A parallel derivation theory of adjuncts

Daniel Milway

dan@milway.ca

1st June 2021

Abstract

I present and argue for a theory of adjuncts according to which, adjuncts and their respective hosts are derived as separate, parallel objects that are not combined until forced to by the process of linearization. I formalize the notion of the workspace, and the workspace-based operation MERGE (Chomsky 2020). Finally, I show that this approach to adjuncts naturally accounts for Adjunct Islands and Parasitic Gaps and is consistent with adjective ordering constraints

[*NOTE: This is an updated version of a paper previously posted to LingBuzz (`https://ling.auf.net/lingbuzz/005281`). The present paper bears a different title and makes a slightly different claim.*]

1 Introduction

Adjuncts and the process of adjunction which produces them 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 such route.

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. This conjecture, it should be noted, is not completely novel. Indeed, Chomsky (1965) conjectured that ...

> [many Adverbials] are Sentence transforms with deleted Subjects. Thus underlying the sentence "John gave the lecture with great enthusiasm," with the Adverbial "with great enthusiasm" is the base string "John has great enthusiasm" ... with the repeated NP "John" deleted as is usual[.] (218 f.)

Similarly, Lebeaux (1988, p. 151) proposes the operation Adjoin-α, which "is simply an operation joining phrase-markers." 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.

The goal of this paper is to propose a theoretical explanation of grammatical adjuncts and adjunction. I will begin in section 2 with some remarks on the empirical scope of my proposal. I continue in section 3, by laying out my relevant theoretical assumptions with special reference to Simplest Merge (Collins 2017) and workspaces (Chomsky 2020). Next, I make my proposal explicit in section 4, 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 5 and some facts that seem to contradict my theory in section 6. Finally, I conclude, discussing the implications of my proposal on the broader theory of grammar in section 7.

2 What is this paper about?

As I mention above, this paper proposes a theory of adjuncts and each reader likely has their own particular rough and ready pretheoretic or quasitheoretic notion of what an adjunct is. More than likely this notion is based on a prototype of adverbs, adjective, prepositional phrases, or the union of all of these categories—indeed perhaps all of my examples of adjuncts will take the form of adjectives, adverbs or prepositional phrases. This notion, no doubt, has furnished each reader with a battery of tests for any would-be theory of adjuncts—a bunch of facts that a theory of adjuncts must account for. Expecting or requiring that theoretical definition of some aspect of nature perfectly matches a pretheoretic notion of that aspect of nature is a fool's errand—the explanatory domain of a theory rarely, if ever, matches any pretheoretic notion, nor should we expect it to.

There are at least two reasons that we ought not to expect the domain of any theory to match our pretheoretic notion. The first is that the very rationale for theoretical investigation of some aspect of nature is our lack of pretheoretic understanding of that aspect of nature. The first step towards theoretical explanation of something, then, is the realization that our intuitive understanding of it is flawed. It is therefore inconsistent to require that implications of a pretheoretic notion be carried over to an explanatory theory.

The second reason is that, historically, the domains of explanatory theories are rarely if ever coextensive with the pretheoretic domain. In one sense, the process of theorizing narrows the domain but, in another sense, explanatory theories tend to have an unexpectedly broad domain. For instance, Generative Grammar doesn't address all of the phenomena covered by the commonsense term "language," but the theory has also been used to provide explanations for aspects of the human faculty of music (Mukherji 2012) and arithmetic (Chomsky 2020). Similarly, pre-Galilean (*i.e.,* Aristotelian) mechanics covered all variety of earthly motion and change, including plant growth, but excluded the motion of the stars and planets, which belonged to the separate field of cosmology (Feyerabend 1993). So, requiring a theory to meet our pretheoretic expectations may preclude theories with surprising explanatory depth.

The case of adjuncts and adjunction, though, is complicated by the fact that, broadly speaking, the current understanding of them is not exactly pretheoretic. As I discuss in the following section, the tem "adjunct" had a precise theoretical meaning in various versions of X-bar theory, but more broadly, the term refers to a possible class subexpressions which do not fit neatly into grammatical theory. In this sense, we could describe the ideas of adjuncts/adjunction as held by syntacticians to be *extra*theoretic.

Yet, so long as it is made somewhat explicit, a pretheoretic notion of any phenomenon is a crucial starting point for any theoretical work, the present one being no exception. So, as a pretheoretic notion, I take adjuncts to be parts of linguistic expressions which are optional, stackable, and freely-orderable, and adjunction to be the process by which adjuncts are introduced to an expression. Two important notes to make regarding this "definition" are that (*i*) it is a conjunction, not a disjunction, of three properties—every adjunct seems to have all three—and (*ii*) it is at best a heuristic device—my theory will take it to be the "base case" for adjuncts. As we shall see in section 6 though, much of the data that seems to contradict the theory I propose involves expressions that do not meet the heuristic definition of adjuncts/adjunction and therefore can be set aside.

3 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, followed by some comments on the other cognitive systems with which the language faculty interacts.

3.1 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

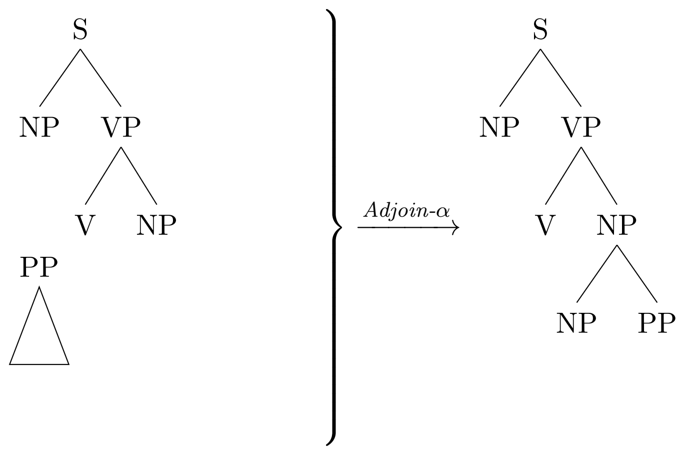
(Chomsky 2013; Hornstein 2009). Most of those theorists[1] have settled on the definion of Merge in (6), sometimes called "Simplest Merge".

(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)

(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.

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

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 (8)

(8) 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.

3.2 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 is to redefine Merge as an operation not on sntactic objects *per se* but on workspaces which contain syntactic objects. Following Chomsky, I will refer new version of Merge as MERGE (pronounced "capital merge"). I will propose formal definitions of workspaces and MERGE in section 4.1.3, but some properties of these constructs are worth mentioning here. The objects that we called stages of derivations—*e.g.*, (10)—are in fact workspaces. The distinction between the two terms—"stage" and "workspace"—is analogous to the distinction between the distinction between "theorem"/"lemma" and "formula" in proof theory—in both cases, the former are a subspecies of the latter that are demonstrably derivable by a system of axioms and rules. So, while we can arbitrarily construct any number of well-formed workspaces for our purposes, there is no guarantee that all of them will be derivable from the lexicon by the grammatical operations. The new operation, MERGE, operates on workspaces as sketched in (14) where *(i)* X and Y are syntactic objects, *(ii)* WS and WS' are workspaces, *(iii)* either X and Y are in WS or X is in WS and contains Y, and *(iv)* WS' contains {X, Y} but does not contain X or Y.

(14) MERGE(WS,X,Y) $\rightarrow$ WS'

Setting aside issues of formalization for the time being, the workspase-based theory proposed by Chomsky (2020) suggests a picture of syntax wherein the derivation of, say, (9) consists, in a sense, of three separate subderivations—one for each argument and one for the clausal spine—which ultimately converge to give a single clause.

3.3 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 3.1, 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 $\beta^\frown\alpha$. 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

[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.

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.

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 counterpart 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 (Heim and Kratzer 1998), event identification (Kratzer 1996), and existential closure (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 dollar bills 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 may sometimes be a useful way 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

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

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 "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.

3.4 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 am assuming that MERGE operates on a workspace by manipulating that workspace's contents. 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.

[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.

4 The proposal

The theory of adjuncts that I propose is best viewed in contrast to the theory of arguments. According to this theory, outlined in section 3.2, an argument is derived in a separately from its clausal spine, and the result of that derivation is merged into clausal spine derivation. An adjunct is also derived in a separately, except that the adung is never merged into the clausal spine derivation. So the syntactic representation of (1) is given in (21) with the adjunct-free sentence (2) derived as the first element of the workspace (SO1), and the adjunct PP *with gusto* derived as the second (SO2).

(21) $\langle \{Rosie, \{\mathrm{T}, \ldots \{sing, \{the, song\}\}\}\}_{SO1}, \{with, gusto\}_{SO2} \rangle$

The expression represented in (21) is grammatical insofar as SO1 is a grammatical clause and SO2 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 workspace.

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 (SO1 $\prec$ SO2), so no linearization problem should occur.

In what follows, I will refine this proposal somewhat, but the core claim—that adjuncts are derived separately and remain separate 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.

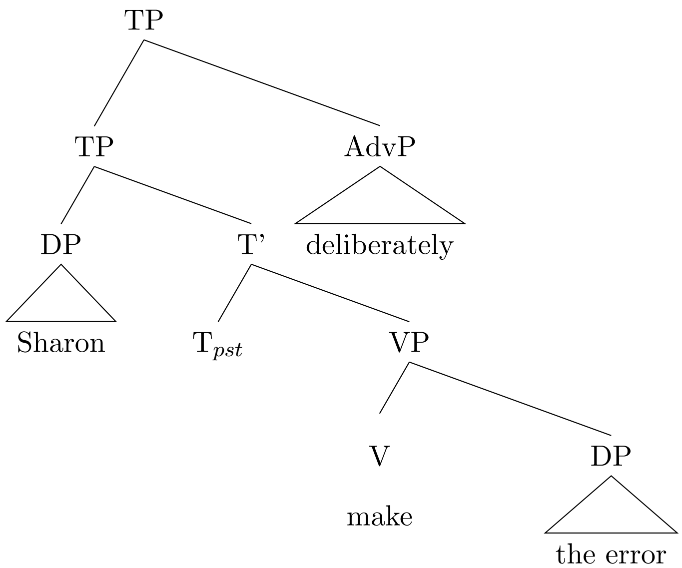
4.1 The problem of adjunct scope

The sentence in (24) is ambiguous.

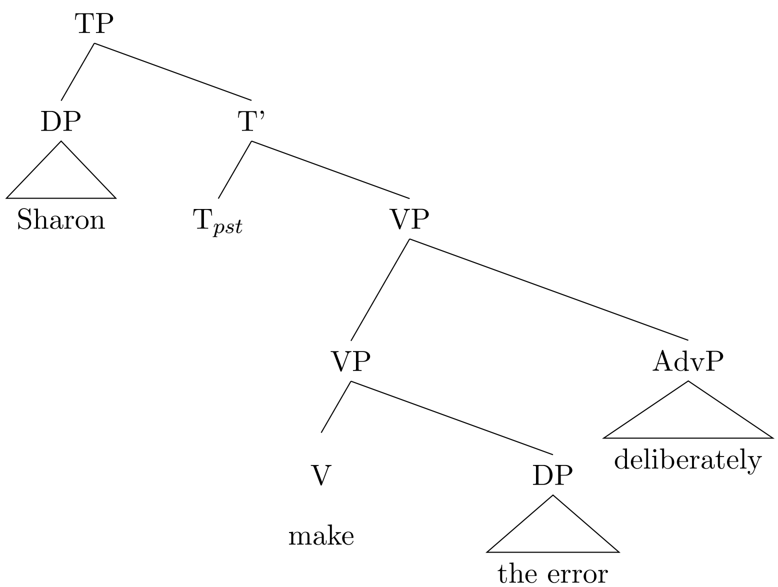
(24) Sharon made the error deliberately.

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 parallel derivation 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) $$\left\langle \begin{array}{c} \{Sharon, \{\mathrm{T}, \ldots \{\text{Voice}, \{make, \{the, error\}\}\}\}\}, \\ deliberately \end{array} \right\rangle$$

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 scopes over Voice.

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

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

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

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, escpecially noting that *Sharon* appears in both host and adjunct:

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

I address these three questions in turn directly.

4.1.1 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 object—the one associated with the clausal host— is given in (31), and the event description contained in the second object—the one associated with the adverbial adjunct—is given in (32).

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

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

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) $(make(e) \,\&\, \textsc{Agent}(e)(\mathbf{sharon}) \,\&\, \textsc{Theme}(e)(\mathbf{the\text{-}error}) \,\&\, deliberately(e))$

Whether or not there is some process for eliminating redundant conjuncts instantiated in our cognitive faculties is not clear. What'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

4.1.2 How is (30) pronounced?

The problem posed for pronunciation by (30) is that the adjunct contains most of a clause which is not pronounced. That is, *Sharon*, T, Voice, *etc.* must be deleted somehow. Recall from section 3.3 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.

4.1.3 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 objects—host and adjunct—are derived independently of each other, and in the second part, the objects are derived in lockstep. So, for instance, merging Asp_{perf} to the host object is accompanied by merging Asp_{perf} to 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) $$\left\langle \begin{array}{l} \{make, \{the, error\}\}_{\text{SO1}}, \\ \{deliberately\}_{\text{SO2}}, Sharon_{\text{SO3}} \end{array} \right\rangle$$

Let's suppose that nothing forces the objects 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 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 derived objects, yet this does not seem possible if the each object's is derivation 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) $$\left\langle \begin{array}{l} \{\text{Voice}, \{make, \{the, error\}\}\}_{\text{SO1}}, \\ \{\text{Voice}, \{deliberately\}\}_{\text{SO2}}, Sharon_{\text{SO3}} \end{array} \right\rangle$$

If we were to MERGE *Sharon* with SO1, as shown in (36), it would be rendered inaccessible to SO2, and vice-versa.

(36) $$\left\langle \begin{array}{l} \{Sharon, \{\text{Voice}, \{make, \{the, error\}\}\}\}_{\text{SO1}}, \\ \{\text{Voice}, \{deliberately\}\}_{\text{SO2}} \end{array} \right\rangle$$

Thus, there would no longer be any way to derive the two objects 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 3.2, Chomsky (2020) argues that the standard conception of Merge—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)-(38).

(37) A derivation D is a finite sequence of workspaces $\langle \text{WS}_1, \text{WS}_2, \ldots, \text{WS}_\text{n} \rangle$, where $\text{D(i)} = \text{WS}_\text{i}$.

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

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 (39), where "+" represents an "append" operation and "−" represents a "delete" operation, meets the two desiderata that I have mentioned thus far.[6]

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

$$\mathrm{MERGE_3}(\omega, \alpha, \beta) \rightarrow \begin{cases} \{\alpha, \beta\} + ((\omega - \alpha) - \beta) & \text{if } \alpha \text{ and } \beta \text{ are in } \omega \\ \{\alpha, \beta\} + (\omega - \alpha) & \text{if } \alpha \text{ is in } \omega \text{ and } \beta \text{ is in } \alpha \\ \text{undefined} & \text{otherwise} \end{cases}$$

Note that the definition of merge in (39) 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 3.2, though, once such varieties of merge are allowed, there is virtually no restriction on what can be derived.

Being a computational procedure, MERGE ought to proceed in steps. Therefore, it should be a curried (or schönfinkeled) function. [7] So, MERGE would be defined as in (40), with $\mathcal{M}$ standing in for the intension of MERGE (*i.e.*, the right side of the arrow in (39)).

(40) $\mathrm{MERGE} = (\lambda\omega.(\lambda\alpha.(\lambda\beta\mathcal{M})))$

[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.

[7]Dmitrii Zelenskii (P.C.) points out that the function defined in (39) can be considered monadic and, therefore, computational if we take it to be a function from a triple ⟨WS, X, Y⟩ to a workspace. While this is true, it would merely kick the can down the road a bit, as we would then need to define a function that creates the appropriate triple for $\mathrm{MERGE_3}$.

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 three steps as in (41).

(41) a. MERGE(W) $\rightarrow$ MERGE$^{\mathrm{W}}$

b. MERGE$^{\mathrm{W}}$(X) $\rightarrow$ MERGE$^{\mathrm{W,X}}$

c. MERGE$^{\mathrm{W,X}}$(Y) $\rightarrow$ {X, Y)} + ((W − X) − Y)

The notation f^x indicates the result of f being applied to x, and should be familiar from Formal Semantics, where $[\![\cdot]\!]^{g,w}$ indicates the interpretation function relative to assignment g and world w.

This definition of MERGE as a curried function also allows us to somewhat explain the accessibility restricions on MERGE in a less stipulative way. We can do so by first hypothesizing that each input to MERGE partially defines the domain of the resulting function. So, the domain of MERGE is the set of workspaces and, when MERGE is applied to a workspace W, it yields MERGE$^{\mathrm{W}}$ as in (41a). The domain of MERGE$^{\mathrm{W}}$, then, is the workspace W and, assuming X is a member of W, applying MERGE$^{\mathrm{W}}$ to X yields MERGE$^{\mathrm{W,X}}$ as in (41b). The domain of MERGE$^{\mathrm{W,X}}$, then, is something like the union of W and X. That is MERGE$^{\mathrm{W,X}}$ can apply to any member of W—yielding External MERGE–or any object contained in X—yielding Internal MERGE. Note that this requires a second hypothesis, namely that workspaces and syntactic objects have distinct notions of "membership", with workspace membership being something like set membership and syntactic object membership being a recursive membership—Y is contained in X iff Y∈X, or for some Z∈ X, Y is contained in Z. Making these two hypotheses, though, gives us some understanding of why MERGE would have Internal and External cases, but not Parallel or Sideward cases.

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 (42).

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

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 (43).

(43) $\left\langle \begin{array}{l} \{\mathit{make}, \{\mathit{the}, \mathit{error}\}\}_{\text{SO1}}, \\ \{\mathit{deliberately}\}_{\text{SO2}}, \mathit{Sharon}_{\text{SO3}} \end{array} \right\rangle$ (= WS1)

The next step is to select $\text{Voice}_{\text{ACT}}$ from the lexicon and merge it with in SO1 and SO2. Selection is acheived by a simple operation Select, which is defined in (44) as extending a workspace to include a token of a lexical item.[8]

(44) For workspace W and lexical item LI, Select(W)(LI) $\rightarrow$ W + LI

Selecting $\text{Voice}_{\text{ACT}}$ for the workspace in (43) proceeds as follows.

(45) $\text{Select(WS1)}(\text{Voice}_{\text{ACT}}) \rightarrow \left\langle \begin{array}{l} \{\mathit{make}, \{\mathit{the}, \mathit{error}\}\}_{\text{SO1}}, \{\mathit{deliberately}\}_{\text{SO2}}, \\ \mathit{Sharon}_{\text{SO3}}, \text{Voice}_{\text{ACT}\text{SO3}} \end{array} \right\rangle$ (= WS2)

Next, we can merge $\text{Voice}_{\text{ACT}}$ with SO1 and SO2 in three steps. First we apply MERGE to WS2, as shown in (46), and then apply the resulting function to $\text{Voice}_{\text{ACT}}$ as in (47)

(46) $\text{MERGE(WS2)} \rightarrow \text{MERGE}^{\text{WS2}}$

(47) $\text{MERGE}^{\text{WS2}}(\text{Voice}_{\text{ACT}}) \rightarrow \text{MERGE}^{\text{WS2,Voice}_{\text{ACT}}}$

The result of these two steps is a function which we can `map` to our host and adjunct objects as in (48).

(48) $\texttt{map}(\text{MERGE}^{\text{WS2,Voice}_{\text{ACT}}})(\langle\text{SO1, SO2}\rangle)$

The final result requires some discussion. By our definition of `map` in (42), the result should be a list of the individual applications of the function, as represented in (49).

[8] Strictly speaking, the token of LI on the righthand side of the arrow should be distinguishable from LI itself on the lefthand side. Collins and Stabler (2016), for instance, distinguish lexical items, which are members of the lexicon, from lexical item tokens which are pairs of lexical items and integers. For simplicity of exposition I will not formally distinguish tokens from proper lexical items.

(49) $\left\langle \mathrm{MERGE}^{\mathrm{WS2},\mathrm{Voice}_{\textsc{Act}}}(\mathrm{SO1}),\ \mathrm{MERGE}^{\mathrm{WS2},\mathrm{Voice}_{\textsc{Act}}}(\mathrm{SO2})\right\rangle$

This, though will be a list of workspaces, given in (50), which is not a legitimate object according to our formalism. What's more, this list of workspaces will be riddled with redundancy—SO1, SO2, and SO3 are contained in both workspaces, albeit within larger SOs in some cases, and in the same order.

(50) $$\left\langle \begin{array}{l} \langle\{\mathrm{Voice}_{\textsc{Act}}, \mathrm{SO1}\}, \mathrm{SO2}, \mathrm{SO3}\rangle, \\ \langle \mathrm{SO1}, \{\mathrm{Voice}_{\textsc{Act}}, \mathrm{SO2}\}, \mathrm{SO3}\rangle, \end{array} \right\rangle$$

This situation is a violation of the broad principle of Resource Restriction proposed by Chomsky (2020), which we can ameliorate with the help of two more specific constraints defined in (51) and (52).

(51) NoRedundancy

Delete all X in U such that there is some Y in U and Y contains X.

(52) ConserveInformation

For all operations f applied to U, yielding U', if relation R holds in U and is not explicitly altered by f, then R holds in U'

These constraints, though expressed in a way useful for a theory of language, seem to be ideal candidates for general cognitive principles. The former, for instance, seems to be active in our perceptual systems which only transmit a fraction of their input to the mind, discarding redundant data. The latter, on the other hand, ensures that cognitive processes can be markovian (*i.e.*, memoryless) without loss of information.

By applying these constraints to (50)—deleting the redundant objects, while maintaining the relative ordering of the remaining objects—we get the single workspace (53).

(53) $$\left\langle \begin{array}{l} \{\mathrm{Voice}_{\textsc{Act}}, \{\mathit{make}, \dots\}_{\mathrm{SO1}}\}_{\mathrm{SO1'}}, \\ \{\mathrm{Voice}_{\textsc{Act}}, \{\mathit{deliberately}\}_{\mathrm{SO2}}\}_{\mathrm{SO2'}}, \mathit{Sharon}_{\mathrm{SO3}} \end{array} \right\rangle (= \mathrm{WS3})$$

Note that, although the above discussion assumed that `map` created the object (50) which was then flattened to (53) by applying the Resource Restriction constrains, a more reasonable assumption is that the constraints apply not to representations, but to operations. So, just as MERGE is defined in (39) such that it removes redundant material, to too should `map` be assumed to respect NoRedundancy and ConserveInformation. Under this assumption, then, the operation in (48) directly generates (53) without the generating intermediate object (50).

The next step in our lockstep derivation is to merge the external argument *Sharon* with both host and adjunct. This step, shown in (54), will proceed much in the same way as the above-described step, but without the need to select anything from the lexicon.

(54) $\texttt{map}(\text{MERGE}(\text{WS3})(\textit{Sharon})) \rightarrow \left\langle \begin{array}{l} \{\textit{Sharon}, \{\text{Voice}_{\text{ACT}}, \{\textit{make}, \dots\}\}\}, \\ \{\textit{Sharon}, \{\text{Voice}_{\text{ACT}}, \{\textit{deliberately}\}\}\} \end{array} \right\rangle (= \text{WS4})$

The derivation will continue in this manner, selecting lexical items as needed and merging them with the two syntactic objects until we reach the point, represented in (55), at which the external argument *Sharon* must internally merge as the subject.

(55) $\left\langle \begin{array}{l} \{\text{T}, \dots \{\textit{Sharon}, \dots \{\textit{make}, \dots\}\}\}_{\text{SO1}}, \\ \{\text{T}, \dots \{\textit{Sharon}, \dots \{\textit{deliberately}\}\}\}_{\text{SO2}} \end{array} \right\rangle (= \text{WSN})$

Here we face a complication. Our first step is to apply MERGE to the workspace WSN yielding $\text{MERGE}^{\text{WSN}}$. Based on the pattern set up above, we might try to apply $\text{MERGE}^{\text{WSN}}$ to the external argument *Sharon*, however this is not a legitimate move, as the *Sharon* is not in WSN, which is the domain of $\text{MERGE}^{\text{WSN}}$. Instead, we `map` $\text{MERGE}^{\text{WSN}}$ to the two objects giving us a list of functions as shown in (56).

(56) $\texttt{map}(\text{MERGE}^{\text{WSN}})(\langle \text{SO1}, \text{SO2} \rangle) \rightarrow \langle \text{MERGE}^{\text{WSN,SO1}}, \text{MERGE}^{\text{WSN,SO2}} \rangle$

To complete this merge step, we need to apply each of these new functions to *Sharon*—we need an inverted `map` function, which applies a list of functions to a single input. We can construct such a function with another higher-order function—`apply`, defined in (57)—and lambda abstraction.

(57) $\texttt{apply}(f)(x) \rightarrow f(x)$

Our final step, then, is shown in (58).

(58) $\texttt{map}(\lambda f.\texttt{apply}(f)(Sharon))(\langle \text{MERGE}^{\text{WSN,SO1}}, \text{MERGE}^{\text{WSN,SO2}}\rangle)$

$\to \langle \text{MERGE}^{\text{WSN,SO1}}(Sharon), \text{MERGE}^{\text{WSN,SO2}}(Sharon)\rangle$

$\to \langle\{Sharon, \text{SO1}\}, \{Sharon, \text{SO2}\}\rangle = (30)$

5 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.

5.1 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 (59) is an ungrammatical question, and (60) is contains an ungrammatical relative clause because they both require an instance of *wh*-movement out of an adjunct.

(59) *What$_i$ did she eat an apple [after washing $\_\_{i}$]?

(60) *The student who$_i$ he invited Barbara [without meeting $\_\_{i}$]

To see how the theory of adjuncts I propose here predicts adjunct island-hood consider the stage of the derivation of (59) immediately before *wh*-movement occurs. As shown in (61), the *wh*-expression *what* is in the adjunct workspace (WS2), which "scopes over" the TP. Note that both workspaces contain a C_{wh} head.

(61) $$\left\langle \begin{array}{l} \{\text{C}_{wh}, \{she, \{\text{T}, \ldots\}\}\}_{\text{SO1}}, \\ \{\text{C}_{wh}, \{after, \{washing, what\}\}\}_{\text{SO2}} \end{array} \right\rangle (= \text{WS1})$$

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

In order to derive (59), we would need a *wh*-movement operation such as (62).

(62) MERGE(WS1)(SO1)(*what*)

The result of this operation, however, is undefined because *what* is neither a member of WS1, nor contained in SO1.

The operation in (63), on the other hand, is defined and would yield the stage in (64).

(63) MERGE(WS1)(SO2)(*what*)

(64) $$\left\langle \begin{array}{l} \{\mathrm{C}_{wh}, \{she, \{\mathrm{T}, \ldots\}\}\}_{\mathrm{SO1}}, \\ \{what\{\mathrm{C}_{wh}, \{after, \{washing, what\}\}\}\}_{\mathrm{SO2}'} \end{array} \right\rangle (= \mathrm{WS1}')$$

This stage is problematic for two reasons. First, the C_{wh} head in SO1 would bear an unsatisfied *wh*-feature which would lead to a crash at the CI interface. Second, (64) would not yield (59) when linearized because *what*, being in SO2′ would be ordered after all of the words in SO1. That is, we would expect (64) to be linearized as (65).

(65) *She ate an apple what after washing.

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

5.2 Parasitic Gaps

The island-hood of adjuncts, though constant across languages, is circumvented in so-called parasitic gap constructions (Engdahl 1983) as in (66) and (67).[10]

(66) What$_\mathrm{i}$ did she eat $\_\__\mathrm{i}$ [after washing *ec*$_\mathrm{i}$]?

(67) The student who he invited $\_\_$ [without meeting *ec*$_\mathrm{i}$]

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

[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.

and the parasitic gap have the same grammatical role (*i.e.* direct object in (66) and (67))—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 (66) shown in (68).

(68) $$\left\langle \begin{array}{l} \{\mathrm{C}_{wh}, \{she, \{\mathrm{T}, \{\ldots, what_i\}\}\}\}_{\mathrm{SO1}}, \\ \{\mathrm{C}_{wh}, \{after, \{washing, what_i\}\}\}_{\mathrm{SO2}} \end{array} (= \mathrm{WS1}) \right\rangle$$

Note that the two instances of *what* here are copies of each other, meaning they share a derivational origin. The final stage of (66), given in (70) is derived, as shown in (69), using the same pattern we used for parallel internal MERGE in (55) to (58).

(69) a. MERGE(WS1) → $\mathrm{MERGE}^{\mathrm{WS1}}$

b. `map`($\mathrm{MERGE}^{\mathrm{WS1}}$)(⟨SO1,SO2⟩) → ⟨$\mathrm{MERGE}^{\mathrm{WS1,SO1}}$, $\mathrm{MERGE}^{\mathrm{WS1,SO1}}$⟩

c. `map`(λf.`apply`(f)($what$))(⟨$\mathrm{MERGE}^{\mathrm{WS1,SO1}}$, $\mathrm{MERGE}^{\mathrm{WS1,SO1}}$⟩)→(70)

(70) $$\left\langle \begin{array}{l} \{what_i\{\mathrm{C}_{wh}, \{she, \{\mathrm{T}, \{\ldots, what_i\}\}\}\}\}_{\mathrm{SO1'}}, \\ \{what_i\{\mathrm{C}_{wh}, \{after, \{washing, what_i\}\}\}\}_{\mathrm{SO2'}} \end{array} \right\rangle$$

As discussed in section 4.1.2, all instances of $what_{\mathrm{i}}$ except for the highest instance in the first SO is deleted, yielding the string (66).

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

5.3 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 (71), is preferred to the reverse order, as in (72).[11]

(71) a small square table

(72) ?*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

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

there is a universal fixed hierachy of functional heads such as SIZE and SHAPE. The second hypothesis is that adjectives, adverbials, etc. 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 (71) can be derived, but (72) cannot.

Before outlining how the proposed theory of adjuncts might account for these fact, it is worth noting that there is no inherent contradiction between the cartographic/nanosyntactic theory of adjectives, adverbs, etc. and the theory of adjuncts being proposed here. As I stated above, the former theory explains (71) and (72) in part by saying that attributive adjectives are not adjuncts but specifiers, and this explanation can be extended to other similar ordering restrictions. The theory proposed here, though, is not a theory of adjective, adverbs, or prepositional phrases—it is a theory of adjuncts. Therefore, the proposition that attributive adjectives are specifiers, rather than adjuncts, merely implies that attributive adjectives are beyond the scope of my theory.

One might object to this by asserting that cartography/nanosyntax in fact makes a stronger claim—that all adjuncts are specifiers. Such a claim, though, is self-contradicting in the same way as a claim that all odd numbers are even would be. A coherent version of this claim is that there are no adjuncts, really—everything we thought was an adjunct is actually a specifier. Such a claim does not so much contradict the theory proposed here as render it empirically inert. The examples in (1) to (3), with which I began this paper, along with the classic example in (73), however, suggest that this strong version of cartography/nanosyntax cannot be maintained.

(73) the tall, tall, tall, ... tall building

The theory being proposed here, then, is compatible with a weak version of cartography/nanosyntax or, at least, a version of moderate strength. My goal in this section, however, is to extend the theory to explain some of the central facts cartogra-

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

phy/nanosyntax, thus putting the two theories in conflict with each other.

The extension of the theory involves two auxiliary hypotheses—the Universal Functional Sequence hypothesis, and the hypothesis that operations on non-contiguous segments of the workspace are more costly than those on contiguous segments.

The first hypothesis, which is lifted from cartography/nanosyntax, is that there is some universal ordered set of functional heads, and that the ordering of that set is reflected in the c-selection relation. So, the data in (71) and (72) is follows, at least partially, from the conjecture that SIZE can c-select SHAPE, but not vice versa. I diverge from the cartography/nanosyntax explanation, though, in that I don't argue that (72) involves SHAPE incorrectly c-selecting SIZE. Rather, the deviance of (72) comes from the fact that it requires an operation on a non-contiguous segment of the workspace, as I demonstrate below.

To begin, I give the derivation of (71)—a nominal phrase with an acceptable adjective sequence—in table 1, followed by the derivation of (72)—a nominal phrase with a deviant adjective sequence— in table 2.[13] Recall that the linear ordering of SOs in a workspace is hypothesized to be reflected in the linear ordering of their respective externalizations. So, in table 1 the fact that the SO based on the adjective *small* precedes the one based on the adjective *square* is reflected in the fact that, whin this workspace is pronounced, "small" preceded "square," and so on. The key point of comparison here is between respective second steps, in which SHAPE is merged. In table 1, this step MAPs $\text{MERGE}^{\text{WS2},\textsc{Shape}}$ to a contiguous segment of the workspace. In table 2, on the other hand, this step MAPs the same curried function to a non-contiguous segment. 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 (72) and, by extension, a possible explanation of adjunct ordering restrictions. That is, violations of adjunct ordering restrictions, rather than being violations of c-selection restrictions, are the result of suboptimal derivations.

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

(Start)		$\left\langle \begin{array}{l} \{small\}_{\text{SO1}}, \\ \{square\}_{\text{SO2}}, \\ \sqrt{\text{TABLE}}_{\text{SO3}}, n, \textsc{Size}, \textsc{Shape} \end{array} \right\rangle$	WS1
MERGE(WS1)(SO3)(n)	$\rightarrow$	$\left\langle \begin{array}{l} \{small\}_{\text{SO1}}, \\ \{square\}_{\text{SO2}}, \\ \{n, \sqrt{\text{TABLE}}\}_{\text{SO3}}, \textsc{Size}, \textsc{Shape} \end{array} \right\rangle$	WS2
MAP(MERGE(WS2)(SHAPE))(⟨SO2,SO3⟩)	$\rightarrow$	$\left\langle \begin{array}{l} \{small\}_{\text{SO1}}, \\ \{\textsc{Shape}, \{square\}\}_{\text{SO2}}, \\ \{\textsc{Shape}, \{n, \sqrt{\text{TABLE}}\}\}_{\text{SO3}}, \textsc{Size} \end{array} \right\rangle$	WS3
MAP(MERGE(WS3)(SIZE))(⟨SO1,SO2,SO3⟩)	$\rightarrow$	$\left\langle \begin{array}{l} \{\textsc{Size}, small\}_{\text{SO1}}, \\ \{\textsc{Size}, \{\textsc{Shape}, square\}\}_{\text{SO2}}, \\ \{\textsc{Size}, \{\textsc{Shape}, \{n, \sqrt{\text{TABLE}}\}\}\}_{\text{SO3}} \end{array} \right\rangle$	WS4

Table 1: The partial derivation of (71)

(Start)		$\left\langle \begin{array}{l} \{small\}_{\text{SO1}}, \\ \{square\}_{\text{SO2}}, \\ \sqrt{\text{TABLE}}_{\text{SO3}}, n, \textsc{Size}, \textsc{Shape} \end{array} \right\rangle$	WS1
MERGE(WS1)(n)(SO3)	$\rightarrow$	$\left\langle \begin{array}{l} \{small\}_{\text{SO1}}, \\ \{square\}_{\text{SO2}}, \\ \{n, \sqrt{\text{TABLE}}\}_{\text{SO3}}, \textsc{Size}, \textsc{Shape} \end{array} \right\rangle$	WS2
MAP(MERGE(WS1)(SHAPE))(⟨SO1,SO3⟩)	$\rightarrow$	$\left\langle \begin{array}{l} \{\textsc{Shape}, square\}_{\text{SO1}}, \\ \{small\}_{\text{SO2}}, \\ \{\textsc{Shape}, \{n, \sqrt{\text{TABLE}}\}\}_{\text{SO3}}, \textsc{Size} \end{array} \right\rangle$	WS3
MAP(MERGE(WS3)(SIZE))(⟨SO1,SO2,SO3⟩)	$\rightarrow$	$\left\langle \begin{array}{l} \{\textsc{Size}, \{\textsc{Shape}, square\}\}_{\text{SO1}}, \\ \{\textsc{Size}, small\}]_{\text{SO2}}, \\ \{\textsc{Size}, \{\textsc{Shape}, \{n, \sqrt{\text{TABLE}}\}\}\}]_{\text{SO3}} \end{array} \right\rangle$	WS4

Table 2: The partial derivation of (72)

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 (74).

(74) 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 (75).

(75) 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.

5.4 Concord vs Agreement

Among languages whose adjectives show φ-feature morphology, there is further division based on the contexts in which that morphology shows up. In French, for instance, φ-

morphology shows up on attributive adjectives—matching the φ features of their host noun—and predicative adjectives—matching the φ features of the subject—as shown in (76) and (77), respectively.

(76) a. la femme grand -e
the.FSg woman tall FSg
"The tall woman"

b. le garçon grand -∅
the.MSg boy tall MSg
"The tall boy"

c. les filles grand -es
the.Pl girls tall FPl
"The tall girls"

(77) a. La femme est grand -e.
the.FSg woman is tall FSg
"The woman is tall"

b. Le garçon est grand -∅.
the.MSg boy is tall MSg
"The boy is tall"

c. Les filles sont grand -es.
the.Pl girls are tall FPl
"The girls are tall"

In contrast, German adjectives show φ-features in attributive positions but not in predicative positions as shown in (78) and (79), respectively.

(78) a. keine groß-e Frau
no.FSgNom tall FSgNom woman
"no tall woman"

b. kein groß-er Jung
no.MSgNom tall MSgNom boy
"no tall bot"

c. keine groß-en Mädchen
no.NPlNom tall NPlNom girls
"no tall girls"

(79) a. Keine Frau ist groß.
no.FSgNom woman is tall

"No woman is tall."

b. Kein Jung ist groß.
no.MSgNom is tall
"No boy is tall."

c. Keine Mädchen sind groß.
no.NPlNom girls are tall
"No girls are tall."

Put in commonly-used descriptive language, both French and German adjectives undergo (nominal) concord—shown in (76) and (78)—while only French adjectives undergo (subject) agreement—shown in (77) and not in (79).

If we assume, following Milway (2019), that (*i*) adjective agreement comes from the same process as finite verb agreement, and (*ii*) French and German, for example, differ from each other in that the French adj^0 head bears unvalued φ-features, while the German one does not, then we can explain the facts demonstrated in (77) and (79). This, however, leaves the question of how concord happens, for which my proposed theory of adjuncts offers an answer.[14]

First, consider the simple German nominal phrase in (80) which is specified for Case, gender, and number.

(80) eine Brücke
a.FNom bridge
"a bridge"

Setting aside gender for now, we can assume that Case an number features are housed not on the noun itself, but on functional heads in the noun's extended projection—Case is on D, number on Num. Therefore, we can analyze (80) roughly as in (81).

(81) $\langle\{ein_{\text{F.Nom}}, \{\text{Num}_{\text{Sg}}, \{\textit{Brücke}\}\}\}\rangle$

Now, consider the nominal phrase (82) which has an adjunct that shows concord.

(82) eine klein -e Brücke
a.FNom small FSgNom bridge
"a small bridge"

[14]See Norris (2017a,b) for a full survey of the attempts to explain concord phenomena.

On the same assumptions as above, we can analyse (82) as in (83).

(83) $$\left\langle \begin{matrix} \{ein_{\text{F.Nom}}, \{\text{Num}_{\text{Sg}}, \{klein\}\}\} \\ \{ein_{\text{F.Nom}}, \{\text{Num}_{\text{Sg}}, \{Br\ddot{u}cke\}\}\} \end{matrix} \right\rangle$$

There is no need to get features from the noun to the adjective here, since the relevant features—F, SG, NOM—are in both workspaces by virtue of lockstep parallel derivation.

This is, of course, far from a full analysis of all concord phenomena,[15] but rather, a proof-of-concept—a demonstration that concord may be explained in this theory of adjuncts without recourse to complicated operations like Agree.

6 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.[16] 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 seems to be 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. Third, my proposal predicts that all adjuncts are islands, though there are certain classes of apparent adjuncts which allow *wh*-extraction from them. Finally, My proposal that adjuncts are separate objects from their hosts seems to clash with cases where adjuncts seem to undergo movement, such as *wh*-questions and topicalization.

[15]Even just within German, there are three sets of concord phenomena—strong, weak, and mixed—that need full analysis.

[16]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.

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

6.1 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 (84).

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

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 (85) and (86).

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

b. * He_i destroyed those pictures near John_i.

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

b. Which pictures near John_i did he_i destroy?

The ungrammatical sentences in (85) show that *he* is able to bind into both an argument (as in (85a)) and an adjunct (as in (85b)). Their counterparts in (86), however, show that binding survives *wh*-movement for the argument case (86a), but not the adjunct case (86b). Lebeaux uses this as evidence for his claim that adjuncts are added late. In modern terms, Lebeaux would propose that in (86a), there is a copy of *John* in the c-command domain of *he*, whereas in (86b) *John* only exists in the fronted *wh*-phrase.

Based on this data, we could propose the generalization in (87).

(87) **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.

(88) 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.

(89) 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.

(90) 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.

(91) 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, a 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 (92).

(92) 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 (92a) and the lack of principle B violations in (92b), taken together, suggest that the possessive pronoun *his* does not c-command the direct object (*himself*/*him*). The principle C violation in (92c), however, sugguest that *his* does indeed c-command the direct object *John*. Thus, Principle C data contradicts Principle A/B data.

It is possible, then, that further development of the proposed theory of adjuncts in tandem with a theory of binding could eventually yield a theory in which all the data adduced in this section is accounted for. It is also possible that these facts are naturally 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.

6.2 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 (93).

(93) 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 (94).

(94) $\left\langle \{\text{Voice}, \{\mathit{sing}, \{\mathit{the}, \mathit{song}\}\}\}_{\text{SO1}}, \{\text{Voice}, \{\mathit{with}, \mathit{gusto}\}\}_{\text{SO2}} \right\rangle$

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 SO1 and the adjunct SO2, 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.[17]

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 reasnoable 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 (93), shown in (95).

(95) $\langle \{\text{C}, \{\text{T}, \{\dots\}\}\}_{\text{SO1}} \rangle (= \text{WS1})$

The VP-fronting step will be one of internally MERGE-ing Voice, as in (96)

(96) MERGE(WS1)(SO1)(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 (97)

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

[17] An anonymous reviewer notes that this, in fact much of the proposed theory, depends on how the operation Transfer is formalized/defined—a task which I do not take up here. This is undoubtedly true—in fact, I would go further and say that virtually any theory of any aspect of syntax depends on a theory of Transfer, and that the development of such a theory is a project on its own. Collins and Stabler (2016) provide a formal definition of Transfer, for instance, but they are quick to point out the empirical flaws in their own definition.

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.

(98) 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.

(99) 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.

6.3 Non-Island Adjuncts

I argued in section 5.1 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 (100).

(100) 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 (101), with auxiliary defintions in (102) and (103).

(101) **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.*

(102) An event grouping $\mathcal{E}$ is a set of core events and/or extended events $\{e_1, \dots e_n\}$ 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)

(103) 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 $e_1, \dots e_n$, 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 (103). 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 (101) 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 (104).

(104) .**Fodor's Generalization** (Truswell 2011, p. 49 following Fodor 1970)
A single verb phrase describes a single event.

These two claims, however, seem to be in tension when we consider (105) and the event it describes.

(105) 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 (101)-(103), *wh*-movement is allowed in a sentence like (105) as shown in (106).

(106) 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 (104). 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 (100) form a class, then that class must be defined syntactically. In fact, if we compare the examples in (107)-(110) to those in (1) to (4) we see that so-called rationale aduncts, which are not islands (see (100)), are decidedly less free than, say manner and temporal adverbials.

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

(108) Zoe came around the cafe.

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

(110) 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 (109), for instance, impressing the barista depends of working on the novel, while in (110), 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.

It is worth noting here that Truswell does not seem to provide a working definition of adjunct, relying instead on the his readers' pretheoretic intuition about what counts as an adjunct. I suspect that if he had provided such a definition, he might come to the same conclusion as I do above—that those cases of non-island "adjunct" are not truly adjuncts at all. This, of course, leaves the question of what they actually are if not adjuncts—a question which I will not take up here.

6.4 Apparent adjunct movement

Consider the sentences (111) and (112), in which the apparent adjuncts, indicated by emphasis, appear to have undergone A'-movement.

(111) **With a bat**, she struck the ball.

(112) **How** did she sing the anthem?

If, as I propose, adjuncts are not part of the clauses that they are "adjoined to", how can they possibly undergo movement within them? This puzzling question, though, conceals a paradox which, in a sense, provides its answer—If topics and *wh*-expressions are in [SPEC, CP], how can they be adjuncts, which, according to the theory proposed here, are not part of their host clauses? The obvious way to answer these qustions is to propose that the expressions in question are externally merged in [SPEC, CP], meaning that they are not adjuncts, and they do not undergo movement.

This may seem like a rather bold claim, but I can think of no good theoretical argument against it. The reason that this claim might seem bold is likely because topicalization and *wh*-questions are commonly subsumed under the banner of "A'-movement", reflecting the GB/Early Minimalist separation of movement from Merge/base generation. With the discovery of Internal Merge as a sub-case of Merge, the notion of A'-movement—or any movement—as a unified phenomenon makes no sense.[18] The emphasized in expressions in (111) and (112) could be considered cases of A'-External-Merge.

7 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

[18] The fact that A'-movement seems to be an identifiable phenomenon without being theoretically identifiable is what makes it worth studying.

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[19]—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. 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 state-

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

ments 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 (113)—and the logical statement in (114), but doing so renders the system useless.

(113) $$\frac{P \to Q, P}{Q}$$

(114) $((P \to Q)\&P) \to Q$

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 (113) but can effectively perform it, we can say that the system in question can *simulate* (113). If a system can prove (114) without it being an axiom, then we can say that the system *generates* (114).

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. 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.

— (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.

Fodor, Jerry A (1970). ‘Three reasons for not deriving” kill” from” cause to die”’. In: *Linguistic Inquiry* 1.4, pp. 429–438.

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

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

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.

Milway, Daniel (2019). ‘Explaining the Resultative Parameter’. Doctoral dissertation. University of Toronto.

Mukherji, Nirmalangshu (2012). *The primacy of grammar*. MIT Press.

Norris, Mark (2017a). ‘Description and analyses of nominal concord (Pt I)’. In: *Language and Linguistics Compass* 11.11, e12266.

— (2017b). ‘Description and analyses of nominal concord (Pt II)’. In: *Language and Linguistics Compass* 11.11, e12267.

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.