 'edges'. Complements, on the other hand, would be (neighbourhoods of) interior points that are not limit points (of the neighbourhood of X in WS).

Accessibility is not defined in Chomsky (2019, 2020a, 2020b) and related works beyond the Phase Impenetrability Condition (other than the claim that 'if something was accessible to MERGE in WS it has to still be accessible in WS", Chomsky 2020a: 56), 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 syntactic object A being contained in an open neighbourhood if it is to be targeted by an operation triggered by a distinct syntactic object B. Closed neighbourhoods prevent accessibility.[15]

3 Metric spaces and distances in syntax

We defined what a space is in a very general way: a set of points and a set of neighbourhoods, plus a set of axioms allowing specific relations between these. We further defended that syntactic workspaces are topological spaces. Now we need to examine some properties of spaces in terms of the axioms that define the possible relations between these points. This is important because different sets of axioms define different spaces, and since the core idea of this paper is that the properties of the workspace determine the kinds of syntactic relations that can be defined between expressions, 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 (Sutherland 2009: 39; Willard 1970: 16). Here we differentiate between two such spaces:

---

15 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, we may 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 syntactic objects with *closed* neighbourhoods are those that cannot be embedded (and thus not even their root can be targeted by syntactic rules): imperatives (Schmerling 1982), vocatives, and interjections (Chomsky 2008: 139).

18)

For x, y, z points in a topological space S, S is ***metric*** iff:	For x, y, z points in a topological space S, S is ***ultrametric*** iff:
a. $d(x, y) > 0$ if $x \neq y$ (*positive property*)	a. $d(x, y) > 0$ if $x \neq y$ (*positive property*)
b. $d(x, y) = 0$ *iff* $x = y$ (*identity property*)	b. $d(x, y) = 0$ *iff* $x = y$ (*identity property*)
c. $d(x, y) = d(y, x)$ (*symmetric property*)	c. $d(x, y) = d(y, x)$ (*symmetric property*)
d. $\boldsymbol{d(x, z) \leq d(x, y) + d(y, z)}$ **(*triangle inequality*)**	**d.** $\boldsymbol{d(x, z) \leq \max\{d(x, y), d(y, z)\}}$ **(*ultrametric inequality*)**

Above we suggested 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 (producing a set of ordered pairs of members of X); d will be called a *metric* iff properties (a–c) plus the triangle inequality hold. The triangle inequality is a crucial property: it determines that distances in metric spaces sum: informally, if x is m units away from y and y is n units away from z in a segment defined in X, then x is $m + n$ away from z. This is an essential property of metric spaces, because it allows us to formulate the notion of *closeness* in comparative terms, such that x is closer to y than z if $d(x, y) < d(y, z)$ (i.e., if $m < n$). In our system, the identity property is intimately related to the addressing system: for every E, E′, $d(\mathrm{E}, \mathrm{E}') = 0$ iff ⦃E⦄ = ⦃E′⦄. This condition underpins the idea that the composition of neighbourhoods, implemented as graph union, delivers structure sharing and in fact enforces it: if an address is called in two or more elementary graphs that undergo composition, the derived graph will contain a single node with that address, which is always at 0 distance from itself. Two properties of metric spaces are crucial for our purposes: first, we can keep distinct points (i.e., expressions with distinct addresses) at non-zero distances from one another.[16] Second, the distance function can take arbitrarily many values, and distances sum.

The fact that distances between points vary and sum distinguishes metric from *ultrametric* spaces. Gajić (2001: 96) provides the following definition (see also Sutherland 2009: 41):

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

16 This requirement is enough to discard *pseudometric* spaces as formalisations for syntactic WS, since in a pseudometric space $d(x, y)$ can be 0 even if $x \neq y$.

Since distances between points are either 0 or 1, ultrametric spaces preserve topological distinguishability but all distinct points are *equidistant*, at distance 1 from each other. One of the main things we want a formalisation of workspaces to do for us is capture the idea that two distinct points X and Y in WS can be arbitrarily near or far apart (however distance is calculated), but never have 0 distance. X and Y can only have a 0 distance iff X = Y; this is crucial for our purposes of defining a notion of workspace that is syntactically usable. We want a space that preserves distinguishability between expressions with distinct addresses and in which variable distances can be defined unambiguously: locality conditions in the establishment of dependencies are based on the idea that some syntactic objects are closer to each other than to others.

One important, yet implicit, assumption in the definition of MERGE in recent work is that the mapping between WS and WS′ does not modify the properties of WS: it just expands the set of accessible elements by one (Chomsky 2021; Kitahara 2021). We depart from this assumption, arguing instead that establishing syntactic relations between points in WS in fact modifies the topological properties of WS, but never its size. A derivation is a dynamical process whereby the topology of WS is disrupted, and a number of locally stable structures are built. Building on the classical definition of a derivation as a finite sequence of steps, and borrowing terminology from dynamical systems theory, let us refer to the initial state of WS as its *ground state* and the final state of WS with respect to a derivation as a *stable state* (possibly one of many, in which case we speak of *multistability*; Kelso 2012). The ground state of WS, in the present view, is what we usually refer to as the Lexicon: the set of basic expressions of the language, indexed by a set of addresses. A local description of a stable state of WS characterises a syntactic output (a structural description).

Let us focus on the initial state of WS. Before the selection of a 'lexical array' to derive a specific sentence, we contend, all expressions are equally accessible and no two expressions are more closely related than any other two.[17] Distinct expressions

17 The selection of expressions is so far as we can see an unavoidable step (but see Chomsky et al. 2019), and one that requires external influence: a mechanism that defines an arc $<x, y>$ given expressions *x* and *y* is of little use in syntactic analysis if we do not know what expressions are to be used. However, what expressions are selected is a consequence not of the definition of workspace or its dynamics, but of the intention that a user of the system may have (in connection to the expression of a conceptual structure, for instance, along the lines of Uriagereka's 2008 'D-Structure'). Hilbert's epsilon operator is not helpful in that it operates non-deterministically when more than one term satisfies a formula (cf. Chomsky 2020a: 50), and this would result in numerous unusable choices. Similarly, whereas linguistic experience may reinforce certain connections in the ground state of the workspace (just like exposure to sensoria creates associations in other cognitive systems), that is not a property of the formal system we are describing, but a consequence of implementing (or 'embodying') that system.

are distinguishable in the lexicon, as the addressing system is independent from the topological properties of WS. However, as far as distances are concerned, all distinct expressions are equidistant and equally likely to be selected.[18] What we have just described is precisely an ultrametric space. We could justify the choice of ultrametricity as the topological characterisation of the initial state of WS in terms of (i) preserving distinguishability between distinct basic expressions, and (ii) not imposing a variable metric over these expression in the absence of syntactic relations. Both, arguably, reasonable assumptions about the lexicon. However, if aiming for explanatory adequacy, we can also ask what conditions imposed over the linguistic system would deliver ultrametricity as the initial state of WS. We will briefly address this question in the following section.

3.1 Frustrated language

The idea that derivations are dynamical processes, with ground states and (meta or multi-)stable states, is intimately related to recent work which has identified a *dynamical frustration* at the core of language, building on old insights. As defined in Binder (2008), a dynamical frustration emerges when a complex dynamical system is subject to mutually incompatible requirements, which gives rise to a form of dynamical instability. For example, a complex system may display local forces (e.g., dissipative) which counteract global (e.g., cooperative) forces; this is a *scale frustration* (Figure 1):

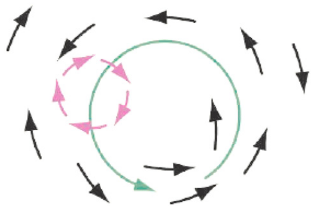


Figure 1: Scale frustration (from Binder 2008: 322).

18 This does not entail, evidently, that any choice may lead to a *convergent* derivation. An issue that immediately arises is how selection proceeds throughout the derivation (once the 'seed' of a local derivation, necessarily a lexical predicate, has been selected). Having Merge satisfy predicates' selectional properties (subcategorisation and theta marking) seems like a reasonable way to do this (see also fn. 17). In contrast, it is hard to see how 'free', 'blind' Merge can be implemented in an empirically fruitful and computationally feasible manner (indeed, Minimalist Grammars do not use free blind Merge).

The concept of dynamical frustration appeared in physics in the context of the study of spin glasses (Stein and Newman 2011). These are magnetic systems in the form of lattices in which electrons are subject to a pairwise antialignment constraint such that no two neighbouring electrons can have the same spin (up or down), which makes the system locally frustrated: there is no stable mode that satisfies the antialignment constraint for all electrons. This is a *geometrical frustration* (Figure 2):

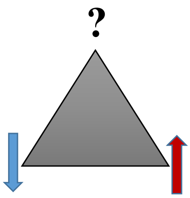


Figure 2: Geometrical frustration.

As Rammal et al. (1986: 771, ff.) argue, 'The crucial ingredients in these models [of spin glass] are disorder [entropy] and frustration.' Since the system cannot achieve a stable state globally, multiple locally optimal solutions appear, and the system oscillates between metastable states.[19] In addition to spin glass, the concept of dynamical frustration has also been fruitfully applied to the study of self-organising biological systems (Yang 2011), neural assemblies (Gollo and Breakspear 2014), protein structure (Nerukh 2009), and computability (Binder 2008).

In linguistic terms, the idea put forth in Uriagereka (2012, 2014) and Saddy (2018) and related works (and which can be traced back to Tesnière 1959: 21) is that morphophonology and semantics impose mutually incompatible requirements over syntactic representations: 'flatness' and low-dimensionality on the PF side (e.g. Idsardi and Raimy 2013; Scheer 2013) and multidimensionality and higher-level relations in the LF side (e.g. Hinzen 2009: 31; also the references on fn. 12). Uriagereka's (2012, 2014) discussion of dynamical frustration in language (which he considers

19 In dynamical systems theory, a metastable state is a state other than the minimum energy state which is easy to destabilise. The metricisation of the workspace must be metastable because the structure so delivered is transient. Saddy (2018), under the assumption that syntactic operations create structures of increasing dimensional complexity (see fn. 12), argues that metricisation is a consequence of the intersection of these structures in the workspace, structures which may then reach critical dimensionality and destabilise along the dimension where forces begin to fade (Carr 2006). The result is a lower-dimensional manifold that expresses the core properties of its high-dimensional version, and which can enter further computations before reaching critical dimensionality again. After dimensional compression, the system goes back to its minimum energy state, which is idealised here as an ultrametric space.

'third factor') refers to as a *hiatus*, a 'discontinuity' (a clash) between bottom-up semantic substance and left-to-right phonological organisation in a revised Minimalist architecture (Figure 3):[20]

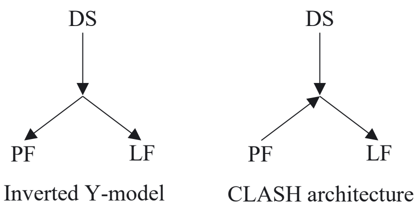


Figure 3: Inverted Y-model versus CLASH architecture (taken from Uriagereka 2014: 367).

Given the conflicting output conditions that sound and meaning systems impose over the space where syntactic structure is defined, we argue that instead of settling to equilibrium (Spencer et al. 2009: 109), the system is in permanent oscillation between metastable states (Binder 2008: 322). In the present context, this oscillation takes place between high-dimensional versus low-dimensional computation, and global versus local requirements.

If, following Saddy and Uriagereka, we propose that core structural properties of language are the result of a dynamical frustration, then the kind of topology delivered by dynamical frustration should be an adequate characterisation of the initial state of the syntactic workspace. Hence, we can tackle the issue 'from the other side' and ask: what kind of spaces do dynamical frustrations characterise? Murtagh (2004: 168) puts it in the following terms:

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

If the conditions under which syntax operates are defined by dynamical frustration, then language may well be one such case, from the perspective of both computation and its neurophysiological underpinnings. However, the proposal in this paper is that the topological properties of the workspace do not remain constant throughout the derivation: an adequate description of the syntactic workspace takes the form of the characterisation of a dynamical system, with structural descriptions being 'snapshots' of (meta)stable states of that system. Ultrametric spaces maintain distinguishability, but do not give us variable distances between distinct points. That is a desirable situation for the lexicon, but we need to modify the topology of WS to obtain usable linguistic outputs. This modification must bring

20 Roughly, DS is 'an array of conceptual structures' (2014: 367), PF is a 'designated class of speech objects', and LF is 'a designated class of chains' (2008: 6).

some expressions closer together, defining neighbourhoods of lexical predicates. At this point, we may summarise the previous discussion thus: a dynamical frustration delivers an ultrametric space, the initial state of the syntactic workspace (Saddy 2018: 317). The topological properties of this space are locally disrupted by means of syntactic operations whereby points (basic expressions) enter structural relations. These structural relations are defined in local domains, *elementary graphs*.

What does a (meta)stable state of WS look like? As observed above, much work in generative grammar suggests that an adequate characterisation of syntactic structures requires being able to sum distances and determine, given syntactic objects X, Y, Z in a structural description, whether X is closer to Y than Z: as mentioned before, conditions like Relativised Minimality (Rizzi 2016), Minimal Link/Shortest Move, Attract Closest (Chomsky 1995b) and Minimal Search (Chomsky 2004 et seq.) among others are, crucially, based on the idea that terms in a structural description can be at variable distances from one another, with some objects being closer than others. In addition to variable distances, we must evidently keep the distinguishability of distinct (indexed) expressions. In this context, *metric* spaces become the natural option as a characterisation of WS once structure has been built (see also Martin et al. 2019: their Hilbert space is a vector space endowed with a metric; Roberts 2015, however, proposes an ultrametric analysis of X-bar trees, which entails an ultrametric syntax). Structural descriptions (here, digraphs) are characterisations of local regions of a metricised WS.

Once syntactic relations are defined via Merge(X, Y) = *e*<X, Y> (and only then) we can define variable distances between nodes in a local structure. This is how elementary graphs are built, and how WS gets metricised:

In Figure 4, (a) is a crude illustration of an ultrametric space: it corresponds to the lexicon, an associative (but unstructured) network. No syntactic relations yet exist between basic expressions, which are all equidistant. In (b), we have selected some expressions, one of which is a lexical predicate with specific subcategorisation properties: it takes two arguments, and assigns a thematic role to each as part of structure building (cf. Chomsky 2021: 18; also e.g. Stroik and Putnam 2013: 104). As in analyses such as Stabler (2011), Wurmbrand (2014), Zyman (2023), and Krivochen (2023a) (among others), Merge satisfies these selectional properties as it establishes syntactic relations between *love*, *Mary*, and *books*: we define a digraph G = {E, V} in WS, where V = {*love, Mary, books*} (basic expressions) and E = *e*<love, Mary>, *e*<love, books> (arcs from the predicate to its arguments). The sequentiality of Merge delivers (19) (omitting functional structure):

a. b.

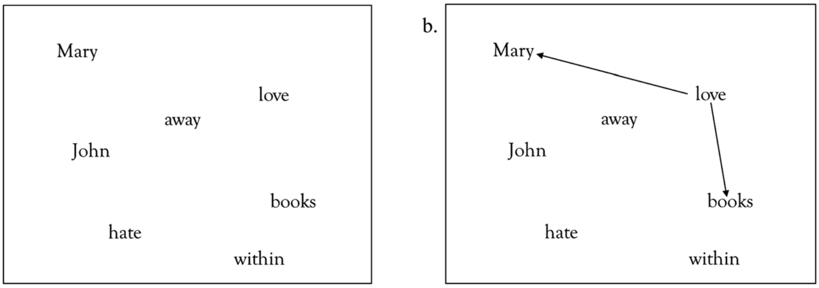


Figure 4: Graphs in WS.

19) a. Merge(love, books) = *e*<love, books>
    b. Merge(love, Mary) = <*e*<love, Mary>, *e*<love, books>>

Now, *love* is closer to *Mary* and *books* than to any other point: this disrupts the ultrametricity of the lexicon and delivers a graph in metric space, which is a structural description of the string *Mary loves books*. Figure 4b diagrams a space that has been locally metricised. Once we have that local metricisation, other linguistic conditions follow. For instance, Saddy (2018) argues that given the ultrametric inequality it is not possible to define linguistically relevant notions like 'edge' in ultrametric trees: the head is equidistant to every node within the tree and what we would call (in X-bar theoretic terms) 'specifier' and 'complement' are also equidistant. However, if the space is metricised, we can define (open) neighbourhoods of lexical heads, which constitute the building blocks of syntax in a lexicalised grammar. As in Section 2, we may distinguish *interior* and *boundary* points (and their neighbourhoods) once the space has been metricised, with familiar consequences for the definition of syntactic dependencies and structure composition: for example, Substitution and Adjunction in TAGs are defined in terms of properties of their target (a node in the frontier of an elementary tree or an internal node, respectively; Frank 2002; Kallemeyer 2004).

It is important to remember that in our perspective there are not *two spaces*, lexicon and syntax: syntactic operations apply to the lexical space (see also Stroik and Putnam 2013: 54). In other words: establishing syntactic relations between basic expressions modifies the topology of the lexicon, delivering a set of directed graphs in a metric space. Merging X and Y entails (i) bringing X and Y closer together than they were and (ii) defining an asymmetric relation between them (in McKinney-Bock and Vergnaud 2014, for instance, these relations can be either Selection or Checking, with which they annotate arcs). Now the metric between any two points in the workspace cannot be just two-valued (0 or 1): syntactic composition of X and Y ($X \neq Y$) puts X closer to Y than to any other point. If Z is merged to X, X will be closer to Y than

Z will be, but still Z will be closer to Y than a point in the workspace that is not part of this local neighbourhood. Properties of the space where a structure is defined constrain the relations that can hold between expressions in that structure and the operations that can apply to it.

In Section 4 we address the issue of how local structures are put together, since so far we have defined a set of elementary structures (the neighbourhoods of lexical predicates) and hinted at the existence of composition operations (based on graph union/structure sharing) but we have not operationalised them. The next section spells out some consequences of our approach for the theories of locality and binding in derived structures.

4 Doing syntax to the workspace

We have a characterisation of the workspace to which syntactic operations apply and, in that space, we have the expressions that appear in structural descriptions defined as points within said space. The initial state of WS is ultrametric: an associative network in which all basic expressions are equidistant. Through the application of syntactic operations, which generate directed graphs with uniquely indexed, categorematic basic expressions as nodes, this metric space is populated by a set of topological graphs that we refer to as 'elementary graphs'. These, recall, are defined as the minimal unit that contains a lexical predicate (the 'anchor' of the elementary graph), its functional modifiers (see Frank's 2002: Chapter 2 discussion of 'extended projections'), and its non-clausal arguments. Elementary graphs are the neighbourhoods of lexical predicates. If these neighbourhoods are open, elements within one neighbourhood become accessible from other neighbourhoods: this allows for the composition of local neighbourhoods and the formation of complex structures. The remainder of the paper will focus on the mechanisms underpinning the generation of derived syntactic objects and the relations that can be established between expressions in these.

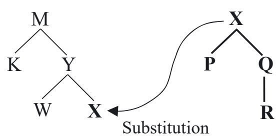
4.1 Structure composition

As an introduction to the problem of structure composition, consider the formulation of the so-called Address Issue:

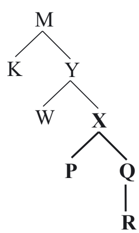
> Whenever a phrase-marker K is divided into complex sub-components L and M […], the daughter phrase-marker M that spells-out separately must correspond to an identical term M within K. (Uriagereka 2012: 75).

Uriagereka's requirement proposes a solution to the problem that arises in all models of cyclic structure interpretation: 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 to get, for example, a compositional semantic interpretation? A theory of syntax must provide a way to capture structure composition in addition to locality, optimally without having to invoke additional structure in the form of semantically vacuous non-terminal nodes. In looking at composition operations, let us start with Substitution. In traditional phrase structure terms, let M be a term, and let Y be a term within M, with a node X in its frontier. Furthermore, let X be the root node of a distinct term (not a part of M). Then, we can substitute the node X in M with the sub-tree X:

20) a.

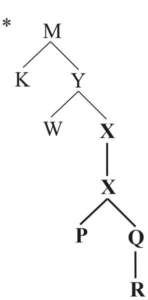


b.



Substitution does not create new nodes: (20b) contains a single node labelled X. This is consistent with a characterisation of structure composition as graph union, where syntactic objects are indexed by uniquely identifying addresses. Substitution never delivers (20c) (see Frank 2002: 17, ff. for discussion):

c. *



This operation is involved in the standard TAG analysis of clausal complementation (Frank 2006: 153; Kroch and Joshi 1985: 27, ff.; XTAG Group 2001: 91, ff.):

21) John wished [that Mary would forgive him]

(21) features two clauses in a hypotactical relation; each of these corresponds to an elementary graph by virtue of containing a single lexical predicate (more specifically, by being the neighbourhood of a single lexical predicate in WS). The bracketed clause (call it L), with lexical anchor *forgive*, is subordinated to the elementary graph anchored by *wish* (call it M). Under standard phrase structure assumptions, the elementary graph of *wish* contains a subject and a placeholder in its frontier that corresponds to the root of the direct object of *wish* (Chomsky 1955: 481, ff., 1995b); the elementary graph of *forgive* contains its arguments (a subject and an object) and a functional modifier in the form of the auxiliary *would,* which cannot anchor an elementary graph. Then, (22) is a derivation of (21) using Substitution:

22) a. [$_{M}$ John wished [L]]]
b. [$_{L}$ that Mary would forgive him]
c. [$_{K}$ [$_{M}$ John wished [$_{L}$ that Mary would forgive him]]] (via Substitution targeting L)

In this case, what we have done is link the neighbourhoods of two basic expressions, *wish* and *forgive*, by identifying the same addressed node in both syntactic objects: the address we have called L in M (22a) points to the same object as the address L in (22b), namely, the root of the sub-graph that is a structural description for the string *that Mary would forgive John*.

This is an important point, since it highlights what our definition of workspaces can do for us. In the structural description of (21) there are two distinct objects *John* and *him* only if a structural description is a set of lexical tokens (Uriagereka 2008: 16): in that case, 'John' and 'him' are lexical tokens introduced at distinct points in the derivation and bearing no relation to each other unless an index says so. This is unavoidable if the lexicon and WS are severed from each other, and structure is strictly set-theoretic (as we saw with (6c)). However, because of how we have defined WS, structural descriptions are specifications of the formal relations between basic expressions defined in a lexically anchored directed graph. Above we emphasised that the composition of local lexicalised neighbourhoods involves structure sharing, in that common addresses in distinct elementary graphs collapse to a single object in a derived structure: structure composition is graph union. Then, we need to revise (22) as in (23), where we substitute the expression *John* with the address ⦃John⦄, which uniquely points to the semantic value of *John*:

23) a. [$_{M}$ ⦃John⦄ wished [L]]
b. [$_{L}$ that Mary would forgive ⦃John⦄]
c. [$_{M}$ ⦃John⦄ wished [$_{L}$ that Mary would forgive ⦃John⦄]]

Once L substitutes for L in M, there are two calls for the address ⦃John⦄, each of which belongs to a distinct neighbourhood. Given the definition of Merge in Section 2.1, coupled with the assumption that predicates directly dominate their arguments (Krivochen 2023a, 2023b; McKinney-Bock and Vergnaud 2014; Osborne et al. 2011; see also Ermolaeva and Kobele 2022 for a related, but fundamentally different, perspective), in the derived structure (23c) we have a single point in WS, with address ⦃John⦄, immediately dominated by *wish* and by *forgive* in their elementary graphs, establishing distinct syntactic relations with each of these predicates. We can sketch a derivation of (21) along familiar Minimalist lines as in (24), where the order between arcs is determined by the sequentiality of Merge (see also fn. 10):

24) <u>Elementary graph 1</u>:
a. Merge(forgive, John) = *e*<forgive, John>
b. Merge(*v*, forgive) = <*e*<*v*, forgive>, *e*<forgive, John>>
c. Merge(*v*, Mary) = <*e*<*v*, Mary>, *e*<*v*, forgive>, *e*<forgive, John>>
d. Merge(would, *v*) = <*e*<would, *v*>, *e*<*v*, Mary>, *e*<*v*, forgive>, *e*<forgive, John>>
e. Merge(would, Mary) = <*e*<would, Mary>, *e*<would, *v*>, *e*<*v*, Mary>, *e*<*v*, forgive>, *e*<forgive, John>>
f. Merge(C_{that}, would) = <*e*<C_{that}, would>, *e*<would, Mary>, *e*<would, *v*>, *e*<*v*, Mary>, *e*<*v*, forgive>, *e*<forgive, John>>
<u>Elementary graph 2:</u>
a. Merge(wish, C_{that}) = *e*<wish, C_{that}>
b. Merge(*v*, wish) = <*e*<*v*, wish>, *e*<wish, C_{that}>>
c. Merge(*v*, John) = <*e*<*v*, John>, *e*<*v*, wish>, *e*<wish, C_{that}>>
d. Merge(T, *v*) = <*e*<T, *v*>, *e*<*v*, John>, *e*<*v*, wish>, *e*<wish, C_{that}>>
e. Merge(T, John) = <*e*<T, John>, *e*<T, *v*>, *e*<*v*, John>, *e*<*v*, wish>, *e*<wish, C_{that}>>

It is important to emphasise that despite the fact that there are two lexical predicates, and therefore two elementary graphs, there are not two workspaces: each lexical predicate does define its own neighbourhood, but these are just two regions of the same space. Because the address ⦃John⦄ is called in both elementary graphs (and *as per* the identity property ⦃John⦄ is at distance 0 from itself) when these are combined the derived graph contains only a single node addressed ⦃John⦄, assigned grammatical functions in distinct elementary graphs (which underpins the pronominal exponent; Krivochen 2023b: 225, ff.; cf. Lees and Klima 1963: 23):

24′) $G_{derived}$ = <*e*<T, John>, *e*<T, *v*>, *e*<*v*, John>, *e*<*v*, wish>, *e*<wish, C_{that}>, *e*<C_{that}, would>, *e*<would, Mary>, *e*<would, *v*>, *e*<*v*, Mary>, *e*<*v*, forgive>, *e*<forgive, John>>

The relation between T and the external argument, which Minimalism requires for Case reasons, is represented without movement, by means of an edge from T to this argument. A more strongly lexicalist analysis is possible (Krivochen 2023b): we may require that arguments be dominated directly by the anchors of elementary graphs in which they occur, since it is the lexical predicate that anchors an elementary graph that determines the number of co-occurring arguments, their category, and their thematic roles. Under these assumptions, Elementary Graph 1 would be defined as in (25):

25) $EG_1 = \langle e\langle C_{that}, \text{would}\rangle, e\langle \text{would, Mary}\rangle, e\langle \text{forgive, Mary}\rangle, e\langle \text{forgive, John}\rangle\rangle$

In (24), *Mary* is an argument of the functional category *v*, and is not dominated by *forgive*. In (25), in contrast, it is the lexical anchor *forgive* that subcategorises for both *Mary* and *John*, and immediately dominates them. This entails, in turn, that Merge has applied to (forgive, John) and (forgive, Mary), and in each of these mergers a selectional feature of *forgive* has been satisfied. Having all arguments in a neighbourhood be immediately dominated by that neighbourhood's anchor seems to us to be a stricter way of lexicalising elementary graphs (see also fn. 23). Note that this does not necessarily entail eliminating *v*, T, etc.: it means that functional categories (justified in terms of their interpretative properties) do not subcategorise or theta-mark arguments. Any relation between a functional category and an argument needs to be independently justified (e.g., C_{wh} must enter a local relation with a *wh* expression to satisfy its *wh* feature; T must enter a local relation with a nominal for agreement purposes).

In a traversal through the derived structure (23c), ⦃John⦄ in M is ordered before ⦃John⦄ in L, since L is embedded in M (i.e., there is a node in M which dominates the root of L): this will become important for pronominalisation, which we will look at shortly. Note that the operation Substitution itself can apply because the grammar can identify identical indexed nodes (in a phrase structure tree, the set of nodes is indexed by a set of labels). No additional mechanism or condition (such as Chomsky's 2021 Stability) is required to deliver this result, since if nodes on distinct elementary graph T and T′ are assigned the same address, then a derived graph T″ defined as the union of T and T′ will contract those nodes into one. In our system, Merge never produces copies (the number of elements in WS never increases during a derivation), it defines arcs between indexed expressions. New relations may be defined throughout the derivation, on top of existing relations.

Importantly, as long as the relevant node receives an interpretation in all elementary graph where it occurs, the neighbourhoods of two or more lexical predicates may share more than one node (see e.g. (17) above). This is precisely what we need for (23): (23a) contains a call for an address (which we called L) that coincides with the root of the graph (23b) in its frontier; in addition to this, there is another common address embedded in both structures, ⦃John⦄. All syntactic dependencies are established within local neighbourhoods, but these neighbourhoods are open, and in consequence can contain points that also belong to other neighbourhoods. Uriagereka's 'identity condition' is defined here in terms of the addressing system.

We must emphasise that, if syntactic operations are required to yield trees in which an element cannot be dominated by more than a single node (the Single Mother Condition), as in Minimalist syntax, then we are required to multiply the entities in structural descriptions: because syntactic context is defined in terms of dominance (e.g., Chomsky 2001: 39), *John* dominated by the root in M and *him* dominated by *forgive* in L already have a mother, and cannot have another.[21] Each must be a distinct object, introduced at different derivational steps. Under strict set-theoretic assumptions, where M and L are sets of (sets of) syntactic terms, we cannot say that ⦃John⦄ in M and ⦃John⦄ in L ($⦃\text{John}⦄_{\mathrm{M}}$ and $⦃\text{John}⦄_{\mathrm{L}}$, for convenience) are either 'the same object' or a 'discontinuous' one (Chomsky et al. 2019: 232), since it would entail that ⦃John⦄ both belongs and does not belong to the set L (see Gärtner 2022 for further discussion). However, this multiplication of entities does not arise in our proposal, as a consequence of the definition of workspace as a topological space: the identification of $⦃\text{John}⦄_{\mathrm{M}}$ and $⦃\text{John}⦄_{\mathrm{L}}$ amounts to having $d(⦃\text{John}⦄_{\mathrm{M}}, ⦃\text{John}⦄_{\mathrm{L}}) = 0$, which -again- is a direct consequence of the identity property. This is only possible because we are defining graphs in the workspace: a topological perspective on the notion of workspace allows us to simplify the mechanisms of the grammar, in this case dispensing with independent indexing mechanisms for identical terms in distinct syntactic contexts in favour of a property of the workspace.

4.2 Chains, copies, and repetitions

The approach to chain formation in Martin and Uriagereka (2014: 174, ff.) allows for a phrase marker to 'fold', collapsing distinct links of the chain into a single object. 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**.

Consider a case like

26) Which paper did John review?

21 Chomsky et al. (2019: 232) define the 'context' of an object as 'its structural sister'. We assume they are referring to the set-theoretic relation of co-membership. In this case, too, *John* and *him* belong two different sets and are introduced at different derivational steps.

In Martin and Uriagereka's view, IM 'turns [a phrase marker] into a topologically more complex sort of object' (2014: 175). This complexity is due to the fact that the phrase marker 'folds' on itself, delivering a toroid shape. The underlying representation would be (27):

27) Which paper did John review ~~which paper~~?

Under current assumptions, the core of Martin and Uriagereka's analysis can be captured straightforwardly. In the simplest analysis available, the sub-graph with root ⦃which⦄ (the irreducible expression of which contains ⦃paper⦄ immediately dominated by ⦃which⦄: *e*<which, paper>) is immediately dominated by both *review* and C_{wh}, with each of which it establishes distinct relations: in McKinney-Bock and Vergnaud's (2014) terms, Selection with *review* and Checking with C_{wh}.[22]

An important consequence of our system is that, as per the identity property and unique indexing, the distance between 'copies' of *which paper* will always be 0, regardless of how much structure is introduced between 'filler' and 'gap':

28) Which paper [did Mary say that Sue suspected that] John reviewed ~~which paper~~?

The derived expression *which paper* is dominated only by *review* and by the matrix C_{wh} in both (27) and (28). Such an analysis follows LTAG's Non-Local Dependency Corollary (NDC):

> Nonlocal dependencies always reduce to local ones [defined within elementary trees] once recursive structure is factored away. (Frank 2002: 27)

Because they are established at the level of local neighbourhoods, which are units of argument structure, all syntactic dependencies must be local: *which paper* is never removed from the neighbourhood of *review*. The relation created between *review* and *which paper* in the directed (sub)graph G = <*e*<review, John>, *e*<review, which>, *e*<which, paper>> is not disrupted throughout the derivation: when C_{wh} is introduced, a new edge *e*<C_{wh}, which> is created via Merge(C_{wh}, which). At this point, *which* is the tail of two edges (one with head *review* and another with head C_{wh}). Nothing is copied, since we can call the address ⦃which⦄ (which is all we need, because *which* immediately dominates *paper* allowing for a walk to be defined from C to *paper*) more than once, and the indexing system implemented in a metric space ensures that any (derived) expression with (derived) address ⦃which paper⦄

22 A more articulated structure for the *wh* operator is also possible (e.g., Cable 2010; Johnson 2016, 2020), where the operator and variable readings are divorced. Johnson's multidominance analysis, for instance, has a separate Q morpheme which dominates DP under CP, but the V that subcategorises for the DP does not dominate Q.

(computed from ⦃which⦄ immediately dominating ⦃paper⦄; see also Gorn 1967a: 215, ff.) is at 0 distance from any expression with the same address.

Working with lexicalised neighbourhoods in WS has other advantages. Compare (26) with (29):

29) *Which paper did John fall asleep while Mary reviewed ~~which paper~~?

An analysis following the lines of Uriagereka (2002), Nunes (2004), or Franks (2014: 224) involves multiple workspaces and derivational cascades: in these frameworks, abstracting form certain differences, the opacity of adjuncts follows from (i) being derived in a separate workspace, and (ii) syntactic dependencies (probe-goal, filler-gap, etc.) operating with elements within a workspace/derivational cascade. In Uriagereka's system, parallel derivational cascades are inserted in the main derivation 'atomised', already Spelled-Out, their internal structure being inaccessible for purposes of operations triggered at the main cascade (2012: 183).[23] Even in a multidominance approach, however, additional restrictions must be put in place to prevent sidewards movement from deriving (29) (as Franks 2014: 224–225 observes). In our theory, there are no 'multiple workspaces' but only one, whose topological properties change (in contrast to Martin and Uriagereka's view, it is not the topological properties of a phrase marker in the workspace that change, but of WS itself). Nothing in the definition of multidominance filters out (29) by itself, but working with lexicalised neighbourhoods and assuming the NDC delivers the desired results: there is no call for the address ⦃which paper⦄ in the neighbourhood of *fall asleep* (taken here as a multi-word basic expression), since the predicate only subcategorises for a subject (see also Frank 2002: 211, ff.):

23 Uriagereka (2002: 52–53), following Chomsky (1995b), claims that the lack of superiority effects in (i) and (ii) is due to equidistance, which he deduces from conditions over Multiple Spell-Out:

i) Which professor *t* saw which student?
ii) Which student did which professor see *t*?

Since *which professor* is a complex phrase in a non-complement position, it must be derived in parallel, and thus *which professor* and *which student* (a complement) 'do not compete within the same derivational space'. In the present framework, the 'equidistance' can be deduced from the simplified format of structural descriptions: *see* immediately dominates the root of the treelets corresponding to *which professor* and *which student*. While in Uriagereka's system locality is enforced through LCA-linearisable command units, for us locality is enforced via lexicalisation *plus* the requirement that all dependencies be local to elementary graphs. In both (i) and (ii) there is a single anchor, thus a single elementary graph, and no inter-graph dependencies. Both subject and object are at the same level of embedding, immediately dominated by *see* (the order between the arcs of which they are tails being determined by the sequentiality of Merge).

30) Elementary graph 1:
EG_1 = <*e*<T, John>, *e*<T, fall-asleep>, *e*<fall-asleep, John>>
Elementary graph 2:
EG_2 = <*e*<while, T>, *e*<T, Mary>, *e*<T, review>, *e*<review, Mary>, *e*<review, which>, *e*<which, paper>>

With EG1 being the matrix structure (the 'initial graph', adapting again TAG terminology), there is no structural position licensed for the filler: if, as in TAGs, we attempt to adjoin EG2 to EG1, we will not get *which paper* taking scope in the matrix structure, as it is not an argument or a modifier of the anchor of the matrix structure.

In the case of parasitic gaps, however, there is a call for the address ⦃which paper⦄ in all neighbourhoods involved (with the structurally highest one receiving an operator interpretation, possibly along the lines of Johnson 2016). Therefore, in

31) Which paper did John file ~~which paper~~ after reviewing ~~which paper~~?

there are two neighbourhoods: one anchored by *file* and the other by *review*. Each of these contains a call for the (derived) address ⦃which paper⦄,[24] which receives the same grammatical function (object) in both, being in the two cases first-Merged with the anchor. This address is at distance 0 from itself. In this context, the results of composition-as-graph union are similar to Martin and Uriagereka's (2014) 'collapse' approach, but in our theory these effects derive from the properties of WS (as defined in Section 2) and from the addressing system. The core idea is that if we compose two (or more) graphs all of which contain an expression with the same address, the resulting structure will only contain one such expression, which enters dominance relations with expressions in each of those graphs.

4.2.1 Deriving terseness in chain dependencies

Let us now look at some apparently simple cases:

32) a. John admires John
b. John admires himself
c. *John admires (intended: 'John admires himself')
d. *Who$_i$ does John$_i$ admire?

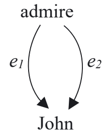
The two *John*'s in (32a), with neutral intonation, do not refer to the same entity: they have distinct semantic value, and consequently distinct addresses (say, ⦃John McLaughlin⦄ and ⦃John Bonham⦄). They would, then, be 'repetitions': the concept of repetition refers only to PF exponents, not to addresses or semantic values. In (32b),

24 Under a structure sharing analysis of English exhaustive obligatory control, ⦃John⦄ is also called in both elementary graphs (Krivochen 2023b: §6.4).

whether there is a copy or not would depend on the derivational origins of reflexives: if reflexives are derivational effects of having two co-indexed nominals in a local[25] environment (Gärtner 2014; Grohmann 2003; Hornstein 2001; Lees and Klima 1963; McCawley 1970, among others), then a copy is needed, but no repetitions. (32c–d) show that additional constraints must hold: if *John* and *himself* are head and tail of a chain, how can we account for the fact that reflexive predicates require the Spell-Out of both calls to the same address but interrogatives do not?

In the present view, the definition of workspace must play a role in accounting for these facts. What matters in (32a) is that both instances of the word 'John' do not point to the same region in WS, as they have different semantic values. This is given, in that (32a) cannot be interpreted as (32b): we need to provide an adequate description of why this is the case. Under the assumption that the syntactic computation manipulates lexical tokens, it is indeed difficult to see how to differentiate between (32a) and (32b) without additional devices. However, if we consider our definition of neighbourhood, we can say something interesting on the matter. Recall that in our approach syntactic operations link points, creating graphs in a metricised space. The crucial fact about (32b) is that there is an address which is required to enter two distinct relations with the predicate *admire* (subject-of and object-of): we follow Reuland and Reinhart (1993: 662) in defining a predicate as reflexive if two of its arguments are coindexed. In our approach, this means having an expression which establishes two distinct grammatical relations with a predicate: derivationally, the predicate first-Merges and second-Merges with the same basic expression. The irreducible graph thus generated would be G = <*e*<admire, John>, *e*<admire, John>>, diagrammed in (33):[26]

33)



The definition of 'reflexive anaphor' in this context (recasting Reuland and Reinhart's insight) is a nominal expression that corresponds to a node that is the tail of parallel arcs (which under present assumptions necessarily makes the 'anaphor' and

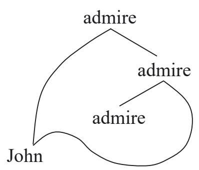
25 'Local' means here 'within the neighbourhood of a single lexical predicate'. Specifically, for this case, the neighbourhood of a transitive verb and which does not include a passive auxiliary.

26 The relation *parallel* is crucial to this analysis, as it allows an expression to be in two different grammatical relations with another. In Postal's (2010: 18) terms,

> one good reason to assume parallel arcs is that for certain types of grammatical relations a single phrase has the possibility of bearing more than one to the same larger constituent.

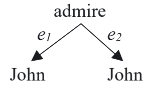
its ‘antecedent’ co-arguments): arcs e_1 and e_2 are parallel iff $e_1\langle v_i, v_j\rangle \wedge e_2\langle v_i, v_j\rangle$. If we adopted a BPS-style diagram, the definition of *parallel arc* would need some reworking, since each call to the address ⦃John⦄ is immediately dominated by distinct segments of the same category:[27,28]

34)



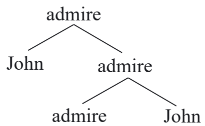
The analysis is different in the case of (32a): here, each *John* corresponds to a distinct indexed expression. The structural description of (32a) features no parallel arcs:

35) a.



Irreducible graph

b.



BPS

The difference between (32a), which in Chomsky’s terms would involve ‘repetitions’ (of an exponent) and (32b), which would involve ‘copies’, is not something that the grammar must invoke any special operations or principles to account for other than properties of the workspace: in one case there is one indexed expression that establishes more than one relation with a single predicate, in the other, there are two distinct nodes with two distinct addresses,

27 Alternatively, BPS-style trees could be read as analogous to TAG derivation trees (Frank 2006: 160; Sarkar and Joshi 1997: 610), keeping track of the order of introduction of arguments and adjuncts and the category of derived expressions resulting from these composition operations (with interesting consequences for the definition of grammatical functions, see Dowty 1982). In this case, however, BPS-style diagrams would be strictly analytical tools, rather than objects in the workspace. We leave this intriguing possibility aside in this paper.

28 (34) can be converted into (33) (and (35b) into (35a)) under a graph-theoretic interpretation of BPS via *edge contraction* (see fn. 11).

which happen to share PF exponents.[29] Crucially, these exponents cannot be part of the addresses, nor of their content: the theory defended here requires either late insertion of PF exponents or the possibility of modifying exponents throughout the derivation.

In this context, a configurational definition of local reflexivity allows us to do two things: first, dispense with indexing and avoid the multiplication of nodes in structural descriptions by not having an endophoric device (anaphor or pronominal) and its antecedent be distinct syntactic objects. This configurational approach to binding is a feature that our analysis has, to greater or lesser extents, in common with that of Hornstein (2001: Chapter 5) -based on Movement-, Gärtner (2014) -implementing multidominance on a Phrase Linking Grammar (see also Peters and Ritchie 1981)-, Perlmutter (1980: 209) -using 'multiattachment' in Relational Grammar- and Grohmann (2003, 2013) -as a repair strategy for anti-locality violations-, among others. Second, and differently from the aforementioned works, we can derive the distinct application of 'generalised terseness' to reflexives and *wh*-dependencies from configurational differences. Let us see how.

Johnson (2020: 113) provides a definition of 'terseness' as a condition on chain Spell-Out (see also Grohmann's 2003, 2013 *Condition on Domain Exclusivity*, which enforces terseness within local syntactic domains):

> Terseness: When a term is moved from one position to another, it gets spoken in only one of those positions.

There is no movement *stricto sensu* in our analysis of anaphoric dependencies, but Johnson's condition can be generalised beyond movement chains:

29 Collins and Groat (2018) consider a multidominance approach (specifically along the lines of Citko 2011, which has some differences with respect to the framework explored here) to the distinction between copies and repetitions in Minimalism. Their critique of the multidominance approach begins as follows:

> One issue that comes up right away is that [a multidominance tree] is a graph theoretic object. In minimalism, Merge forms sets {X, Y}

The argument proceeds by rejecting a graph-theoretic approach and considering problems with a set like {John$_1$ {T, {be, {seen, John$_1$}}}}, Collins and Groat correctly point out that the only way to make it work in Minimalism (where syntactic terminals are lexical tokens) is by introducing indices, thus violating the Inclusiveness Condition (see also Krivochen 2015b). The system proposed here does not require diacritics: either the two 'John's have the same address or they do not. Our system is based on the idea that WS contains graphs rather than sets, but even in a strict set-theoretic Minimalist approach the core issue seems to be defining how syntactic terminals are defined and what nodes in a phrase structure tree stand for (see also Gärtner 2022).

> Generalised terseness: When the address corresponding to a term is called from more than one syntactic context, it gets spoken in only one of those contexts.

We have a puzzle in our hands (as observed by a reviewer): if the same mechanism underlies both reflexives and *wh*-dependencies, one may ask why terseness only applies to one of these. In other words: why is (36a) grammatical but (36b) ungrammatical?

36) a. Who does John admire ~~who~~?
 b. *John admires ~~John~~ (intended: John admires himself)

The first thing to note is that despite the formal mechanism deriving both cases being the same (and in both cases we have a single elementary graph due to there being a single lexical anchor), the syntactic configuration delivered by this mechanism is different: only (36b), in the intended reading, could contain parallel arcs. If parallel arcs were involved in (36a) (such that *who* and *John* were 'coindexed'), the result would be a case of strong crossover (cf. (32d)), which is independently ruled out. (36a) and (36b) have distinct structures, (37a-b) respectively:

37) a. G = <*e*<C_{wh}, who>, *e*<C_{wh}, T>, *e*<T, John>, *e*<T, admire>, *e*<admire, John>, *e*<admire, who>>
 b. G = <*e*<T, John>, *e*<T, admire>, *e*<admire, John>, *e*<admire, John>>

There is a configurational difference between (36a) and (36b) that disappears under traditional Merge.

As for why generalised terseness applies to only one of these configurations (or, rather, why it cannot apply to (36b)), our proposal is that the answer lies in what Grohmann (2003: 78, 2013) calls 'drastic effects on the output': an 'unexpected' realisation of a syntactic object SO at PF, motivated by a violation of anti-locality. Grohmann defines contextual syntactic domains (which he calls 'Prolific Domains'), and requires that each syntactic object have a unique address identification (an LF occurrence of SO) per domain. If movement of SO does not cross the boundaries of a Prolific Domain, a violation of anti-locality ensues. Such a violation causes the derivation to crash, unless having more than one occurrence of a SO in a domain yields an output effect. In our case, the 'drastic effect' is the materialisation of a reflexive exponent *himself*, triggered by the assignment of two different grammatical functions (subject and object) to the same basic expression within an elementary graph: this double call for the address ⦃John⦄ as the tail of distinct arcs headed by the same predicate counts as 'too local' in the same way that movement within a Prolific Domain does.

The core idea is that assigning two distinct grammatical functions to the same addressed expression ⦃John⦄ by a unique lexical predicate within an elementary graph licenses a violation of generalised terseness (in the form of a minimally distinct feature-matching exponent to the tail of the lowest arc) in the same way that Grohmann's (2003, 2013) Copy Spell-Out does, but -emphatically- without movement or indeed copies.[30] The situation is different in *wh*-extractions: there is only one primary grammatical function involved (object, in (36a)), and conditions for these 'drastic effects' to arise do not obtain. Accordingly, we expect resumption (also a violation of generalised terseness) in *wh*-interrogatives to be banned in these contexts:

38) a. *Who$_i$ does John admire him$_i$?
    b. *Who$_i$ he$_i$ admires John?

This analysis also correctly excludes the possibility of Spelling-Out A-movement 'copies' in the standard cases: in passivisation, there is only one grammatical function assigned to the raised syntactic object; in raising there is more than one elementary graph involved (Frank 2002: Chapter 3) such that the raised term (which is structure-shared; see Börjars et al. 2019: 102, ff.; Pollard and Sag 1994: 136) gets assigned a single grammatical function in each: either subject upstairs and downstairs (with verbs such as *seem*) or object upstairs and subject downstairs (with verbs such as *expect*). This approach entails that the conditions for the assignment of PF exponents to syntactic terminals are checked at the level of elementary graphs, as is semantic interpretation (along the lines of Jacobson's 2012 direct compositionality Type 3): elementary graphs are the unit to which output constraints apply. As such, while in our theory non-monotonicity does not require the multiplication of

30 Copy Spell-Out is a mechanism whereby the lower copy of a syntactic object created via movement within a local domain can be Spelled-Out as a distinct minimally feature-matching exponent. In Grohmann's (2003, 2013) system, Copy Spell-Out is a repair strategy for violations to his Condition on Domain Exclusivity, which bans multiple occurrences of an SO within a local syntactic domain ('An object O in a phrase marker must have an exclusive Address Identification AI per Prolific Domain ΠΔ, unless duplicity yields a drastic effect on the output'; 2003: 78). A semi-formal formulation of Copy Spell-Out is as follows (from Grohmann 2013):

> Copy Spell-Out (PF-driven viz. the CDE [Condition on Domain Exclusivity])
> Structural Description: Given copies C_1 and C_2, where C_1 LCA-precedes C_2, and C_1 and C_2 are in the same Prolific Domain.
> Structural Change: Spellout-as-Modified C_2

The basic idea is clear: syntactic dependencies are local, but cannot be *too* local. The major difference between Grohmann's view and ours is that we propose no movement (thus no copies) in the derivation of reflexives.

workspaces (compare with Nunes 2004; Uriagereka 2002), locality is nonetheless enforced, via lexicalised neighbourhoods.[31]

The identification of identical addresses in distinct contexts follows from our characterisation of WS, and proves particularly useful to address the problem of distinguishing between *copies* and *repetitions*. The terms 'occurrence', 'copies', and 'repetitions' have been used in Minimalist syntax to argue for or against specific forms of structure building (Chomsky 2019, 2021; Collins and Groat 2018); in all cases their problematic nature is widely acknowledged. Many of the problems identified in the literature arise because derivations operate over sets of tokens lexical items and sets of sets of tokens of lexical items. For example, Chomsky (2013: 40) uses 'copies' and 'repetitions' as types of occurrences. Chomsky (2000: 115) says that "an occurrence of α in K to be the full context of α in K", where K is a syntactic object. As observed above, 'context' refers to the mother-daughters of α in a PS tree, but set-theoretically we should refer exclusively to relations of membership and co-membership, not 'dominance' (also Chomsky 2001: 39). Once we do, however, we need either indexing to keep track of occurrences, or allow for multisets to have strict identity between copies (despite changes in their featural makeup). Both options present, as has been pointed out in the literature, additional problems.

At the core of the problem of distinguishing copies and repetitions there is a confusion, we think, between phrase markers as sets, phrase markers as graphs, and diagrams of phrase markers (see McCawley 1998: 47–48; Postal 2010: 7, ff. for discussion). Making explicit the mechanisms of each is necessary to properly compare the empirical adequacy of set theoretic and graph theoretic syntax.

Because we are dealing with two or more calls to the same location in WS, we can capture the idea of *copies* without multiplying nodes or extending WS: this multiplication is unavoidable in set-theory based syntax but avoided in graph-theoretic syntax. When neighbourhoods are combined (at the root or non-root), identical addresses in distinct neighbourhoods are treated as a single syntactic object, as addresses are instructions to retrieve an interpretation (or access a portion of the lexical space; see Manzini and Savoia 2011: Introduction for related discussion) and structure composition is graph union.

31 Linguistic variation would arise when considering (i) 'Which language makes available which grammatical formative in which structures?' (Grohmann 2013) and (ii) the size of elementary graphs, including the empirical issue of determining what counts as a lexical predicate in different languages (Krivochen and Padovan 2021).

5 Conclusions

Let us take stock. Our starting point was the contrast between the lack of attention to the formal properties of *workspace* in Minimalist syntax and its centrality for operations of structure building and mapping. Because the properties of workspaces are often left undefined, it is hard to link properties of objects or operations to properties of the workspace. In this context, we needed to determine what we want the theory of workspaces to do for us:

- Appealing to workspaces must make the definition of syntactic relations more transparent or explicit, it must capture some property of grammatical entities that would be otherwise overlooked or mischaracterised.
- We want to capture the fact that 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.
- We want to preserve distinguishability: two distinct points A and B in a space X can be arbitrarily near or far apart, but never have 0 distance. A and B can have 0 distance in X iff A = B.
- We want to be able to define the neighbourhood of a point in the space, and determine whether that neighbourhood is accessible to other points in the space or not. This results in a workspace-based theory of local and nonlocal relations.

We proposed a topological definition of workspace, and related its properties to the kinds of operations that apply and relations that can hold in that space. We built on previous work on dynamical frustration at the core of language to argue that the initial (pre-syntactic) state of the workspace is ultrametric: points in that space are basic expressions of the language indexed by a set of uniquely identifying addresses. Syntactic operations create neighbourhoods for lexical predicates (which enforce their subcategorisation frames) and create intersections between these neighbourhoods. In connecting expressions, syntactic operations modify the topology of the workspace, delivering a set of directed graphs in metric space, where internal and boundary sets can be defined. The metric space which results from the composition of local neighbourhoods delivers topological distinguishability as well as variable distances between points (Munkres 2000; Willard 1970). Structure sharing is enforced by the *identity property* of metric spaces and the addressing system. Locality is enforced by lexicalisation (not by topology), given the Elementary TAG Hypothesis: all syntactic dependencies must be established at the level of elementary graphs.[32]

32 In Krivochen (2023a) we propose that Minimal Search is also elementary graph bound.

Architecturally, IM and EM are unified as a single operation: Merge(X, Y) = *e*<X, Y>. Graph-theoretic Merge satisfies the conditions formulated in Chomsky (2021): Minimal Yield (no new elements are created), Binarity (as arcs only connect a head and a tail), and Stability (via unique indexing and multidominance). Furthermore, the workspace is never expanded. Under graph-theoretic Merge, IM and EM only differ in terms of the configurations that are created and their relation to argument structure (such that subcategorisation properties of anchors of elementary graphs are always satisfied by EM).

The empirical component of this paper pertains to the analysis of simple cases of copies versus repetitions that have been raised in the literature. Conceiving of syntactic terms in structural descriptions as addresses allows us to dispense with the multiplication of nodes in dependencies involving *copies* in favour of multiple calls for the same address, all of which are at 0 distance from each other following the identity property. It becomes possible to define structural descriptions that avoid set-membership paradoxes and simplify the system by dispensing with independent indexing mechanisms: this offers promising perspectives in the analysis of locality, long-distance dependencies, and co-reference. The relation between copies and theta-theory (highlighted in Chomsky 2020a: 44, 2021: 21, ff.) follows if Merge satisfies selectional requirements of predicates: in Chomsky's (2021: 21) words, '*Merge must satisfy LSC* [Language Specific Conditions], *including* Θ-*Theory*'. In the present framework, it does so by defining lexicalised neighbourhoods, where a lexical predicate immediately dominates the arguments it subcategorises and theta-marks.

We eliminate the distinction between a lexical space and a syntactic space: there is only *one* workspace, derivationally metricised, where strictly ordered directed graphs are constructed.

References

Alsina, Alex. 2008. A theory of structure-sharing: Focusing on long-distance dependencies and parasitic gaps. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG08 conference*. Stanford: CSLI.

Bell, Sarah. 1983. Advancements and ascensions in Cebuano. In David Perlmutter (ed.), *Studies in relational grammar 1*, 143–218. Chicago: University of Chicago Press.

Binder, Philipe. 2008. Frustration in complexity. *Science* 320. 322–323.

Blümel, Andras. 2017. *Symmetry, shared labels, and movement in syntax*. Berlin: de Gruyter.

Bondy, J. Adrian & U. S. R. Murty. 2008. *Graph theory*. New York: Springer.

Börjars, Kersti, Rachel Nordlinger & Louisa Sadler. 2019. *Lexical-functional grammar: An introduction*. Cambridge: CUP.

Cable, Seth. 2010. *The grammar of Q*. Oxford: OUP.

Carr, Jack. 2006. Center manifold. *Scholarpedia* 1(12). 1826.

Chomsky, Noam. 1955. *The logical structure of linguistic theory*. Mimeographed, MIT. Available at: http://alpha-leonis.lids.mit.edu/wordpress/wp-content/uploads/2014/07/chomsky_LSLT55.pdf.
Chomsky, Noam. 1995a. Bare phrase structure. In Héctor Campos & Paula Kempchinsky (eds.), *Evolution and revolution in linguistic theory*, 51–109. Washington D.C.: Georgetown University Press.
Chomsky, Noam. 1995b. *The Minimalist program*. Cambridge, Mass.: MIT Press.
Chomsky, Noam. 2000. Minimalist inquiries: The framework. In Roger Martin, David Michaels & Juan Uriagereka (eds.), *Step by step. Essays in Minimalist syntax in honor of Howard Lasnik*, 89–155. Cambridge, Mass.: MIT Press.
Chomsky, Noam. 2001. Derivation by phase. In Michael Kenstowicz (ed.), *Ken Hale: A life in language*, 1–52. Cambridge, Mass.: MIT Press.
Chomsky, Noam. 2004. Beyond explanatory adequacy. In Adriana Belletti (ed.), *Structures and beyond*, 104–131. Oxford: OUP.
Chomsky, Noam. 2008. On phases. In Robert Freidin, Carlos Otero & María Luisa Zubizarreta (eds.), *Foundational issues in linguistic theory*, 133–166. Cambridge, Mass.: MIT Press.
Chomsky, Noam. 2012. Foreword. In Ángel Gallego (ed.), *Phases: Developing the framework*, 1–7. Berlin: de Gruyter.
Chomsky, Noam. 2013. Problems of projection. *Lingua* 130. 33–49.
Chomsky, Noam. 2019. Some puzzling foundational issues: The reading program. *Catalan Journal of Linguistics*. 263–285. https://doi.org/10.5565/rev/catjl.287.
Chomsky, Noam. 2020a. The UCLA lectures. Available at: https://ling.auf.net/lingbuzz/005485.
Chomsky, Noam. 2020b. Minimalism: Where we are now, and where we are going. Lecture at 161st meeting of Linguistic Society of Japan. Available at: https://www.youtube.com/watch?v=X4F9NSVVVuw.
Chomsky, Noam. 2020c. Puzzles about phases. In Ludovico Franco & Paolo Lorusso (eds.), *Linguistic variation: Structure and interpretation*, 163–168. Berlin: de Gruyter.
Chomsky, Noam. 2021. Minimalism: Where are we now, and where can we hope to go. *Gengo Kenkyu* 160. 1–41.
Chomsky, Noam, Ángel Gallego & Dennis Ott. 2019. Generative grammar and the faculty of language: Insights, questions, and challenges. *Catalan Journal of Linguistics*. 229–261. https://doi.org/10.5565/rev/catjl.288.
Chomsky, Noam & Howard Lasnik. 1995. The theory of principles and parameters. In Noam Chomsky (ed.), *The minimalist program*, 13–128. Cambridge, Mass.: MIT Press.
Citko, Barbara. 2011. Multidominance. In Cedric Boeckx (ed.), *Handbook of linguistic Minimalism*, 119–142. Oxford: OUP.
Collins, Chris. 2002. Eliminating labels. In Samuel Epstein & Daniel Seely (eds.), *Derivation and explanation in the Minimalist program*, 42–64. Oxford: Blackwell.
Collins, Chris. 2017. Merge(X, Y) = {X, Y}. In Leah Bauke & Andras Blümel (eds.), *Labels and roots*, 47–68. Berlin: de Gruyter.
Collins, Chris & Edward Stabler. 2016. A formalization of Minimalist syntax. *Syntax* 19(1). 43–78.
Collins, Chris & Erich Groat. 2018. Distinguishing copies and repetitions. Available at: http://ling.auf.net/lingbuzz/003809.
Dalrymple, Mary, John Lowe & Louise Mycock. 2019. *The Oxford reference guide to lexical functional grammar*. Oxford: OUP.
Dowty, David. 1982. Grammatical relations and Montague grammar. In Pauline Jacobson & Geoffrey Pullum (eds.), *The nature of syntactic representation*, 79–130. Dordrecht: Reidel.

Dowty, David. 2003. The dual analysis of adjuncts/complements in Categorial Grammar. In Ewald Lang, Claudia Maienborn & Cathrine Fabricius-Hansen (eds.), *Modifying adjuncts*, 33–66. Berlin: Mouton de Gruyter.
Dowty, David, Robert Wall & Stanley Peters. 1981. *Introduction to Montague semantics*. Dordrecht: Kluwer.
Epstein, Samuel. 1999. Un-principled syntax: The derivation of syntactic relations. In Samuel Epstein & Norbert Hornstein (eds.), *Working Minimalism*, 317–345. Cambridge, Mass.: MIT Press.
Epstein, Samuel, Hisatsugu Kitahara & Daniel Seely. 2015a. Derivation(s). In *Explorations in maximizing syntactic minimization*, 1–23. London: Routledge.
Epstein, Samuel, Hisatsugu Kitahara & Daniel Seely. 2015b. Labeling by minimal search. In *Explorations in maximizing syntactic minimization*, 201–221. London: Routledge.
Epstein, Samuel, Hisatsugu Kitahara & Daniel Seely. 2020. Unifying labeling under minimal search in "Single-" and "Multiple-Specifier" configurations. *Coyote Papers: Working Papers in Linguistics* 22. 1–11.
Ermolaeva, Marina & Gregory Kobele. 2022. Agree as information transmission over dependencies. *Syntax* 25. 466–507.
Frank, Robert. 2002. *Phrase structure composition and syntactic dependencies*. Cambridge, Mass.: MIT Press.
Frank, Robert. 2006. Phase theory and tree adjoining grammar. *Lingua* 116. 145–202.
Frank, Robert. 2013. Tree adjoining grammar. In Marcel den Dikken (ed.), *The Cambridge handbook of generative syntax*, 226–261. Cambridge: CUP.
Frank, Robert & Tim Hunter. 2021. Variation in mild context-sensitivity: Derivational state and structural monotonicity. *Evolutionary Linguistic Theory* 3(2). 181–214.
Franks, Steven. 2014. Multiattachment syntax, 'movement effects' and spell-out. In Peter Kosta, Steven Franks, Teodora Radeva-Bork & Lilia Schürcks (eds.), *Minimalism and beyond*, 195–235. Amsterdam: John Benjamins.
Gajić, Ljiljana. 2001. On ultrametric space. *Novi Sad Journal of Mathematics* 31(2). 69–71.
Gärtner, Hans-Martin. 2002. *Generalized transformations and beyond*. Berlin: Akademie Verlag.
Gärtner, Hans-Martin. 2014. Strange loops: Phrase-linking grammar meets Kaynean pronominalization. *Linguistische Berichte* 24. 1–13.
Gärtner, Hans-Martin. 2022. Copies from "standard set theory"? A note on the foundations of Minimalist syntax in reaction to Chomsky, Gallego and Ott (2019). *Journal of Logic, Language and Information* 31. 129–135.
Gollo, Leonardo & Michael Breakspear. 2014. The frustrated brain: From dynamics on motifs to communities and networks. *Philosophical Transactions of the Royal Society B* 369. 20130532.
Gorn, Saul. 1967a. Handling the growth by definition of mechanical languages. In *Proceedings of the April 18–20, 1967, spring joint computer conference*, 213–224. New York: Association for Computing Machinery.
Gorn, Saul. 1967b. Explicit definitions and linguistic dominoes. In John Hart & Satoru Takasu (eds.), *Systems and computer science*, 77–115. Toronto: University of Toronto Press.
Grohmann, Kleanthes. 2003. *Prolific domains*. Amsterdam: John Bejnamins.
Grohmann, Kleanthes. 2013. Spell-out rules. In Hans Broekhuis & Ralf Vogel (eds.), *Linguistic derivations and filtering: Minimalism and optimality theory*, 316–352. Sheffield: Equinox.
Hale, Kenneth & Samuel J. Keyser. 2005. Aspect and the syntax of argument structure. In Nomi Erteschik-Shir & Tova Rapoport (eds.), *The syntax of aspect*, 11–41. Oxford: OUP.
Han, Chung-Hye & Anoop Sarkar. 2017. Coordination in TAG without the conjoin operation. In *Proceedings of the 13th international workshop on tree adjoining grammars and related formalisms (TAG+13)*, 43–52. Umeå: Association for Computational Linguistics.
Heim, Irene & Angelika Kratzer. 1998. *Semantics in generative grammar*. Oxford: Blackwell.

Hinzen, Wolfram. 2009. The successor function + LEX = Human Language. In Kleanthes Grohmann (ed.), *InterPhases: Phase-theoretic investigations of linguistic interfaces*, 25–47. Oxford: OUP.
Hornstein, Norbert. 2001. *Move! A Minimalist theory of construal*. Oxford: Blackwell.
Hornstein, Norbert & Jairo Nunes. 2008. Adjunction, labeling, and bare phrase structure. *Biolinguistics* 2(1). 57–86.
Huck, Geoffrey. 1984. *Discontinuity and word order in Categorial Grammar*. Chicago: University of Chicago PhD dissertation.
Idsardi, William & Eric Raimy. 2013. Three types of linearization and the temporal aspects of speech. In Theresa Biberauer & Ian Roberts (eds.), *Challenges to linearization*, 31–56. Berlin: de Gruyter.
Jacobson, Pauline. 2012. Direct compositionality. In Wolfram Hinzen, Edouard Machery & Markus Werning (eds.), *The Oxford handbook of compositionality*, 109–129. Oxford: OUP.
Jayaseelan, Karattuparambil. 2017. Parallel workspaces in syntax and the inexistence of internal merge. In Gautam Sengupta, Shruti Sircar, Madhavi Gayathri Raman & Rahul Balusu (eds.), *Perspectives on the architecture and the acquisition of syntax*, 115–136. Singapore: Springer.
Johnson, David & Paul 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. Available at: https://people.umass.edu/kbj/homepage/Content/Multi_Movement.pdf.
Johnson, Kyle. 2020. Rethinking linearization. In András Bárány, Theresa Biberauer, Jamie Douglas & Sten Vikner (eds.), *Syntactic architecture and its consequences II: Between syntax and morphology*, 113–135. Berlin: Language Science Press.
Kallemeyer, Laura. 2004. Tree-local multicomponent tree-adjoining grammars with shared nodes. *Computational Linguistics* 31(2). 187–225.
Karttunen, Lauri & Martin Kay. 1985. Structure sharing with binary trees. In *Proceedings of the 23rd association for computational linguistics*, 133–136. Chicago: University of Chicago.
Kato, Takaomi, Hiroki Narita, Hironobu Kasai, Mihoko Zushi & Naoki Fukui. 2016. On the primitive operations of syntax. In Koji Fujita & Cedric Boeckx (eds.), *Advances in biolinguistics*, 29–45. London: Routledge.
Keenan, Edward & Bernard Comrie. 1977. Noun phrase accessibility and universal grammar. *Linguistic Inquiry* 8. 63–99.
Kelso, Scott. 2012. Multistability and metastability: Understanding dynamic coordination in the brain. *Philosophical Transactions of the Royal Society of London B* 367(1591). 906–918.
Kitahara, Hisatsugu. 2020. 'Multiple specifier' configurations revisited. *Reports of the Keio Institute for Cultural and Linguistic Studies* 51. 207–216.
Kitahara, Hisatsugu. 2021. On the notion copy under MERGE. *Reports of the Keio Institute of Cultural and Linguistic Studies* 52(3). 133–140.
Komachi, Masayuki, Hisatsugu Kitahara, Asako Uchibori & Kensuke Takita. 2019. Generative procedure revisited. *Reports of the Keio Institute of Cultural and Linguistic Studies* 50. 269–283.
Krivochen, Diego Gabriel. 2015a. On phrase structure building and labeling algorithms. *The Linguistic Review* 32(3). 515–572.
Krivochen, Diego Gabriel. 2015b. Copies and tokens: Displacement revisited. *Studia Linguistica* 70(3). 250–296.
Krivochen, Diego Gabriel. 2018. *Aspects of emergent cyclicity in language and computation*. Reading: University of Reading PhD dissertation.
Krivochen, Diego Gabriel. 2023a. The search for minimal search. *Biolinguistics* 17. e9793.
Krivochen, Diego Gabriel. 2023b. *Syntax on the edge*. Leiden: Brill, in press.
Krivochen, Diego Gabriel & Andrea Padovan. 2021. Lexicalised locality: Local domains and non-local dependencies in a lexicalised tree adjoining grammar. *Philosophies* 6(3). 70.

Krivochen, Diego Gabriel & Luis García Fernández. 2019. On the position of subjects in Spanish periphrases: Subjecthood left and right. *Borealis* 8(1). 1–33.
Kroch, Anthony. 2001. Asymmetries in long-distance extraction in a tree-adjoining grammar. *Unpublished manuscript*. https://www.ling.upenn.edu/~kroch/online.html (accessed 25 March 2016).
Kroch, Anthony & Aravind Joshi. 1985. The linguistic relevance of a tree adjoining grammar. Available at: http://babel.ling.upenn.edu/papers/faculty/tony_kroch/papers/relevance3.pdf.
Kural, Murat. 2005. Tree traversal and word order. *Linguistic Inquiry* 36(3). 367–387.
Kuroda, Sige-Yuki. 1976. A topological study of phrase-structure languages. *Information and Control* 30. 307–379.
Larson, Richard. 2014. *On shell structure*. London: Routledge.
Lasnik, Howard & Juan Uriagereka. 2005. *A course in minimalist syntax*. Oxford: Blackwell.
Lees, Robert & Edward Klima. 1963. Rules for English pronominalization. *Language* 39(1). 17–28.
MacFarlane, John. 2017. Logical constants. In Edward Zalta (ed.), *The Stanford encyclopedia of philosophy*. Stanford: Stanford University. Available at: http://plato.stanford.edu/archives/fall2015/entries/logical-constants/
Manzini, Maria Rita & Leonardo Savoia. 2011. *Grammatical categories: Variation in Romance languages*. Cambridge: CUP.
Martin, Roger & Juan Uriagereka. 2014. Chains in Minimalism. In Peter Kosta, Steven Franks, Teodora Radeva-Bork & Lilia Schürcks (eds.), *Minimalism and beyond: Radicalizing the interfaces*, 169–194. Amsterdam: John Benjamins.
Martin, Roger, Román Orús & Juan Uriagereka. 2019. Towards matrix syntax. *Catalan Journal of Linguistics* 2019. 27–44.
McCawley, James. 1968. Concerning the base component of a transformational grammar. *Foundations of Language* 4. 243–269.
McCawley, James. 1970. Where do noun phrases come from? In Roderick Jacobs & Peter Rosenbaum (eds.), *Readings in English transformational grammar*, 166–183. Waltham: Ginn & Co.
McCawley, James. 1982a. Parentheticals and discontinuous constituent structure. *Linguistic Inquiry* 13(1). 91–106.
McCawley, James. 1982b. The nonexistence of syntactic categories. In James McCawley (ed.), *Thirty million theories of grammar*, 176–203. Chicago: University of Chicago Press.
McCawley, James. 1998. *The syntactic phenomena of english*. Chicago: University of Chicago Press.
McKinney-Bock, Katherine & Jean-Roger Vergnaud. 2014. Grafts and beyond: Graph-theoretic syntax. In Katherine McKinney-Bock & María Luisa Zubizarreta (eds.), *Primitive elements of grammatical theory*, 207–236. London: Routledge.
Milway, Daniel. 2022. A parallel derivation theory of adjuncts. *Biolinguistics* 16(2022). e9313.
Mori, Nobue. 2005. *A syntactic structure of lexical verbs*. Maryland: University of Maryland PhD dissertation.
Morin, Yves-Charles & Michael O'Malley. 1969. Multi-rooted vines in semantic representation. In *Papers from the fifth regional meeting of the Chicago Linguistic Society*, 178–185. Chicago: University of Chicago Press.
Munkres, James. 2000. *Topology*. London: Pearson.
Murtagh, Fionn. 2004. On ultrametricity, data coding, and computation. *Journal of Classification* 21. 167–184.
Müller, Gereon. 2004. Phrase impenetrability and Wh-intervention. In Arthur Stepanov, Gisbert Fanselow & Ralf Vogel (eds.), *Minimality effects in syntax*, 289–325. Berlin: de Gruyter.
Müller, Stefan. 2015. HPSG – a synopsis. In Tibor Kiss & Artemis Alexiadou (eds.), *Syntax – Theory and analysis: An international handbook*, 937–973. Berlin: de Gruyter.

Nerukh, Dimitry. 2009. Dynamical frustration of protein's environment at the nanoseconds time scale. *Journal of Molecular Liquids* 145(3). 139–144.
Nunes, Jairo. 2004. *Linearization of chains and sidewards movement*. Cambridge, Mass.: MIT Press.
Oehrle, Richard, Emmon Bach & Deirdre Wheeler. 1988. Introduction. In Richard Oehrle, Emmon Bach & Deirdre Wheeler (eds.), *Categorial Grammars and natural language structures*, 1–16. Dordrecht: Foris.
Osborne, Tim, Michael Putnam & Thomas Groβ. 2011. Bare phrase structure, label-less trees, and specifier-less syntax: Is Minimalism becoming a dependency grammar? *The Linguistic Review* 28. 315–364.
Perlmutter, David. 1980. Relational grammar. In Edith Moravcsik & Jessica Wirth (eds.), *Syntax and semantics 13: Current approaches to syntax*, 195–229. New York: Academic Press.
Peters, P. Stanley & R. W. Ritchie. 1981. *Phrase linking grammars*. California: Ms. Stanford University and Palo Alto.
Pollard, Carl & Ivan Sag. 1994. *Head-driven phrase structure grammar*. Chicago: University of Chicago Press.
Postal, Paul. 2010. *Edge-based clausal syntax*. Cambridge, Mass.: MIT Press.
Putnam, Michael & Thomas Stroik. 2011. Syntax at ground zero. *Linguistic Analysis* 37(3–4). 389–404.
Rammal, Rammal, Gérard Toulouse & Miguel Virasoro. 1986. Ultrametricity for physicists. *Reviews of Modern Physics* 58(3). 765–788.
Reid, Miles & Balázs Szendröi. 2005. *Geometry and topology*. Cambridge: CUP.
Reuland, Eric. 2011. *Anaphora and language design*. Cambridge, Mass.: MIT Press.
Reuland, Eric & Tanya Reinhart. 1993. Reflexivity. *Linguistic Inquiry* 23(4). 657–720.
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.
Saddy, Doug. 2018. Syntax and uncertainty. In Ángel Gallego & Roger Martin (eds.), *Language, syntax, and the natural sciences*, 316–332. Cambridge: CUP.
Sarkar, Anoop & Aravind, Joshi. 1997. *Handling coordination in a tree adjoining grammar*. Pennsylvania: University of Pennsylvania. Technical report. Available at: 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 & Robert Chametzky (eds.), *Papers from the parasession on nondeclaratives*, 202–218. Chicago: Chicago Linguistic Society.
Schmerling, Susan F. 1983. Two theories of syntactic categories. *Linguistics and Philosophy* 6. 393–421.
Schmerling, Susan F. 2018. *Sound and grammar*. Leiden: Brill.
Seely, Daniel. 2015. Merge, derivational c-command, and subcategorization in a label-free syntax. In *Explorations in maximizing syntactic minimization*, 116–154. Oxford: Blackwell.
Spencer, John, Sammy Perone & Jeffrey Johnson. 2009. Dynamic field theory and embodied cognitive dynamics. In John Spencer (ed.), *Toward a unified theory of development*, 86–118. Oxford: OUP.
Stabler, Edward. 2011. Computational perspectives on minimalism. In Cedric Boeckx (ed.), *Oxford handbook of minimalism*, 617–641. Oxford: OUP.
Stein, Daniel & Charles Newman. 2011. Spin glasses: Old and new complexity. *Complex Systems* 20. 115–126.
Stroik, Thomas. 2009. *Locality in Minimalist syntax*. Cambridge, MA: MIT Press.
Stroik, Thomas & Michael Putnam. 2013. *The structural design of language*. Oxford: OUP.
Sutherland, Wilson. 2009. *Introduction to metric and topological spaces*. Oxford: OUP.
Tesnière, Lucien. 1959. *Éléments de syntaxe structural*. Paris: Klincksieck.

Turing, Alan. 1936. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society* 42(2). 230–265.
Uriagereka, Juan. 2002. Multiple spell-out. In *Derivations: Exploring the dynamics of syntax*, 45–65. London: Routledge.
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 Franks, Teodora Radeva-Bork & Lilia Schürcks (eds.), *Minimalism and beyond: Radicalizing the interfaces*, 363–391. Amsterdam: John Benjamins.
Van Steen, Maarten. 2010. Graph theory and complex networks: An introduction. Available at: https://www.distributed-systems.net/index.php/books/gtcn/gtcn/.
Willard, Stephen. 1970. *General topology*. Reading, MA.: Addison-Wesley.
Wilson, Robin. 1996. *Introduction to graph theory*. London: Adison-Wesley.
Wurmbrand, Susi. 2014. The merge condition: A syntactic approach to selection. In Peter Kosta, Steven Franks, Teodora Radeva-Bork & Lilia Schürcks (eds.), *Minimalism and beyond: Radicalizing the interfaces*, 130–167. Amsterdam: John Benjamins.
XTAG Group. 2001. *A lexicalized TAG for English*. Pennsylvania: University of Pennsylvania. Technical report. Available at: https://repository.upenn.edu/cgi/viewcontent.cgi?article=1020&context=ircs_reports.
Yang, Yasheng. 2011. *Geometric and dynamic frustration in self-organizing systems*. Waltham: Brandeis University PhD dissertation.
Zwicky, Arnold & Stephen Isard. 1963. Some aspects of tree theory. Working Paper W-6674, The MITRE Corporation. Available at: https://web.stanford.edu/~zwicky/some-aspects-of-tree-theory.pdf.
Zyman, Erik. 2023. On the definition of Merge. *Syntax*. Forthcoming. Available at: https://drive.google.com/open?id=1KWyZrOhrf-exnZpEqyYrm0UYdAqauZ6R&authuser=0.