A workspace-based theory of adjuncts

Daniel Milway

dan.milway@gmail.com

7th December 2020

1 Introduction

Adjuncts occupy a somewhat paradoxical place in biolinguistic grammatical theory, being both ubiquitous and peripheral. They are empirically ubiquitous—a language without adjuncts would be remarkable, and it is quite difficult to even use language without adjuncts, but they are theoretically peripheral—no theory of grammar naturally accounts for adjuncts and some seem to predict that adjuncts ought not exist. This has made adjuncts into something of a thorn in the side of grammatical theorists, stopping them from developing a complete and uniform theory of grammar. In this paper, I propose that, while one recent theoretical development in biolinguistics/minimalism—the decoupling of phrase-building and labeling—has closed off one possible route to explaining adjuncts, another development—derivation by workspace—has opened up another.

The question of adjuncts can be put as follows. How is (1) structured/derived such that *(i)* it means what it means, and *(ii)* (2)-(4) are grammatical and mean what they mean?

(1) Rosie sang the song with gusto.

(2) Rosie sang the song.

(3) Rosie sang the song with gusto before dinner.

(4) Rosie sang the song before dinner with gusto.

The answer that I propose in this paper is, in its most basic expression, that adjuncts (*i.e.*, *with gusto* and *before dinner* in (1)-(4)) and their hosts (*i.e.*, *Rosie sang the song* in (1)-(4)) are derived separately from each other and only joined post-syntactically. It would, of course, be easy to answer theoretical questions if all one had to do was conjecture as I have just done. The task of the theorist is to show that such a conjecture can be made to follow from an independently plausible theory, and that is the task taken up in this paper.

I begin in section 2, by laying out my relevant theoretical assumptions with special reference to simplest merge (Collins 2017) and workspaces (Chomsky 2019). Next, I make my proposal explicit in section 3, starting at a very coarse-grain and getting progressively finer. After that, I discuss some facts that are naturally accounted for by the proposal in section 4 and some facts that seem to contradict my theory in section 5 Finally, I conclude, discussing the implications of my proposal on the broader theory of grammar in section 6

2 Theoretical Context

The current proposal is situated in the biolinguistic/minimalist theory of grammar. The core conjecture of this theory is that the human language faculty is a mentally-instantiated computational procedure which generates an infinite array of structured expressions by the recursive application of the simplest combinatory operation MERGE. The task of theorizing under this approach can be divided into two related subtasks—the formalization of the operation MERGE, and the formalization of the derivational architecture. While the former has largely been the centerpiece of minimalist program, the latter has been brought into sharp relief quite recently. In this section I will discuss current approaches to the two subtasks with reference to adjuncts where relevant.

2.1 merge and adjuncts

From the earliest work in transformational grammar (Chomsky 1957, 1965) up until early theories in the minimalist program (Chomsky 1995, 2000) the generative component of the language faculty was divided into a base subcomponent, and a transformational subcomponent. In all of these theories the base included both the mechanism for generating complex structures from simple items, and the mechanism for labelling those structures. The latter was written directly into the particular phrase-structure rules of the early theories, then derived from general X-bar principles in later theories and finally assigned by early definitions of MERGE, given below in (5) where the choice of the label γ was generally assumed to follow X-bar principles.

(5) $\text{Merge}_{v1}(\alpha, \beta) \rightarrow \{\gamma, \{\alpha, \beta\}\}$

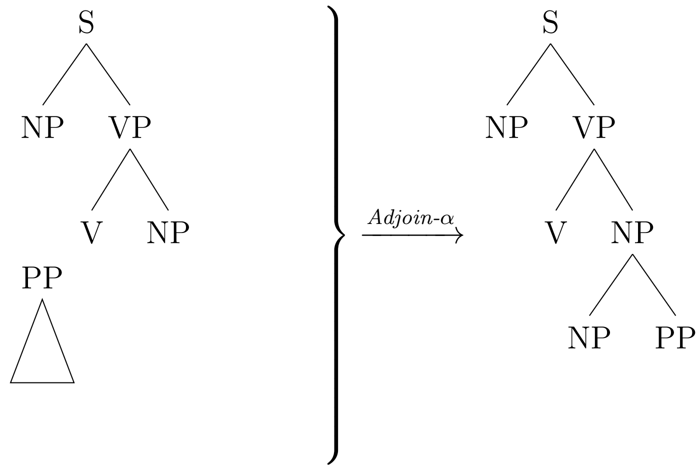
Theorists working within the minimalist program, however, have put forth various proposals for decoupling labelling from MERGE, either by eliminating labels altogether (Collins 2017) or proposing labelling as a process separate from structure building (Chomsky 2013; Hornstein 2009). Most of those theorists[1] have settled on the definion of MERGE in (6), sometimes called "simplest merge".

(6) $\text{Merge}_{simplest}(\alpha, \beta) \rightarrow \{\alpha, \beta\}$

This move, though seemingly a minor one, has major implications for the theory of grammar generally and the possibilities for a theory of adjuncts more particularly.

A move to a label-free definition of MERGE has implications for the theory of adjuncts because the theories of adjuncts within X-bar theories and early minimalist theories depended on the nature of labels and their importance for the c-command relation. For instance, Lebeaux (1988) proposed a transformation *Adjoin-*α which attaches an adjunct phrase to the maximal projection of a host phrase and then labels the resulting structure with the label of the host phrase as shown in (7)

[1] Hornstein (2009) differs, defining MERGE, not as set-formation but as concatenation.

(7) 

In contrast, Chametzky (1996), critiquing Lebeaux's proposal, argues that the node created by adding an adjunct is unlabelled. Stepanov (2001) adapts Lebeaux's theory of adjuncts to an early minimalist theory and argues that adjuncts can be added counter-cyclically without violating the *least tampering principle* because the node dominating the adjunct is not a full-fledged label but a segment of that label. Regardless of the soundness of these proposals within their respective theories, they all crucially assumed a generative procedure in which labelling and structure building were intrinsically linked. Therefore, none of these theories of adjuncts can be neatly translated into a theory in which labelling and structure building are separate from each other.

The move to a "simplest merge" theory of syntax, then, demands a novel theory of adjuncts. Chomsky (2013) has suggested that adjuncts are the result of an operation *pair-merge* which creates ordered pairs rather than sets, as demonstrated in
crefdef:PairMerge

(8) $\text{Pair-Merge}(\alpha, \beta) \rightarrow \langle \alpha, \beta \rangle$

This conjecture, though, does not constitute a novel theory of adjuncts, as there has been little to no effort to demonstrate that the empirical properties of adjuncts follow from pair-merge. So, simplest merge theories of syntax lack a theory of adjuncts.

2.2 The derivational architecture

Early minimalist theorizing focused on simplifying the architecture of the grammar by eliminating levels of representations like D-Structure, S-Structure in favour of a single derivational cycle with interfaces to independent cognitive systems. Discussion of the architecture of that derivational cycle, though has been quite limited until recently. Generally, it has been assumed that a given sentence is generated from a finite lexical array in a single linear derivation, perhaps punctuated by phases.

Recently, though, there has been increasing interest in the idea that a sentence is derived in possibly multiple subderivations, each corresponding to either the clausal spine of the sentence or its complex constituents. So, for instance, a transitive sentence like (9) would be derived in three subderivations—one corresponding to the clausal spine, and one each for the nominal arguments.

(9) The customers purchased their groceries.

Chomsky (2020) gives an explicit argument for the idea of subderivations based on extensions of Merge—Parallel Merge (Citko 2005), in particular— which exploit the fact that the domain of Merge is rather undefined. Take, for example, the hypothetical stage of a derivation in (10) consisting of an already constructed phrase $\{\alpha, \beta\}$ and an atomic object γ.

(10) $[\{\alpha, \beta\}, \gamma]$

At this stage, according to Chomsky, there should be two basic options—Internal Merge and External Merge. Internal Merge would involve Merging α or β with the set $\{\alpha, \beta\}$ resulting in a stage resembling (11), while External Merge would involve Merging γ with the set $\{\alpha, \beta\}$ resulting in the stage (12)

(11) $[\{\beta, \{\alpha, \beta\}\}, \gamma]$

(12) $[\{\gamma, \{\alpha, \beta\}\}]$

Parallel Merge, though, involves Merging α or β with γ to give a stage resembling (13).

(13) $[\{\alpha, \beta\}, \{\beta, \gamma\}]$

This, Chomsky argues, is an inevitable but unacceptable result of defining Merge as in (6), as it could be used to violate any concievable locality constraint.

The solution that Chomsky proposes involves two related conjectures—that each complex object in an expression is derived in its own encapsulated *workspace* and that a new version of MERGE, called MERGE that operates on workspaces be formulated. I will propose formal definitions of workspaces and MERGE in section 3.1.3, but some properties of these constructs are worth mentioning here. What we formerly called a stage of a derivation—*e.g.*, (10)—we now call a workspace, while stages of a derivation will be collections of workspaces. The new operation MERGE, operates on workspaces as sketched in (14) where (*a*) X and Y are syntactic objects, (*b*) WS and WS' are workspaces, (*c*) either X and Y are in WS or X is in WS and contains Y, and (*d*) WS' contains {X, Y} but does not contain X or Y.

(14) MERGE(X,Y,WS) → WS'

Setting aside issues of formalization for the time being, the theory of workspaces proposed by Chomsky (2020) suggests a picture of syntax wherein (9) is derived in three initially parallel subderivations, each associated with an encapsulated workspace, which ultimately converge to give a single clause.

2.3 The language faculty and other cognitive systems

Thus far I have only been discussing the human capacity for combining meaningful expressions to create larger meaningful expressions, often called the narrow faculty of language (FLN). Many of the empirical properties of language, though, spring from how the FLN interacts with other cognitive systems, namely the sensorimotor (SM) system which produces and prcesses external expression of language and the conceptual-intentional (CI) system which uses linguistic objects for mind-internal processes such as planning and inference. These are called *systems* rather than *modules* to indicate that they seem to be multifaceted, likely consisting of numerous interacting modules. The complexity of these systems is reflected in the difficulty of developing unified theories of morpho-phonology and semantics-pragmatics. While I will not be wading too deep into these waters, any theorizing regarding FLN requires getting one's feet wet. In this section I will discuss the aspects of the SM and CI systems and their respective interactions with FLN insofar as they will be relevant to my theory of adjuncts. Specifically, I will discuss the SM problem of mapping hierarchical structures to linear ones, the CI problem of compositionality, and the problem of distinguishing copies from repetitions which affects both systems.

In section 2.1, I discussed the fact that simplest merge decoupled phrase structure from labelling. What I neglected to mention was that it also decoupled phrase structure from linear order—the set $\{\alpha, \beta\}$ could just as easily be linearized as $\alpha^\frown\beta$ or $\alpha^\frown\beta$. In order to express a linguistic object, either in speech, sign, or writing, that object must be at least partially[2] put in a linear order. The linear order, then, must be derivable from the structures created by FLN by various principles and parameters in a way which is definite within a language but particular to that language. One of those principles is Richard Kayne's (1994) Linear Correspondence Axiom (LCA), a version of which is given in (15).

(15) **The Linear Correspondence Axiom**

For syntactic object x and y, if x asymmetrically c-commands y, then $x \prec y$.

The key insight of the LCA is that asymmetric c-command is equivalent to linear precedence in that it both are antisymmetric—if $x \leq y$ and $y \leq x$ then $x = y$—and transitive—if $x \leq y$ and $y \leq z$ then $x \leq z$. One need not look very far to find the shortcomings of the LCA *qua* theory of linearization, and likely it is only one of the many axioms at play in the linearization process. But regardless of its shortcomings, the LCA is an important proof of concept, showing that linear ordering can be derived from structure without being encoded directly in it.

[2] All modes of expression allow for some sort of simultaneous pronunciation, be it facial expressions in sign language, intonation in spoken language, or typography in written language.

Turning to the CI system, I will now address what I, perhaps misleadingly, called the problem of compositionality, which tends to be taken as the semanticists couterpart to the linearization problem. The problem is usually stated as follows: The FLN generates hierarchically structured expressions but the CI system operates on formulas of a likely higher-order predicate calculus. To solve this problem, semanticists propose various compositional principles such as function application, predicate modification (I. Heim and A. Kratzer 1998), event identification (Angelika Kratzer 1996), and existential closure (Irene Heim 1982), among others. The degree to which the problem as stated exists, though, has been called into question within biolinguistic/minimalist theorizing. Chomsky (2013, and elsewhere) argues that language is primarily an instrument of thought, which contradicts the premise that linguistic objects must be transformed into or mapped onto thought objects. If linguistic objects are thought objects, than such a premise would be akin to requiring that one convert US Federal Reserve notes to US dollars before engaging in commerce. I will be adopting this position with two caveats. First, to say that the problem of compositionality as stated is non-existent is not to say that there are no problems of linguistic interpretation. We will encounter several as I propose and refine my theory of adjuncts. Second, I will on occasion choose to represent the interpretation of some expression in formal logic when such a representation is the most perspicuous way to demonstrate some relevant property of the expression. This is not to say that formal logic has any sort of privileged status, only that it mat be useful to highlight certain properties of expressions.

Finally, I must discuss the copy-repetition distinction. Simplest merge, which decoupled phrase-structure from labelling, also combined phrase structure and transformations as its external and internal modes of operation respectively. While External Merge adds a new item to a syntactic object, Internal Merge merges one object with an object that that object contains as demonstrated in (16).

(16) $\text{Merge}_{simplest}(\beta, \{\alpha, \beta\}) \rightarrow \{\beta, \{\alpha, \beta\}\}$

The two βs on the righthand side of the arrow in (16) are *copies* of each other which means that the object represented on the righthand side of the arrow here doesn't contain two βs but rather, that β is in two positions in the newly created object. To make this more concrete, consider the passive in (17) and its approximate syntactic representation in (18).

(17) A man was seen.

(18) $\{\{a, man\}, \{\text{T}, \{\ldots\{v_{pass}\{see, \{a, man\}\}\}\ldots\}\}\}$

By hypothesis, (18) is formed by Internal Merge, combining the theme *a man* with the TP that contains it, making the two instances of $\{a, man\}$ copies of each other. Because the two instances are copies of each other, they are really only one object and therefore, they refer to the same individual and are pronounced only once. Compare this to the active in (19) and its approximate syntactic representation in (20).[3]

(19) A man saw a man.

(20) $\{\{a, man\}, \{\text{T}, \{\ldots\{v_{act}\{see, \{a, man\}\}\}\ldots\}\}\}$

In this case, the two instances of $\{a, man\}$ are not copies of each other, but merely repetitions. So, the lower instance was Externally Merged with the verb and then later the second instance was Externally Merged higher. Because the two instances are not copies, of each other, they are distinct objects and therefore, they do not necessarily corefer and they are both pronounced.

I mentioned above that copies undergo deletion by the SM system while repetitions do not. This much follows from both simplest merge and the facts of language, but question of which copies delete and when turns out to be quite complicated. If we started with the basic facts of English passives and *wh*-questions, we might propose a principle that states that only the highest copy—the copy that c-commands all other copies—is pronounced. Like the LCA, one need not look far to find exceptions,[4] but also like the LCA, the principle of

[3] I abstract away from the predicate-internal subject hypothesis for simplicity

[4] All varieties of covert movement, such as quantifier rasing (May 1978) and *wh*-in-situ (Lu, Thompson and Yoshida 2020) would contradict this proposal. Trinh (2009) discusses more nuanced copy deletion data and arrives at a constraint on the delete-low-copies principle.

"pronounce the highest copy" can serve as a demonstration that the choice of which copy to pronounce can be derived from a structure without being encoded in it.

2.4 Summary

The forthcoming proposal is made in the theoretical context of biolinguistics/minimalism, a label that, admittedly, covers a wide range of theoretical positions. In this section, I have done my best to make explicit the relevant positions under that label which I will be taking in my theoretical proposal. First, I am assuming that the basic, likely only, innate language-specific combinatory operation is simplest merge, which creates unlabelled binary sets and encompasses both the base component and the transformational component of the narrow syntax. Second, I assume that complex constituents of expressions like clauses are derived separately from each other in workspaces, a notion that requires further formalization. A corrollary of my first two assumptions is that merge must operate on workspaces. Third, I assume that, while the narrow faculty of language (FLN) is simple, perhaps consisting only of merge and the derivational architecture, the systems the interpret the objects generated by FLN, either for externalization (SM) or mind-internal computation (CI), are complex, encompassing a number of principles parameters and operations of which we understand very little.

3 The proposal

The theory of adjuncts that I propose is best viewed in contrast to the workspace theory of arguments. According to this theory, outlined in section 2.2, an argument is derived in a separate workspace from its clausal spine, and the result of that derivation is merged into clausal spine derivation. An adjunct is also derived in a separate workspace, except that that workspace is never merged into the clausal spine derivation. So the syntactic representation of (1) is given in (21) with the adjunct-free sentence derived (19) in WS1, and the adjunct

PP *with gusto* derived in WS2.

(21) $\langle[\{Rosie, \{\mathrm{T}, \ldots \{sing, \{the, song\}\}\}\}]_{\mathrm{WS1}}, [\{with, gusto\}]_{\mathrm{WS2}}\rangle$

The expression represented in (21) is grammatical insofar as the object in WS1 is a grammatical clause and the object in WS2 is a grammatical PP. Furthermore, the grammaticality of the each of the two objects—the clause and the PP—is independent of the grammaticality of other. Therefore, the clause would be grammatical without the PP, or if there were additional adjuncts, regardless of the ordering. Note that these are the three characteristic properties of adjuncts: optionality, stackability, and freedom of order.

This independence, of course, carries over to the interpretation of (21). That is, *Rosie sang the song* and *with gusto* in (21) should be interpreted the same way as a sequence of independent expressions like (22) is—conjunctively.

(22) Susan entered the room. The lights were off.

If (22) can be given a truth-value it would be the same as the truth-value of the conjunction of the two sentences. In the same way, (21) is interpreted more or less as in (23).

(23) Rosie sang the song. It was with gusto.

There is one major difference, though, between the actual interpretation of (1) and that of (23)—the former entails that the anthem-singing event and the gusto-having event are the same, while in the latter, that identity is only an implicature. This might suggest that the adjunct *with gusto* is, in fact, semantically dependent on its host clause, but such a conclusion is unwarranted. It is not so much that the adjunct is about what its host is about but rather that the host and adjunct are about the same thing. This is the case, I propose, because the host and the adjunct are constructed in the same derivation.

Turning to pronunciation, it might be suggested that my proposal introduces new complexity to the already complicated nature of pronunciation That is, our best theories suggest that c-command is vital for linearization, but there can be no c-command relation across workspaces. Such an objection, however, would mistake the nature of the linearization problem, namely that Merge creates unordered objects that must be converted to ordered object for pronunciation. A derivation stage such as (21), though, is already ordered (WS1 $\prec$ WS1), so no linearization problem should occur.

In what follows, I will refine this proposal somewhat, but the core claim—that adjuncts are in separate workspaces from their hosts—will remain the same. I pause here to note that this solution broadly accounts for adjunct without recourse to novel operations or major modifications to the architecture of the grammar, and is therefore preferable, on minimalist grounds, to theories which do introduce novel theoretical machinery such as Pair Merge.

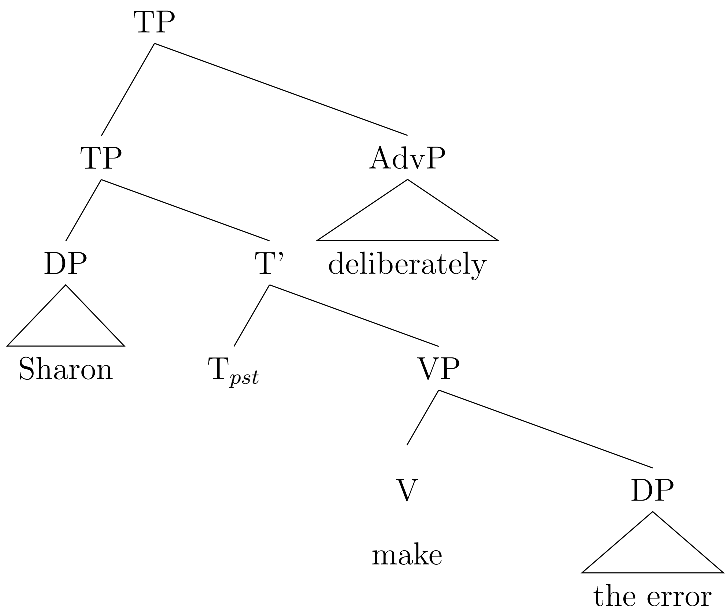
3.1 The problem of adjunct scope

The sentence in (24) is ambiguous.

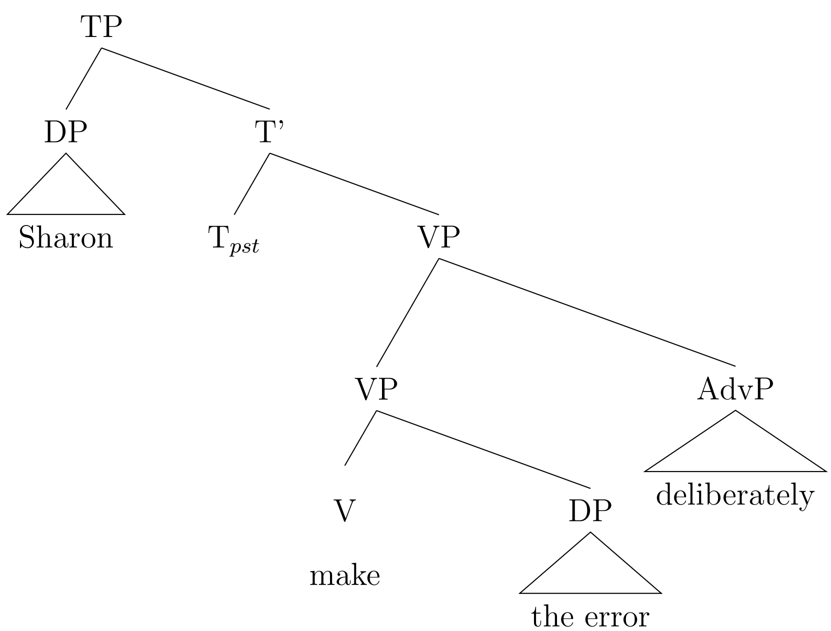
(24) Sharon made the error deliberately.

It can be interpreted as saying either that Sharon intended to make the error in question, or that she made the error in a deliberate manner. The conclusion drawn from this sort of ambiguity is that the adverb *deliberately* has two possible scopes—A high scope resulting in the first interpretation, and a low scope resulting in the second interpretation. Under an X-bar theory of adjuncts, this can be easily accounted for by aligning scope with attachment site as in (25) and (26).

(25) **The high-scope interpretation of (24) in X-bar theory**



(26) **The low-scope interpretation of (24) in X-bar theory**



As it stands, however, the workspace theory of adjuncts cannot account for adjunct scope. Or, to be more precise, it cannot account for the fact that adjuncts can have multiple scope possibilities. This can be seen when we consider how we would represent (24) in a workspace-based analysis—as the juxtaposition of *Sharon made the error* and *deliberately* as shown in (27).

(27) $$\left\langle \begin{array}{c} [\{Sharon, \{\mathrm{T}, \ldots \{\mathrm{Voice}, \{make, \{the, error\}\}\}\}\}], \\ [deliberately] \end{array} \right\rangle$$

If we take a full declarative clause to describe a situation or state of affairs, then, according to (27), (24) would describe a situation s, such that in s Sharon made the relevant error, and that s was brought about by a deliberate choice of the agent of s. In other words, the proposed workspace-based theory of adjuncts seems to predict only the high-scope interpretation of (24).

In order to modify our proposal to allow for adjunct scope, we must first realize that adjunct scope-taking is different from other kinds of scope-taking, such as quantifier scope. Usually, when we talk about scope, we have in mind an asymmetric relation. So the two readings of (28) can be described by saying which of the two quantifier phrases scopes over the other.

(28) Every student read a book.

a. $\forall s(\exists b(read(b, s)))$

b. $\exists b(\forall s(read(b, s)))$

The relationship between a modifier and a modified expression, however, is generally considered to be symmetric, at least in terms of their interpretation.[5] So, in the low-scope interpretation of (24), the logical predicate expressed by *deliberately* is conjoined with the one expressed by *make an error*, as shown in open formula (29).

(29) $(make(\text{the-error}, e) \,\&\, deliberate(e))$

It does not, then, make sense to say that *deliberately* "scopes over" the VP. We can still ask, though, why does *deliberately* conjoin with the VP and not, say, with AspP, or TP. The answer, at least in X-bar terms is obvious—the adverb and the VP conjoin because they are in the same position, that is [Comp, Voice]. In other words, *deliberately* conjoins with the VP, because both scope directly under Voice, and therefore, indirectly under everything that

[5] Setting aside cases of non-intersective modification.

scopes over Voice.

This rethinking of adjunct scope, then suggests a workspace-based analysis of the low scope interpretation of (24), shown in (30).

(30) $$\left\langle \begin{array}{c} [\{Sharon, \{\text{T}, \ldots \{\text{Voice}, \{make, \{the, error\}\}\}\}\}], \\ [\{Sharon, \{\text{T}, \ldots \{\text{Voice}, \{deliberately\}\}\}\}], \end{array} \right\rangle$$

Here we can say that *deliberately* and the VP are in the same position, as they are both the complement of Voice in their respective workspaces. Such a representation, however, raises three obvious questions:

1. How is (30) interpreted?
2. How is (30) pronounced?
3. How is (30) derived?

I address these three questions in turn directly.

3.1.1 How is (30) interpreted?

The derivation stage in (30) contains two workspaces, each of which contains a finite clause. I will assume that the interpretation of each clause contains an event description and a specification of how the event described relates to the context of utterance. For the sake of clarity, I will consider only the event-description portion of the meaning.

So the event description contained in the first workspace—the one associated with the host— is given in (31), and the event description contained in the second workspace—the one associated with the adjunct—is given in (32).

(31) $(make(e)\,\&\,\textsc{Agent}(e)(\textbf{sharon})\,\&\,\textsc{Theme}(e)(\textbf{the-error}))$

(32) $(\textsc{Agent}(e)(\textbf{sharon})\,\&\,deliberately(e))$

If, as I conjectured in the first part of this section, (31) and (32) yields the conjunction of the two, and if we take the further simplifying step of eliminating redundant conjuncts, we get the correct interpretation in (33).

(33) $(make(e)\ \&\ \textsc{Agent}(e)(\textbf{sharon})\ \&\ \textsc{Theme}(e)(\textbf{the-error})\ \&\ deliberately(e))$

Whether or not there is some process for eliminating redundant conjuncts instantiated in our cognitive faculties is not clear. That's more, it is not obvious how we could test for such a process. Assuming that redundant conjuncts are eliminated in the final interpretations of expressions like (24), however, will save space in this paper and reduce the amount of typing on my part, so I will do so going forward.

More could be said, of course, about the interpretation of (30), but I will leave this as a task for further research and move on to the question of pronunciation

3.1.2 How is (30) pronounced?

The problem posed for pronunciation by (30) is that the adjunct workspace contains most of a clause which is not pronounced. That is, *Sharon*, T, Voice, *etc.* must be deleted somehow. Recall from section 2.3 that the basic rule of deletion is that if a syntactic object contains two constituents, α and β, such that $\alpha = \beta$ and α asymmetrically c-commands β, then β is deleted.

The notion of identity here, must capture copies, but not repetitions, so in order for the various phrases and heads to be deleted from the adjunct we must show that they can be treated as copies of the corresponding phrases and heads in the host. Since the distinction between copies and repetitions is to follow from the derivational history of an expression, I will postpone the question of identity until the following section and stipulate, for the moment, that *Sharon*, T, Voice, *etc.* in the adjunct are considered copies of their counterparts in the host.

As for the c-command requirement for deletion, it is quite plain that it cannot apply to the deletion of copies in different workspaces as in (30). Since the c-command relation is dependant on Merge, the domain of which is limited to the workspace, it cannot hold across workspaces. However, if we broaden the c-command requirement on deletion to one of a more general ordering ($\alpha > \beta$) then it can apply to elements in separate workspaces, since workspaces in a derivation are ordered with respect to each other.

This broadening of the c-command requirement may seem *ad hoc* on its face, but there is a good reason to think that an operation like deletion is not sensitive specifically to c-command. That reason is that, as decades of research suggest, the syntactic component is the only component of the language faculty that is particular to the language faculty. It follows from this that deletion, an operation of the externalization system, is not particular to language. Since it is not particular to language, it should not be defined in language-particular terms. Therefore, defining deletion in terms of ordering as opposed to c-command is theoretically preferred.

So, turning back to the task at hand, (30) is pronounced by deleting all the redundant structure in the adjunct. This occurs because every element of the deleted structure is identical to an element in the host and ordered with respect to that matching element.

3.1.3 How is (30) derived?

The derivation of host-adjunct structures such as (30) can be divided into to parts. In the first part, the two workspaces—host and adjunct—are derived independently of each other, and in the second part, the workspaces are derived in lockstep. So, for instance, merging Asp_{perf} to the root of the host objects is accompanied by merging Asp_{perf} to the root of the adjunct object, and so on. The first part represents the standardly assumed operation of workspaces, and is, therefore, already understood, at least insofar as workspaces are understood. The second part—the part involving lockstep derivation—is novel and its explanation will occupy this section.

The result of the first part of the derivation is given in (34) below.

(34) $$\left\langle \begin{array}{l} [\{make, \{the, error\}\}, \text{Voice}, \ldots, T]_{\text{WS1}}, \\ [\{deliberately\}, \text{Voice}, \ldots, T]_{\text{WS2}}, [Sharon]_{\text{WS3}} \end{array} \right\rangle$$

Let's suppose that nothing forces the workspaces to derive in lockstep, but rather they derive freely and only result in a host-adjunct structure if their respective derivations mirror each other. This, however, would lead to two problems.

The first problem this poses has to do with the copy/repetition distinction. The externalization system, by hypothesis, deletes copies, not repetitions. Recall that T, Voice, the subject, *etc.* of the adjunct workspace delete in this case. This deletion would only occur if those objects and their counterparts in the host object were copies of each other and, while the necessary and sufficient conditions on copy-hood are not well understood, there is good reason to believe that content-identity is not sufficient. That is, two instances of, say, $\text{Voice}_{\text{Act}}$ are not copies just because they have identical content—it seem they must have an identical derivational history. This could not possibly hold of Voice, T, *etc* if the second stage of the derivation under discussion proceeds freely.

The second problem has to do with the subject *Sharon*. In (30), *Sharon* is in both workspaces, yet this does not seem possible if the each workspace's is derivatio is fully independent of the other's. Suppose we reach a stage of the derivation as shown in (35) where the next step must be to incorporate *Sharon* into WS1 and WS2 and merge it as the Agent.

(35) $$\left\langle \begin{array}{l} [\{\text{Voice}, \{make, \{the, error\}\}\}, \ldots, T]_{\text{WS1}}, \\ [\{\text{Voice}, \{deliberately\}\}, \ldots, T]_{\text{WS2}}, [Sharon]_{\text{WS3}} \end{array} \right\rangle$$

If we were to incorporate *Sharon* into WS1, as shown in (36), it would be rendered inaccessible to WS2, and vice-versa.

(36) $$\left\langle \begin{array}{l} [\{\text{Voice}, \{make, \{the, error\}\}\}, \ldots, T, Sharon]_{\text{WS1}}, \\ [\{\text{Voice}, \{deliberately\}\}, \ldots, T]_{\text{WS2}} \end{array} \right\rangle$$

Thus, there would no longer be any way to derive the two workspaces in lockstep. While this problem seems to be distinct from that of the copy/repetition problem above, it has the same solution—defining MERGE such that it lockstep derivation can be forced. I turn to such a definition presently.

Formal definitions of MERGE As discussed in section 2.2, Chomsky (2020) argues that the standard conception of Merge—$\text{Merge}(\alpha, \beta) \rightarrow \{\alpha, \beta\}$—needs to be replaced with a new one, called MERGE, which meets a number of desiderata. One such desideratum is that MERGE should be defined in terms of workspaces, rather than syntactic objects. In order to do this we must first provide some definitions for workspaces and other derivational notions. These definitions are given in (37)-(39).

(37) A derivation D is a finite sequence of stages $\langle S_1, S_2, \ldots, S_n \rangle$, where $D(i) = S_i$.

(38) A stage S is a finite sequence of workspaces $\langle WS_1, WS_2, \ldots WS_n \rangle$, where $S(i) = WS_i$.

(39) A workspace WS is a finite sequence of syntactic objects $\langle SO_1, SO_2, \ldots SO_n \rangle$, where $WS(i) = SO_i$.

In addition to the workspace desideratum, MERGE should also "restrict computational resources" (Chomsky 2020), by ensuring that when a new object is created by MERGE, its constituent parts do not remain accessible in the workspace. That is, MERGE substitutes the new object for the old objects. The definition of MERGE in (40), where "+" represents an "append" operation and "−" represents a "delete" operation, meets the two desiderata that I have mentioned thus far.[6]

(40) Where ω is a workspace, and α and β are syntactic objects,

$$\text{MERGE}_3(\omega, \alpha, \beta) \rightarrow \begin{cases} \{\alpha, \beta\} + ((\omega - \alpha) - \beta) & \text{if } \alpha \text{ and } \beta \text{ are in } \omega \\ \{\alpha, \beta\} + (\omega - \alpha) & \text{if } \alpha \text{ is in } \omega \text{ and } \beta \text{ is in } \alpha \\ \text{undefined} & \text{otherwise} \end{cases}$$

This definition, however, seems to over-generate. Consider the derivation in (41)

(41) $WS = \langle P, Q, X, Y \rangle$ (P, Q, X, and Y are lexical item tokens)

a. $\text{MERGE}_3(\text{WS, P, Q}) \rightarrow \langle \{P, Q, X, Y \rangle (= WS')$

b. $\text{MERGE}_3(\text{WS}', X, Y) \rightarrow \langle \{P, Q\}, \{X, Y\} \rangle (= WS'')$

[6]The astute reader will likely note that my definition of MERGE sacrifices the simplicity of Merge to meet the Chomsky's desiderata. This, I believe, reflects the fact that we lack a sufficient model of neural computation in which to ground our grammatical theory. Such a model would likely meet the "restrict resources" desideratum automatically.

If such a derivation were possible within a single workspace, then we could derive an entire clause—including complex nominal arguments—within a single workspace. This would, at best, render workspaces redundant, perhaps making the grammar indeterminate—any sentence would be derivable in at least two distinct ways.

The situation gets worse when we consider the fact that the definition of merge in (40) stipulates the distinction between internal and external merge. By hypothesis, though, the two cases of merge should fall out from a single definition of merge. Without the stipulation, it's likely that unrestricted parallel merge (Citko 2005) or sideward merge (Nunes 2004) would be derivable in this system. As discussed in section 2.2, though, once such varieties of merge are allowed, there is virtually no restriction on what can be derived. Thus, a definition of merge like that in (40) would likely over-generate.

This issue can be overcome in a non-stipulative way by eliminating one of the syntactic-object arguments from the definition of merge and defining merge as in (42).

(42) Where ω is a workspace, and α is a syntactic object,

$$\text{MERGE}_2(\omega,\alpha) \rightarrow \begin{cases} \{\alpha,\omega(1)\} + ((\omega-\alpha)-\omega(1)) & \text{if } \alpha \text{ is in } \omega \\ \{\alpha,\omega(1)\} + (\omega-\omega(1)) & \text{if } \alpha \text{ is in } \omega(1) \\ \text{undefined} & \text{otherwise} \end{cases}$$

I have restricted merge here by identifying a privileged member of a given workspace—the first member $\omega(1)$. This is what is sometimes referred to as the root of the tree. This is a justifiable step in that the first member of a workspace has a unique property among workspace members—the existence of a workspace depends only on the existence of its first member. That is, there are workspaces of length 1, 2, 3, *etc* but no workspaces of length 0. A corollary of this is that the proposition in (43) is only true for $i = 1$.

(43) For every workspace ω, $\omega(i)$ is defined.

By restricting merge in this way, we can rule out the derivation in (41). All instances of MERGE_2 modify WS(1). WS″(1) and WS′(1) in (41) are identical. Therefore No instance of

MERGE_2 could derive WS″ from WS′.

Being a computational procedure, MERGE ought to proceed in steps. Therefore, it should be a curried (or schönfinkeled) function. So, MERGE would be defined as in (44), with $\mathcal{M}$ standing in for the intension of MERGE (*i.e.*, the right side of the arrow in (40)).

(44) $\text{MERGE} = (\lambda\omega.(\lambda\alpha.\mathcal{M}))$

Curried functions are a variety of higher-order functions because they have functions as outputs in contrast first-order functions whose inputs and outputs are strictly non-functional. Under this version of MERGE a step of external merge is divided into two steps as in (45).

(45) a. $\text{MERGE(W)} \rightarrow \text{MERGE}^{\text{W}}$

b. $\text{MERGE}^{\text{W}}\text{(X)} \rightarrow \text{MERGE}^{\text{W,X}} \rightarrow \{\text{X, W(1)}\} + ((\text{W} - \text{X}) - \text{W(1)})$

Note here that, since lambda abstraction and reduction is sensitive only to the form of the variables, the order of these steps, dictated by the order of lambda expressions in (44), is arbitrary. We could, in principle, reorder the lambda expressions in (44) and we would have a different order of operations in (45) with the same result. This fact will come into play shortly.

The `map` function In the previous section I noted that curried functions are a class of higher-order functions because they have functions as outputs. In this section I will introduce a higher-order function that takes functions as inputs—the `map` function—which will be key to achieving lockstep parallel derivations. Informally speaking, `map` takes a function and applies it to a list of arguments. Formally, `map` is defined in (46).

(46) $\texttt{map}(f, \langle x_0, x_1, \ldots x_n \rangle) \rightarrow \langle f(x_0), f(x_1), \ldots f(x_n) \rangle$

Now, lets consider how lockstep parallel derivations would proceed. The stage at which the lockstep derivation begins was given in (34) and repeated here as (47).

(47) $\left\langle \begin{array}{l} [\{make, \{the, error\}\}, \text{Voice}, \ldots, T]_{\text{WS1}}, \\ [\{deliberately\}, \text{Voice}, \ldots, T]_{\text{WS2}}, [Sharon]_{\text{WS3}} \end{array} \right\rangle$

The next step is to merge Voice in WS1 and WS2 and to do that we start with MERGE curried in the reverse order of (44), shown in (48), with and α and ω ranging over SOs and workspaces, respectively.[7]

(48) R-MERGE = $(\lambda\alpha.(\lambda\omega.\mathcal{M}))$

Our first step, then, is to apply R-MERGE to Voice as in (49)

(49) R-MERGE(Voice) $\rightarrow$ R-MERGE$^{\text{Voice}}$

Next we `map` this function to WS1 and WS2 as in (50).

(50) $\texttt{map}(\text{R-MERGE}^{\text{Voice}}, \langle\text{WS1, WS2}\rangle) \rightarrow \left\langle \begin{matrix} [\{\text{Voice}, \{\textit{make}, \{\textit{the}, \textit{error}\}\}\}, \ldots, T], \\ [\{\text{Voice}\{\textit{deliberately}\}\}, \ldots, T] \end{matrix} \right\rangle$

And so on like that for the remainder of the derivation MAP-ing a curried MERGE to sequences of workspaces. Thus we can derive (30).

Identity across workspaces If (49) and (50) are two steps in the derivation of (30), we still need to explain how the the two instances of Voice can be considered copies of each other in order to explain how one of them deletes.

I mentioned in section 3.1.2 that, under a derivational theory of syntax, copies can be distinguished from repetitions in that the former share a derivational history, while the latter do not. In order for two objects to share a derivational history, they must have the same origin. The origin of any syntactic object in a given derivation is a tokening operation (Select in terms of Collins and Stabler (2016)) in the case of lexical item tokens or a subderivation in the case of derived objects like complex nominals.

In the case of Voice, since it a lexical item token, it's two instances in (30) must be linked by a single instance of the tokening operation Select, defined in (51).

(51) $\text{Select}(\alpha, \omega) \rightarrow \omega + \alpha$

Where α is a lexical item and ω is a workspace

[7]Note, though, that R-MERGE is not a newly proposed operation. It has the same intension as MERGE—represented as $\mathcal{M}$—with inverted lambda terms. Both R-MERGE, and MERGE, then, are derived from $\mathcal{M}$ by currying.

Of course, this operation can be curried as in (52) and mapped so that a single instance of Select can put a single token in two workspaces as in (53).

(52) $(\lambda\alpha.(\lambda\omega.\omega + \alpha))$

(53) a. Select(Voice) $\rightarrow$ Select$^{\text{Voice}}$

b. Map(Select$^{\text{Voice}}$, $\langle$WS1, WS2$\rangle$) $\rightarrow$ $\langle$WS1+Voice, WS2+Voice$\rangle$

So, the two instances of Voice share a single tokening operation, and therefore are the same object.[8]

4 Corroborating Evidence

In this section, I will outline a few problems related to adjunction that the proposed theory provides natural solutions to. First, I will address the island-hood of adjuncts. Then, I will discuss parasitic gaps, whereby adjunct island-effects are ameliorated. Finally, I will discuss a class of facts commonly associated with Cartographic/Nanosyntactic approaches to syntax—adjunct ordering constraints.

4.1 The Island-hood of adjuncts

A well-known property of adjuncts is that they are islands to movement. Indeed, Bošković (To Appear) points out that, while the island-hood of many other constructions varies across languages, adjunct island-hood seems to be constant.[9] So, for instance (54) is an ungrammatical question, and (55) is contains an ungrammatical relative clause because they both require an instance of *wh*-movement out of an adjunct.

(54) *What$_i$ did she eat an apple [after washing __$_i$]?

(55) *The student who$_i$ he invited Barbara [without meeting __$_i$]

[8]This also seems to be how we identify individual objects in general: I am the same individual as I was last year because both versions of me share the same birth event—the same origin.

[9]Bošković notes that, since the Coordinated Structure Constraint is also constant across languages, it should be unified with adjunct island-hood.

To see how the theory of adjuncts I propose here predicts adjunct island-hood consider the stage of the derivation of (54) immediately before *wh*-movement occurs. As shown in (56), the *wh*-expression *what* is in the adjunct workspace (WS2), which "scopes over" the TP. Note that both workspaces contain a C_{wh} head.

(56) $$\left\langle \begin{array}{l} [\{\mathrm{C}_{wh}, \{she, \{\mathrm{T}, \ldots\}\}\}]_{\mathrm{WS1}}, \\ [\{\mathrm{C}_{wh}, \{after, \{washing, what\}\}\}]_{\mathrm{WS2}} \end{array} \right\rangle$$

In order to derive (54), we would need a *wh*-movement operation such as (57).

(57) MERGE(WS1)(*what*)

The result of this operation, however, is undefined because *what* is neither a member of WS1, nor contained in the root object of WS1.

The operation in (58), on the other hand, is defined and would yield the stage in (59).

(58) MERGE(WS2)(*what*)

(59) $$\left\langle \begin{array}{l} [\{\mathrm{C}_{wh}, \{she, \{\mathrm{T}, \ldots\}\}\}]_{\mathrm{WS1}}, \\ [\{what\{\mathrm{C}_{wh}, \{after, \{washing, what\}\}\}\}]_{\mathrm{WS2}} \end{array} \right\rangle$$

This stage is problematic for two reasons. First, the C_{wh} head in WS1 would bear an unsatisfied *wh*-feature which would lead to a crash at the CI interface. Second, (59) would not yield (54) when linearized because *what*, being in WS2 would ordered after all of the words in WS1. That is, we would expect (59) to be linearized as (60).

(60) *She ate an apple what after washing

Thus the island-hood of adjuncts follows naturally from my proposed theory of adjuncts.

4.2 Parasitic Gaps

The island-hood of adjuncts, though constant across languages, is circumvented in so-called parasitic gap constructions (Engdahl 1983) as in (61) and (62).[10]

[10] I represent the gaps within the adjuncts here as $\{ec\}$s because, depending on the analysis, they are alternately identified as traces of movement or null proforms.

(61) What$_{\text{i}}$ did she eat __$_{\text{i}}$ [after washing ec_{i}]?

(62) The student who he invited __ [without meeting ec_{i}]

Here the parasitic gaps in the adjuncts, represented here as *ec*s, are licensed if there is a parallel trace in the host. This required parallelism is both syntactic—the trace and the parasitic gap have the same grammatical role (*i.e.* direct object in (61) and (62))—and semantic—the trace and parasitic gap co-refer.

Here, the mechanism for ensuring lockstep derivation—higher-order functions—allows us to derive parasitic gaps. To demonstrate this, consider the penultimate stage in the derivation of (61) shown in (63).

(63) $$\left\langle \begin{array}{l} [\{\mathrm{C}_{wh}, \{she, \{\mathrm{T}, \{\ldots, what_i\}\}\}\}]_{\mathrm{WS1}}, \\ [\{\mathrm{C}_{wh}, \{after, \{washing, what_i\}\}\}]_{\mathrm{WS2}} \end{array} \right\rangle$$

Note that the two instances of *what* here are copies of each other, meaning they share a derivational origin. The final stage of (61), given in (65) is derived in two steps given in (64).

(64) a. R-MERGE($what_{\text{i}}$) $\rightarrow$ R-MERGE$^{what_{\text{i}}}$

b. `map`(R-MERGE$^{what_{\text{i}}}$, $\langle$WS1, WS2$\rangle$) $\rightarrow$ (65)

(65) $$\left\langle \begin{array}{l} [\{what_i\{\mathrm{C}_{wh}, \{she, \{\mathrm{T}, \{\ldots, what_i\}\}\}\}\}]_{\mathrm{WS1}}, \\ [\{what_i\{\mathrm{C}_{wh}, \{after, \{washing, what_i\}\}\}\}]_{\mathrm{WS2}} \end{array} \right\rangle$$

As discussed in section 3.1.2, all instances of $what_{\text{i}}$ except for the highest instance in the first workspace is deleted, yielding the string (61).

Thus parasitic gaps are naturally accounted for in the theory I propose here.

4.3 Cartography's facts

There are well-known restrictions on the ordering of adjectives—for instance an ordering of size adjectives before shape adjectives, as in (66), is preferred to the reverse order, as in (67).[11]

[11] See Sproat and Shih (1991) for further discussion of the adjective ordering restriction

(66) a small square table

(67) ?*a square small table

Facts such as these are explained within the cartographic/nanosyntactic framework (see Cinque and Rizzi 2010) with two related hypotheses. The first hypothesis is that there is a universal fixed hierachy of functional heads such as Size and Shape. The second hypothesis is that adjuncts are merged as specifiers of their appropriate functional heads.[12] So, If Size and Shape select *small* and *square* as their respective specifiers, and Size selects ShapeP as its complement, then (66) can be derived, but (67) cannot.

While the first hypothesis is compatible with the workspaces-based theory of adjuncts, the second directly contradicts it. A workspace-theoretic approach can, however, provide a different explanation, given a few auxiliary hypotheses.

To begin, I give the derivation of (66)—a nominal phrase with an acceptable adjective sequence—in (68), followed by the derivation of (67)—a nominal phrase with a deviant adjective sequence— in (69).[13]

---

[12]See Ernst (2014) for a discussion of this hypothesis, which he refers to as th "F-Spec" hypothesis.

[13]I leave out Select operations for the sake of brevity.

(68)

(Start)		$\left\langle \begin{array}{l} [\{\mathit{small}\}, \textsc{Size}]_{WS1}, \\ [\{\mathit{square}\}, \textsc{Size}, \textsc{Shape}]_{WS2}, \\ [\sqrt{\textsc{table}}, n, \textsc{Size}, \textsc{Shape}]_{WS3} \end{array} \right\rangle$	0
R-MERGE(n)(WS3)	$\rightarrow$	$\left\langle \begin{array}{l} [\{\mathit{small}\}, \textsc{Size}]_{WS1}, \\ [\{\mathit{square}\}, \textsc{Size}, \textsc{Shape}]_{WS2}, \\ [\{\sqrt{\textsc{table}}, n\}, \textsc{Size}, \textsc{Shape}]_{WS3} \end{array} \right\rangle$	1
$\textsc{map}$(R-MERGE($\textsc{Shape}$))($\langle$WS2,WS3$\rangle$)	$\rightarrow$	$\left\langle \begin{array}{l} [\{\mathit{small}\}, \textsc{Size}]_{WS1}, \\ [\{\textsc{Shape}, \mathit{square}\}, \textsc{Size}]_{WS2}, \\ [\{\textsc{Shape}, \{n, \sqrt{\textsc{table}}\}\}, \textsc{Size}]_{WS3} \end{array} \right\rangle$	2
$\textsc{map}$(R-MERGE($\textsc{Size}$))($\langle$WS1,WS2,WS3$\rangle$)	$\rightarrow$	$\left\langle \begin{array}{l} [\{\textsc{Size}, \mathit{small}\}]_{WS1}, \\ [\{\textsc{Size}, \{\textsc{Shape}, \mathit{square}\}\}]_{WS2}, \\ [\{\textsc{Size}, \{\textsc{Shape}, \{n, \sqrt{\textsc{table}}\}\}\}]_{WS3} \end{array} \right\rangle$	3

(69)

(Start)		$\left\langle \begin{array}{l} [\{\mathit{square}\}, \textsc{Size}, \textsc{Shape}]_{WS1}, \\ [\{\mathit{small}\}, \textsc{Size}]_{WS2}, \\ [\sqrt{\textsc{table}}, n, \textsc{Size}, \textsc{Shape}]_{WS3} \end{array} \right\rangle$	0
R-MERGE(n)(WS3)	$\rightarrow$	$\left\langle \begin{array}{l} [\{\mathit{square}\}, \textsc{Size}, \textsc{Shape}]_{WS1}, \\ [\{\mathit{small}\}, \textsc{Size}]_{WS2}, \\ [\{\sqrt{\textsc{table}}, n\}, \textsc{Size}, \textsc{Shape}]_{WS3} \end{array} \right\rangle$	1
$\textsc{map}$(R-MERGE($\textsc{Shape}$))($\langle$WS1,WS3$\rangle$)	$\rightarrow$	$\left\langle \begin{array}{l} [\{\textsc{Shape}, \mathit{square}\}, \textsc{Size}]_{WS1}, \\ [\{\mathit{small}\}, \textsc{Size}]_{WS2}, \\ [\{\textsc{Shape}, \{n, \sqrt{\textsc{table}}\}\}, \textsc{Size}]_{WS3} \end{array} \right\rangle$	2
$\textsc{map}$(R-MERGE($\textsc{Size}$))($\langle$WS1,WS2,WS3$\rangle$)	$\rightarrow$	$\left\langle \begin{array}{l} [\{\textsc{Size}, \{\textsc{Shape}, \mathit{square}\}\}]_{WS1}, \\ [\{\textsc{Size}, \mathit{small}\}]_{WS2}, \\ [\{\textsc{Size}, \{\textsc{Shape}, \{n, \sqrt{\textsc{table}}\}\}\}]_{WS3} \end{array} \right\rangle$	3

The key point of comparison here is between respective second steps, in which SHAPE is merged. In (68), this step MAPs R-MERGE(SHAPE) to a contiguous sub-sequence of the active workspaces. In (69), on the other hand, this step MAPs the same curried function to a non-contiguous sub-sequence. If we make the auxiliary hypothesis that MAPping over a contiguous sequence is more computationally efficient than MAPping over a non-contiguous sequence, then we have a possible explanation of the deviance of (67) and, by extension, a possible explanation of adjunct ordering restrictions. That is, violations of adjunct ordering restrictions, rather than being violations of selection restrictions, are the result of suboptimal derivations.

Under the present approach, adjectives still merge with their respective functional heads, but as complements. That is, the structural relation between functional heads, like SIZE, and modifiers, like *small*, is the same as the relation between roots and their categorizing heads. It follows from this that modifiers merged with the interpretive relation between functional head and modifier should be the same as the one between categorizing heads and roots. This prediction is borne out in the intuitive understanding of polysemy.

Consider, for instance, how one would define the word *work*. Since it is polysemous we would have to give a list of definitions—we would say "*work* as a noun means ..." followed by "*work* as a verb means ...", or vice versa. We could formalize these as in (70).

(70) a. $\text{SEM}(\{n, \sqrt{\text{WORK}}\}) = \ldots$

b. $\text{SEM}(\{v, \sqrt{\text{WORK}}\}) = \ldots$

Now compare this to the adjective *light* which is many ways polysemous. Our list of definitions would be as follows—"*light* as a colour adjective means ...", "*light* as a weight adjective means ...", "*light* as an evaluative adjective means ...", and so on. Again, we can formalize these as in (71).

(71) a. $\text{SEM}(\{\text{COLOUR}, \textit{light}\}) = \ldots$

b. $\text{SEM}(\{\text{WEIGHT}, \textit{light}\}) = \ldots$

c. $\text{SEM}(\{\text{VALUE}, \textit{light}\}) = \ldots$

In both cases, we replace the *as-a* relation with the head-complement relation. If such a move were made in isolation, it would would be quite innocuous, even trivial. In the current context, though, the move was a logical result of a substantive hypothesis and should, therefore, be seen as corroborating evidence in favour of that hypothesis.

5 Apparent Counterexamples

Any worthwhile scientific theory should make empirical predictions. The preceding section discusses some of the correct empirical predictions of the theory that I have proposed. An honest assessment of the history of science, however, would show that most new theories make several wrong empirical predictions.[14] In this section I will discuss three apparently faulty predictions of my theoretical proposal.

The first such prediction is that host elements cannot c-command any adjunct elements unless they are also adjunct elements. There are many instances, though, in which a pronoun in the host clause is able to bind, and therefore c-command, an R-expression in an adjunct. The second is that, according to my proposal, a host and adjunct do not form a constituent. Many standard constituency tests, though, suggest otherwise. Finally, my proposal predicts that all adjuncts are islands, though there are certain classes of apparent adjuncts which allow *wh*-extraction from them.

In the remainder of this section I will discuss each of these in turn.

5.1 Adjuncts and Principle C

An anonymous reviewer notes that despite my proposal's predictions to the contrary, there is evidence that elements in the host of a sentence can c-command into an adjunct. The evidence that they gave was in the form of the principle C violation in (72).

(72) $\text{He}_{i/*j}$ asked which picture that John_j liked Mary bought.

[14]Feyerabend (1993) goes farther, arguing that *every* successful theory began its life unable to account for all of the phenomena that its predecessors accounted for. See also Piattelli-Palmarini, Uriagereka and Salaburu (2009, pp. 35–36) for discussion of early empirical falsification of special relativity.

Other than the island constraints, there is perhaps no greater source of data that informs theorizing about adjuncts than binding principle C. Unlike the data from island constraints—which is rather uniform—the data from principle C is varied and rather muddy.

Lebeaux (1988), for instance showed that fronted phrases that contained adjuncts showed antireconstruction effects with respect to principle C. Compare the sentences in (73) and (74).

(73) a. * He_i destroyed those pictures of John_i.

b. * He_i destroyed those pictures near John_i.

(74) a. * Which pictures of John_i did he_i destroy?

b. Which pictures near John_i did he_i destroy?

The ungrammatical sentences in (73) show that *he* is able to bind into both an argument (as in (73a)) and an adjunct (as in (73b)). Their counterparts in (74), however, show that binding survives *wh*-movement for the argument case (74a), but not the adjunct case (74b). Lebeaux uses this as evidence for his claim that adjuncts are added late. In modern terms, Lebeaux would propose that in (74a), there is a copy of *John* in the c-command domain of *he*, whereas in (74b) *John* only exists in the fronted *wh*-phrase.

Based on this data, we could propose the generalization in (75).

(75) **Lebeaux's Generalization**

If A is adjoined to X, and Y c-commands X, then Y c-commands A and its contents, unless A has been fronted.

Speas (1990, pp. 51–52), however, presents data that confounds such a generalization, showing that some types of adjuncts trigger principle C violations even when fronted.

(76) Temporal location vs. locative

a. In Ben_i's office, he_i is an absolute dictator.

b. * In Ben_i's office, he_i lay on his desk.

(77) Rationale vs. benefactive

a. For Mary$_i$'s valor, she$_i$ was awarded a purple heart.

b. * For Mary$_i$'s brother, she$_i$ was given some old clothes.

(78) Temporal vs. locative

a. On Rosa$_i$'s birthday, she$_i$ took it easy.

b. * On Rosa$_i$'s lawn, she$_i$ took it easy.

(79) Temporal vs. instrumental

a. With John$_i$'s novel finished, he$_i$ began to write a book of poetry.

b. * With John$_i$'s computer, he$_i$ began to write a book of poetry.

So, there are cases in which host-elements seem to c-command into adjuncts and there are cases where they do not.

Faced with such a situation, an theorist of adjuncts has two options, neither of which is good. Either they construct a theory in which the c-command into adjuncts is predicted to be the norm or they construct a theory in which c-command into adjuncts is barred as the norm. In either case the theorist will have exceptions when it comes to the principle C data presented here.

Beyond the muddiness of the principle C data, I would be remiss if I didn't note two of its shortcomings as a source of theoretically useful data. First is the fact that we currently lack a proper theory of binding within the biolinguistic/minimalist theory. Hornstein (2009, pp. 20–25) proposes a theory of principles A and B, but stops short of discussing principle C in detail. Second, there is some evidence that principle C binding is not entirely based on c-command. Compare the sentences in (80).

(80) a. * His$_i$ mother loves himself$_i$.

b. His$_{i/j}$ mother loves him$_i$.

c. His$_{i/*j}$ mother loves John$_j$.

The principle A violation in (80a) and the lack of principle B violations in (80b), taken together, suggest that the prossessive pronoun *his* does not c-command the direct object (*himself*/*him*). The principle C violation in (80c), however, sugguest that *his* does indeed c-command the direct object *John*.

It is possible, then, that further development of the proposed theory of adjuncts in tandem with a theory of binding could eventually yeild a theory in which all the data adduced in this section is accounted for. It is also possible that these facts are natuarlly accounted for by another theory of adjuncts. Since there is no current candidate for this other theory of adjuncts, I will leave the datapoints in this section as fodder for future research.

5.2 Adjuncts and Constituency tests

If adjuncts are completely separate objects from their hosts, as this paper proposes, then host and adjunct together should not form a constituent. An anonymous reviewer, however, points out that if a sentence like (1) undergoes VP-fronting, the adverbial adjunct is fronted along with the VP host as in (81).

(81) Sing the song with gusto, Rosie did.

This seems to indicate, contra my proposal, that *sing the song with gusto* is a constituent. There is however, an alternative explanation once one considers the fuller theory of grammar which my proposal is embedded in.

The first hint at this explantaion is that the thing that moves in VP-fronting is likely a phase which, according to Chomsky (2013), means it has undergone labeling. Consider, then, the structure of the fronted "VP" which undergoes labeling in (82).

(82) $\left\langle [\{\text{Voice}, \{sing, \{the, song\}\}\}]_{WS1}, [\{\text{Voice}, \{with, gusto\}\}]_{WS2} \right\rangle$

The labeling algorithm of Chomsky (2013) does a minimal search and returns the most prominent element of an object as its label. In the case of both the host in WS1 and the adjunct in WS2, the label will be Voice. What's more, by hypothesis, the Voice head in the host and the one in the adjunct are copies of each other, which means the respective labels of the object will be copies of each other.

Now, turning to the actual process of VP-fronting, let's hypothesize that, when possible, syntactic operations refer to labels, rather than whole objects. This, I believe, is a reasonable hypothesis, because searching for a single atomic element is likely more efficient than searching for a complex object. This gain in efficiency, though, comes at a cost of precision. Consider, the stage of the penultimate stage of the derivation of (81), shown in (83).

(83) $\langle [\{\mathrm{C}, \{\mathrm{T}, \{\ldots\}\}\}]_{WS1} \rangle$

The VP-fronting step will be one of internally MERGE-ing Voice, as in (84)

(84) MERGE(WS1)(Voice)

Since the host and the adjunct are both labeled by the same Voice head, they will both be targeted by this MERGE operation and therefore they will be fronted together.

Note that this explanation predicts that VP-fronting always fronts any VP adjuncts along with their hosts. This prediction does seem to be borne out as shown by the fact that the VP host cannot be fronted on its own as in (85)

(85) * Sing the song Rosie did with gusto.

Note that other constituency tests, which likely do not involve an actual movement operation, are able to target the host, the adjunct, and both together.

(86) a. It was sing the song with gusto that Rosie did.
b. It was sing the song that Rosie did with gusto.
c. It was with gusto that Rosie sang the song.

(87) We expected Rosie to sing the song with gusto, and ...
a. she did so.
b. she did so with gusto.
c. she sang the song so.

There is, no doubt much more to be said about this data, and its implications for the interpretation of constituency tests. I will leave that discussion for future research, noting only that the data in question does not seem to rule out a workspace-based theory of adjuncts.

5.3 Non-Island Adjuncts

I argued in section 4.1 that my theory of adjuncts predicts their islandhood. Several commentors, though, note that this prediction is contradicted by cases in which adjuncts seem not to be islands to movement. In particular, they point to the cases investigated by Truswell (2011), such as those in (88).

(88) a. What did you come round [to work on __]?

b. Who did John get upset [after talking to __]?

c. What did John come back [thinking about __]? (Truswell 2011, p. 129)

Truswell (2011) argues that extraction out of adjuncts is governed by what he dubs the Single Event Grouping Condition, given in (89), with auxiliary defintions in (90) and (91).

(89) **The Single Event Grouping Condition** (Truswell 2011, p. 157)

An instance of wh-movement is legitimate only if the minimal constituent containing the head and the foot of the chain can be construed as describing a single *event grouping*.

(90) An event grouping $\mathcal{E}$ is a set of core events and/or extended events $\{e_1, \ldots e_n\}$ such that:

a. Every two events $e_1, e_2 \in \mathcal{E}$ overlap spatiotemporally;

b. A maximum of one (maximal) event $e \in \mathcal{E}$ is agentive. (Truswell 2011, p. 157)

(91) An event e is agentive iff:

a. e is an atomic event, and one of the participants in e is an agent;

b. e consists of subevents $e_1, \ldots e_n$, and one of the participants in the initial subevent e_1 is an agent. (Truswell 2011, p. 158)

If the possibility of *wh*-extraction is governed by purely semantic considerations, as Truswell suggests, then theories,such as the one proposed in this paper, which derive island-hood on purely syntactic grounds are wrong-headed. There are, however, a few theoretical flaws in Truswell's proposal that seriously hamper its adequacy as a purely semantic account.

The first flaw, perhaps a minor one, is in the definition of an *agentive event* in (91). The first condition in that definition requires that agentive events be *atomic* events, while the second allows for that atomic event to consist of multiple subevents. By definition, however, atoms are not divisible, so this is a contradiction in terms. Perhaps this can be fixed, but the second flaw is a deeper one.

The second flaw is that the very notion of an event is not well enough defined to form the basis of a theory of *wh*-extraction. The condition in (89) requires that event groupings be countable—some expressions describe one event grouping while others must describe mutiple event groupings—and therefore they must be discrete in some way. That discreteness cannot come from the extra-mental world, where phenomena are continuous, a conclusion with which Truswell seems to concur, and therefore must have some cognitive source. While Truswell discusses a wide variety of data regarding event individuation, he does not present a theory of it. The closest he comes is the proposal that event (or event groupings) can have at most one agent, and Fodor's Generalization, given in (92).

(92) .**Fodor's Generalization** (Truswell 2011, p. 49)
A single verb phrase describes a single event.

These two claims, however, seem to be in tension when we consider (93) and the event it describes.

(93) Susan sold Geri a book.

Intuitively, this sentence discribes a single event, and Fodor's Generalization would back that up, however, it seems to describe an event with two agents. In order for a event to be an event of selling, there must be two active, intentional, willing, participants (*i.e.*, Agents) enacting the event. If one of those participants is not an Agent, then the event becomes one of theft, or foisting-upon, or the like. And, contra (89)-(91), *wh*-movement is allowed in a sentence like (93) as shown in (94).

(94) What did Susan sell Geri?

Truswell, then, is unable to provide a semantic basis for event individuation.

It is more plausible that event individuation is governed by syntactic principles such as (92). If this is the case, then even if Truswell's analysis is correct, *wh*-movement is governed by syntactic principles. It follows from this that, if the non-island adjuncts represented in (88) form a class, then that class must be defined syntactically. In fact, if we compare the examples in (95)-(98) to those in (1)-(4) we see that so-called rationale aduncts, which are not islands (see (88)), are decidedly less free than, say manner and temporal adverbials.

(95) Zoe came around the cafe to work on her novel.

(96) Zoe came around the cafe.

(97) Zoe came around the cafe to work on her novel to impress the cute barista.

(98) Zoe came around the cafe to impress the cute Barista to work on her novel.

While all of these are grammatical, the hosts and adjuncts are not independent of each other as they are in (1)-crefex:DinnerGusto and as my theory predicts they would be. In (97), for instance, impressing the barista depends of working on the novel, while in (98), the reverse is the case.

So, my proposed theory of adjuncts can be maintained against Truswell's data, by making one of two theoretical moves. We could divide adjuncts into *free adjuncts* and *restricted adjuncts* and limit the scope of my theory to the former, or we could make the stronger claim that the so-called adjuncts that Truswell (2011) is concerned with are not truly adjuncts and therefore not within the scope of my theory. I see no reason not to make the latter move.

6 Conclusion

I have argued in this paper that the basic facts about adjuncts only make sense if we assume that adjuncts are not truly attached to their hosts. While previous theories of grammar have not offered any way of formalizing this assertion, I proposed that the relatively new notion of workspaces offers such a possibility. That is, I proposed that adjuncts, like arguments, are derived in their own workspaces, but, unlike arguments, they are not incorporated into the "main" workspace. I formalized this proposal and, in the process, proposed a workspace-based formalization of MERGE. I then applied this formalized proposal to some generalizations related to adjunct—Islands, Parasitic Gaps, and adjective ordering constraints—showing that those generalizations are either predicted by my proposal or consistent with it.

Before concluding, though, I would like to discuss some possible implications of some of my proposals—specifically, the introduction of higher-order functions. My proposal makes crucial use of the higher-order function `map`, and this suggests an obvious minimalist criticism—namely that I have introduced unnecessary complexity to the grammar. Put concisely: If adding Pair-Merge to the grammar is illegitimate, then why isn't the addition of `map`? I will propose and discuss two possible answers to this challenge. First, I will discuss the possibility that higher-order functions like `map` are derivable from MERGE—that they "come for free". Second, I will discuss the possibility that it is these higher-order functions, rather than MERGE, which are the fundamental basis of language.

The idea that one could derive higher-order functions from MERGE begins with the suggestion—made frequently by Chomsky[15]—that internal MERGE is sufficient to explain the human faculty of arithmetic. The reasoning is as follows: The simplest case of Merge is vacuous internal Merge (Merge(x) $\rightarrow \{x\}$), which is identical to the set-theoretic definition of the successor function ($S(n) = n + 1$). Since the arithmetic is reducible to a notion of 0 or 1, the successor function and a few other axioms, Merge suffices to generate arithmetic.

[15] See Chomsky (2019, p. 274) for an instance in writing.

The process of learning arithmetic, then, is merely the process of setting the axioms of the system.

This result should not be surprising, though, since theoretical models of computation are closely linked to arithmetic. In fact, early models of computation were largely models of arithmetic—where the set of determinable functions that could be represented in model X is the set of X-computable functions on the natural numbers. An assumption generally made, called the Church–Turing thesis, is that a general class of computable functions is identical to the class of functions computable by a Turing machine. So, if we assume that a Merge-based computation system is capable of general computation, then it should be capable of performing every computable function. Since higher-order functions are computable functions, then a Merge-based system should allow for them.

This reasoning hinges on a few hypotheses, but even if it could be done completely deductively, it would still face the serious problem that models of computation and related systems assume a strict distinction between operations and atoms. Take, for instance, the process of deductive reasoning, which derives statements from from statements following rules of inference. In this case our operations are the rules of inference and the atoms are the statements. As Carroll (1895) famously illustrated, it is very easy to blur the lines between a rule of inference—such as *modus ponens*, given in (99)—and the logical statement in (100), but doing so renders the system useless.

(99) $$\frac{P \rightarrow Q, P}{Q}$$

(100) $((P \rightarrow Q)\&P) \rightarrow Q$

The former is a rule of inference that may or may not be active in a logical system, while the latter is a statement which may or may not be true in a logical system. If a system doesn't explicitly include (99) but can effectively perform it, we can say that the system in question can *simulate* (99). If a system can prove (100) without it being an axiom, then we can say that the system *generates* (100).

In the grammatical system that I have been assuming, MERGE corresponds to the rules of inference, and the syntactic objects and workspaces correspond to the atoms. In my reasoning above, I concluded that a MERGE-based system could simulate higher-order functions like `map`, but it cannot be concluded from this that `map` could be an integral part of adjunction. The human mind is capable of simulating wide variety of systems. For instance, a skilled Python programmer is effectively able to simulate a Python interpreter, but such a simulation requires learning, practice and considerable mental effort. Adjunction, on the other hand, seems to be fully innate and mostly effortless.

The second possibility is to propose that higher-order functions, or some principle that allows for them, are the basis for language. That is, we accept the minimalist evolutionary proposal that a single mutation separates us from our non-linguistics ancestors, but we propose that instead of MERGE/Merge, the result of that mutation was higher-order functions. There are a number of issues of varying levels surmountability with this proposal which I discuss below.

The first issue is that, while Merge/MERGE is a single operation and, therefore, easily mappable to a single genetic change, higher-order functions are a class of functions, making the task of linking them to a single mutation non-trivial. However, if they do form a (natural) class of functions, then they must share some singular feature, which can be mapped to a single mutation. The definition of a higher-order function as one that takes or gives a function as an input or output, respectively, suggests a such a feature—abstraction.

If abstraction is to be the defining feature of the faculty of language, then it behooves us to give a concrete definition of it. In the mathematico-computational sense, abstraction can be seen as the ability of system to treat functions as data. Applied to our cognitive system, this seems to allow meta-thinking—thinking about thinking, reasoning about reasoning, reflecting upon reflections, and so on, what Hofstadter (1979) calls "jumping out of the system." This kind of meta-thinking, though, is commonly associated with consciousness, which leads to two problems with this approach. The first problem is the hard problem of consciousness—if abstraction and consciousness are the same, then we may never fully understand either. The second problem is more mundane—We are no more conscious of adjunction than we are of MERGE, yet my reasoning here suggests that perhaps we should be conscious of the former.

There is however, a third possibility—a synthesis of the two previous possibilities. The early results of computability theory (Gödel 1931; Turing 1936) made crucial use of abstraction—using, say, number theory to reason about the axioms and operations of number theory. In fact, every simple model of computation allows for abstraction of the sort I am considering here.[16] This seems to suggest that the choice between the two possibilities above is a false one—that MERGE and abstraction cannot truly be disentangled. This does not allow us to avoid the problems that I have raised, though, but it does suggest that they can be combined and perhaps be solved together.

References

Bošković, Željko (To Appear). 'On Unifying the Coordinate Structure Constraint and the Adjunct Condition'. In: *Rethinking Grammar (Festschrift for Ian Roberts)*. Ed. by A. Bárány et al. Forthcoming.

Carroll, Lewis (1895). 'What the tortoise said to Achilles'. In: *Mind* 4.14, pp. 278–280.

Chametzky, Robert (1996). *A theory of phrase markers and the extended base*. SUNY Press.

Chomsky, Noam (1957). *Syntactic Structures*. The Hague/Paris: Mouton.

— (1965). *Aspects of the theory of syntax*. Cambridge: MIT Press.

— (1995). *The minimalist program.*

— (2000). 'Minimalist inquiries: The framework'. In: *Step by step: Essays on minimalist syntax in honor of Howard Lasnik*, pp. 89–155.

— (2013). 'Problems of projection'. In: *Lingua* 130, pp. 33–49.

[16] The abstraction feature of simple models of computation seems to allow self-reference, which inevitably leads to paradoxes. Such paradoxes are eliminated by complicating the models with type systems or arbitrary restrictions on abstraction.

Chomsky, Noam (2019). ‘Some Puzzling Foundational Issues: The Reading Program’. In: *Catalan Journal of Linguistics*, pp. 263–285.

— (2020). ‘The UCLA Lectures’. URL: https://ling.auf.net/lingbuzz/005485.

Cinque, Guglielmo and Luigi Rizzi (2010). ‘The cartography of syntactic structures’. In: *Oxford Handbook of linguistic analysis*. Ed. by Bernd Heine and Heiko Narrog. Oxford University Press.

Citko, Barbara (2005). ‘On the nature of merge: External merge, internal merge, and parallel merge’. In: *Linguistic Inquiry* 36.4, pp. 475–496.

Collins, Christopher (2017). ‘Merge (X, Y)={X, Y}’. In: *Labels and Roots*. Ed. by Leah Bauke, Andreas Blümel and Erich Groat. De Gruyter Mouton, pp. 47–68.

Collins, Christopher and Edward Stabler (2016). ‘A Formalization of Minimalist Syntax’. In: *Syntax* 19.1, pp. 43–78. DOI: 10.1111/synt.12117. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/synt.12117. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/synt.12117.

Engdahl, Elisabet (1983). ‘Parasitic gaps’. In: *Linguistics and philosophy*, pp. 5–34.

Ernst, Thomas (2014). ‘Adverbial adjuncts in Mandarin Chinese’. In: *The handbook of Chinese linguistics*. Wiley Online Library, pp. 49–72.

Feyerabend, Paul (1993). *Against method*. Verso.

Gödel, Kurt (1931). ‘Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I’. In: *Monatshefte für mathematik und physik* 38.1, pp. 173–198.

Heim, I. and A. Kratzer (1998). *Semantics in Generative Grammar*. Blackwell Textbooks in Linguistics. Wiley. ISBN: 9780631197133. URL: https://books.google.ca/books?id=jAvR2DB3pPIC.

Heim, Irene (1982). ‘The Semantics of Definite and Indefinite Noun Phrases’. Doctoral dissertation. UMass Amherst.

Hofstadter, Douglas R (1979). *Gödel, Escher, Bach: an eternal golden braid*. Vol. 13. New York: Basic Books.

Hornstein, Norbert (2009). *A theory of syntax: minimal operations and universal grammar.* Cambridge: Cambridge University Press.

Kayne, R.S. (1994). *The Antisymmetry of Syntax*. Linguistic inquiry monographs. Cambridge, MA: MIT Press. ISBN: 9780262611077.

Kratzer, Angelika (1996). 'Severing the external argument from its verb'. In: *Phrase structure and the lexicon*. Springer, pp. 109–137.

Lebeaux, David (1988). 'Language Acquisition and the Form of the Grammar'. In: *Doctoral dissertation, University of Massachusetts*.

Lu, Jiayi, Cynthia K Thompson and Masaya Yoshida (2020). 'Chinese wh-in-situ and islands: A formal judgment study'. In: *Linguistic Inquiry* 51.3, pp. 611–623.

May, Robert Carlen (1978). 'The grammar of quantification.' Doctoral dissertation. Massachusetts Institute of Technology.

Nunes, Jairo (2004). *Linearization of chains and sideward movement*. Vol. 43. MIT press.

Piattelli-Palmarini, Massimo, Juan Uriagereka and Pello Salaburu, eds. (2009). *Of Minds and Language: A Dialogue with Noam Chomsky in the Basque Country*. Oxford University Press.

Speas, M.J. (1990). *Phrase Structure in Natural Language*. Studies in Natural Language and Linguistic Theory. Springer Netherlands.

Sproat, Richard and Chilin Shih (1991). 'The cross-linguistic distribution of adjective ordering restrictions'. In: *Interdisciplinary approaches to language*. Springer, pp. 565–593.

Stepanov, Arthur (2001). 'Late Adjunction and Minimalist Phrase Structure'. In: *Syntax* 4.2, pp. 94–125. DOI: `https://doi.org/10.1111/1467-9612.00038`.

Trinh, Tue (2009). 'A constraint on copy deletion'. In: *Theoretical Linguistics* 35.2-3, pp. 183–227.

Truswell, Robert (2011). *Events, Phrases, and Questions*. Oxford: Oxford University Press.

Turing, Alan Mathison (1936). 'On computable numbers, with an application to the Entscheidungsproblem'. In: *J. of Math* 58.345-363, p. 5.