THE UNIVERSITY OF CHICAGO

INTERMEDIATE CONSTITUENTS

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE HUMANITIES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF LINGUISTICS

BY
ZACHARY LEBOWSKI

CHICAGO, ILLINOIS
AUGUST 2025

For Frank

Some implications of our procedure need to be made explicit.

–Rulon S. Wells, "Immediate Constituents"

TABLE OF CONTENTS

ACKNOWLEDGMENTS

I would like to start by thanking my parents, who usually did their best to let me make decisions for myself, even when they felt I was making the wrong choice. I think they would agree that it mostly worked out. And thank you to my sister, who has been there for me when times were tough.

To the members of my committee, you each played a crucial part in this thesis. Karlos, I respect your commitment to your craft and to your students. Thank you for all the time combing through the details, but more importantly, thank you helping me to produce work of decent quality. The shortcomings are my own. Erik, your passion for syntax played an important role in my decision to go to grad school. I have appreciated your guidance and friendship over the years. Alan, thank you for being welcoming to me and for generally being a nice person to talk to and work with. Also, thank you for reading a syntax dissertation.

Thank you to the rest of the University of Chicago linguistics community. I've enjoyed getting to know the faculty, the staff, and the students over the last six years. I feel lucky to have been a part of a very stimulating and supportive intellectual community.

I would also like to extend my appreciation to the linguistics department at my undergraduate institution, the University of California, Santa Cruz. UCSC is where I discovered how much I loved linguistics. A special thanks goes out to Jorge Hankamer and Jim McCloskey, who were the first to tell me that I could be a linguist. I wouldn't have even considered the possibility without you two.

Thanks to all my friends (you know who are). Stay cool. I love you all.

Thank you to participants of the Morphology and Syntax Workshop, P-interest at Stanford, and LOOC 138 for your feedback on my work, and to anyone else who has helped along the way.

Thank you, Frank. I dedicate this to you for all the joy you bring me.

ABSTRACT

The prevailing view in minimalist syntactic theory is that syntactic representations are generated step-by-step over the course of a derivation. The steps of a derivation are (sets of) representations, and syntactic operations map (sets of) representations to other (sets of) representations. Much recent theoretical work has been narrowly focused on defining syntactic operations (Citko, 2005; Epstein et al., 2012; Collins and Stabler, 2016; Collins, 2017; Merchant, 2019; Ermolaeva and Kobele, 2023; Milway, 2021; Zyman, 2024). Furthermore, there has been a widespread adoption of several principles barring syntactic operations from changing their inputs in any way (Chomsky, 1993, 1995, 2000, 2007, 2008). This dissertation is focused on one of those principles—that is, the Extension Condition. Informally, the Extension Condition says that structure-building operations must apply to and extend uncontained syntactic objects (i.e. the roots of syntactic trees).

I propose a formalization of minimalist syntax that allows us to study the properties of syntactic structure-building operations. I find that conventional minimalist principles permit an infinite and varied set of operations. I pay particularly close attention to a family of operations that I call the Extension Condition Family, which is defined as the set of operations that (essentially) obey the Extension Condition (Chomsky, 1993, 1995, 2000). This set of operations is especially interesting because, if they are the only licit operations, a very clear (partial) ordering of operations follows. A consequence of restricting a theory to the use of Extension Condition Family members is that a syntactic object will be a subconstituent of the output of a derivation if and only if the subconstituent was a rooted syntactic object at some earlier stage in the derivation. As a consequence, every derivational theory of syntax that only permits the use of Extension Condition Family members has a strongly equivalent declarative (also known as representational) theory; when two theories are *strongly equivalent*, they generate the same set of abstract representations.

The objective of the thesis isn't to argue for declarative theory over a derivational theory,

nor do I intend on convince you to adopt any particular operation or set of operations. Instead, for one, I aim to convince you that every Extension Condition compliant syntactic theory has a strongly equivalent declarative alternative. Two, if your syntactic theory permits operations that violate the Extension Condition and you make use of those operations in an analysis, then some intermediate representations of your derivations will not be a constituent in the final representations. I call intermediate representations of this kind *intermediate constituents*. I hope to compel you to look for evidence of intermediate constituents because evidence of that kind can reveal something deep and interesting about the architecture of grammatical processes.

Finding evidence that distinguishes declarative and derivational alternatives is usually challenging. I review some uses of syntactic operations that violate the Extension Condition. In particular, I study Richards's tucking-in analysis of multiple wh-movement in Bulgarian (Richards, 1997, 1999, 2001), as well as Lebeaux's (1991) late adjunction analysis of anti-reconstruction effects. I show that, for each of the case studies, the analysis that makes use of operations outside the Extension Condition Family—that is, operations violate the Extension Condition. Theoretically, these analyses should make different predictions than an alternative that only uses Extension Condition Family members; however, I was unable to find any empirical evidence in support of one theory over the other.

The challenge extends to theories of the syntax-prosody interface as well. I provide an analysis of the interaction between intonation, syntax, and information structure in English. I show that, in some cases, conflicts between prosodic requirements are resolved with operations that are non-extending (i.e violate the Extension Condition). And more, I find that there is evidence for the intermediate constituents that are opacified by operation outside the Extension Condition Family.

CHAPTER 1

INTRODUCTION

The prevailing view in early minimalist syntactic theory is that syntactic representations are generated step-by-step over the course of a derivation.[1] The mapping from one representation to the next is governed by principles of derivational economy (Chomsky, 1993, 1995). Much of the earliest work in minimalist syntax investigated potential implementations of economy (Pesetsky, 1989; Chomsky, 1993, 1995; Kitahara, 1995; Collins, 1997; Epstein et al., 1998; Richards, 1999). The proposed economy principles were designed to derive conditions on transformations and filters on representations (i.e. the Cyclic Principle (Chomsky, 1995), the Strict Cycle Condition (Chomsky, 1973, 1977), the Empty Category Principle (Chomsky, 1981; Chomsky, 1993), and several others) that played important roles in minimalism's predecessor, the Principles and Parameters framework. Furthermore, until the notion of local economy became widely adopted (Collins, 1997), there was a strong tendency for economy principles to be stated as global trans-derivational constraints; the grammar chose the most economical of possible derivations, where being most economical meant containing the fewest derivational steps, the shortest movements, and/or some other property with a superlative in its name. However, much recent theoretical work has been narrowly focused on defining syntactic operations (Citko, 2005; Epstein et al., 2012; Collins and Stabler, 2016; Collins, 2017; Merchant, 2019; Ermolaeva and Kobele, 2023; Milway, 2023; Zyman, 2024); given that operations, like transformations, map representations to representations, I view this shift of perspective from derivations to operations as a return to conditions on transformations and representational filters.

The shift of attention from derivational economy to operations arrived shortly after widespread adoption of several assumptions about the properties of syntactic operations. A syntactic operation is taken to be a mapping from syntactic objects to one syntactic ob-

1. The Lexico-Logical Form (LLF) theory of Brody (1995) is often cited as a counterexample.

ject. The inputs to a syntactic operation must be left unchanged. For structure-building operations (e.g. Merge, Move, and Select), *unchanged* means that the inputs are contained in the output, have the same exact structure (i.e. the No-Tampering Condition (Chomsky, 2005, 2007, 2008)), and have no new features (i.e. the Inclusiveness Condition (Chomsky, 1995, 2000)). For non-structure building (e.g. Agree), unchanged means that the inputs have the exact same structure, but their set of features may have changed in some way (e.g. valued, deleted, or checked). And more, in the case of structure-building operations, one input (for internal and parallel merge) and two inputs (for external merge) must be roots—that is, operations need to obey the Extension Condition (Chomsky, 1993, 1995, 2000).[2]

This thesis continues along the trend lines I've just drawn out: It is concerned with definitions of syntactic operations, and certainly not with derivational economy. With that said, the thesis isn't only concerned with the select few set of operations that are typically posited in syntactic analyses. I find it interesting that there are several potential syntactic structure-building operations that are never considered. These operations are compliant with even the strictest possible interpretations of the principles of minimalism like the Extension Condition, the No-Tampering Condition, and the Inclusiveness Condition. However, theoretical syntacticians often assume that there is one structure-building operation (i.e. Merge). In §2.1, I propose a formalization of minimalist syntax that allows us to study the properties of syntactic structure-building operations. I find in §2.2 that conventional minimalist principles permit an infinite and varied set of operations. Some of these operations have been proposed in the literature in some form, but some others seem to have no place in a theory of syntax.

The framework also provides the tools necessary to categorize operations into families and classes, according to their properties. I pay particularly close attention to a family of

2. Even within a framework that places a large number of restrictions on operations, a lot of work needs to be done to restrict the conditions on the applicability of operations as well as the order in which the operations can apply. For example, it wouldn't be sufficient to say that structure-building operations, like Merge, can apply to any two syntactic objects. There are apparent lexically specified requirements (e.g. selection) and general restrictions (e.g. locality conditions) placed on the applicability of structure-building operations.

operations that I call the Extension Condition Family, which is defined as the set of operations that (essentially) obey the Extension Condition. This set of operations is especially interesting to me because, if they are the only licit operations, a very clear (partial) ordering of operations follows. A consequence of restricting a theory to the use of Extension Condition Family members is that a syntactic object will be a subconstituent of the output of a derivation if and only if the subconstituent was a rooted syntactic object at some earlier stage in the derivation. In §2.2.3, I zoom in on derivational theories of syntax that only permit the use of Extension Condition Family members, and I argue that every theory of that kind has a strongly equivalent declarative (also known as representational) theory; when two theories are *strongly equivalent*, they generate the same set of abstract representations. I then go on, in §2.3, to prove that one popular formalization of Minimalism (namely, the formalization in Collins and Stabler 2016) (i) only permits the use of Extension Condition Family members and (ii) has a strongly equivalent declarative alternative.

Given that many of the quintessential minimalist principles lead us to theories that don't need to posit derivations, because they have strongly equivalent declarative alternatives, I don't find it surprising that work on derivational economy has slowed. And when I think of recent work that goes against this trend by positing economy constraints, like Heck 2016; Müller 2017; Pesetsky 2021; Müller 2023, I realize that they all also posit syntactic structure-building operations that don't comply with the Extension Condition.

The objective of this thesis isn't to argue for declarative theory over a derivational theory, nor do I intend on convincing you to adopt any particular operation or set of operations. But there are many things I hope to convince you of. For one, if your syntactic theory is Extension Condition compliant, then the intermediate stages of your proposed derivations will have no bearing on the predictions of your theory, because all of the information encoded in your intermediate representations are encoded in the final outputs of your derivations. Two, if your syntactic theory permits operations that violate the Extension Condition and you make use

of those operations in an analysis, then some intermediate representations of your derivations will not be a constituent in the final representations. I call intermediate representations of this kind *intermediate constituents*. Despite the fact that it will be challenging, I hope you make an attempt to find evidence of intermediate constituents because evidence of that kind can reveal something deep and interesting about the architecture of grammatical processes.

Finding evidence that distinguishes declarative and derivational alternatives is usually challenging.[3] In their landmark article (Liberman and Prince, 1977, pp.256-261), Liberman and Prince note that their strictly representational theory of stress in English is "weakly equivalent"[4] to the derivational (or cyclic) theory of *generative phonology* (Chomsky and Halle, 1968). Although the two theories make different predictions, it is very difficult to test any of the predictions unique to a particular theory.[5] For this reason, each of the arguments in support of the representational theory are conceptual—with one exception (see fn.5).

The challenge of finding empirically testable differences between derivational and declarative theories isn't restricted to stress in English. In Chapter 3, I review some uses of syntactic operations that violate the Extension Condition. In particular, I study Richards's tucking-in analysis of multiple wh-movement in Bulgarian (Richards, 1997, 1999, 2001), as well as Lebeaux's (1991) late adjunction analysis of anti-reconstruction effects. I show that, for each of the case studies, the analysis makes use of operations outside the Extension Condition Family—that is, operations violate the Extension Condition. Theoretically, these analyses should make different predictions than an alternative that only uses Extension Condition Family members; however, I was unable to find any empirical evidence in support of one theory over the other. What I did find, though, is that the Extension Condition violations in

3. Of course, distinguishing between strongly equivalent declarative and derivational theories is impossible: They make identical predictions.

4. In formal language theory, two systems are *weakly equivalent* if they generate the same set of strings but not the same set of underlying structures.

5. They do, however, provide one example whose stress pattern is predicted by their representational theory, but not the cyclic theory. For the record, I share their judgments and those of their consultants.

syntactic analyses correlate with some desire to maintain another component of the grammar. In the case of Richards's tucking-in analysis, using an operation that violates the Extension Condition allowed Richards to maintain a particular theory of locality of movement. As for Lebeaux's late adjunction proposal, sacrificing the Extension Condition is necessary to maintain a longstanding theory of binding.

The challenge extends to theories of the syntax-prosody interface as well. Many contemporary syntactic theories assume that the syntactic component of the grammar interacts cyclically with its interfaces—prosody/phonology being one interface—so, it's particularly curious that we don't find much evidence for intermediate constituents at that interface. The notion of derivational interaction at the interfaces is known in syntactic theory as *derivation by phase* (Uriagereka, 1999; Chomsky, 2001) or *phase theory* (Abels, 2012). In phase theory, syntactic objects known as *phases* are built in the syntax, and then a subconstituent of the phase, called the *spellout domain*, is transferred to the interfaces (or *spelled out*). Some researchers have used phase theory to explain interactions between syntax and prosody/phonology (Pak, 2008; Elfner, 2012; Sande et al., 2020). However, these kinds of approaches are few and far between. Furthermore, the idea that phase theory can be successfully applied to the syntax-prosody interface has faced criticism (Cheng and Downing, 2016). In Chapter 4, I take on the challenge of looking for evidence of intermediate constituents in the derivation of prosodic structure. I provide an analysis of the interaction between intonation, syntax, and information structure in English. I show that, in some cases, conflicts between prosodic requirements are resolved with operations that are non-extending (i.e violate the Extension Condition). And more, I find that there is evidence for the intermediate constituents that are opacified by operation outside the Extension Condition Family.

CHAPTER 2

THE TROUBLE WITH STRICT EXTENSION

I want to convince you that derivational theories are odd if they include certain assumptions that strongly determine the order of operations. When I say that they are odd, I don't mean that they fail to meet standards of explanatory adequacy, and I'm not making claims about redundancy, elegance, or naturalness. I just want to show you that derivational theories that posit strictly extending, bottom-up structure building lead us to an absurd conclusion: These theories both presuppose the existence of derivations and entail that nothing can support their existence. I'm not arguing in this chapter that declarative theories are preferable to derivational theories (or vice versa). Instead, the goal is to show what an argument for a derivational theory can't look like. In Chapter 3, I show you what an argument for a derivational theory might look like.

Syntactic derivations are the chronicles of syntactic structure construction—the structure in each of its states over the course of its construction. Declarative theories (also commonly known as representational theories) don't posit the existence of derivations. Generally, the architects of declarative theories believe that derivations don't help us answer questions related to natural language syntax. For such a theoretician, it could be that there is a syntactic process but the particular process is irrelevant. If we want someone to build a house, we can give them schematics and ask them to build the structure according to the specifications provided. As long as the house meets these specifications, we don't care which of the several possible processes the constructors used to build the house[1]. For the architects of derivational theories, the syntactic process is in many ways the object of study. If we discover the process, then the process (together with lexical information) will give us the set of possible syntactic structures as well as the connection between those structures

1. For more on this idea, I refer you to Shieber (1986). For relevant sections search the document for "declarative", "declarativeness", "procedure", and "procedural". There is also some discussion of this topic in Shieber (1988).

and their phonological and semantic forms.

The chapter proceeds as follows. In §2.1, I present a formalization of several common concepts in syntactic theory. The formalisms will help to define a set of syntactic operations that I call the Extension Condition Family (§2.2). I'll argue that (derivational) theories that are restricted to operations in the Extension Condition Family will only derive syntactic representations that contain every and all the derivation's syntactic representations. I call derivations of this kind *strictly extending*. It should be clear to see that all of the intermediate steps of a derivation can be deduced from the final representation of a strictly extending derivation. What is the function of the derivation in these theories? It's not clear to me that there is one. In §2.3, I argue that every derivational theory that only permits strictly extending derivations has a strongly equivalent declarative (or representational) alternative. Two theories are strongly equivalent when they generate the exact same set of representations. §2.3 presents a proof of concept. I show that the formalization in Collins and Stabler 2016 (summarized in §2.3.2), a well-known formalism of minimalist syntax, is a strictly extending theory of syntax. I, then, prove, in §2.3.4, that there is a strongly equivalent declarative alternative to Collins and Stabler's (2016) formalization.

2.1 Definitions of syntactic objects, relations, and derivations

The goal of this chapter is to investigate the properties of syntactic structure-building operations. Given that these operations apply to and derive syntactic objects, the natural place to start is with a definition of syntactic object. It will also be necessary to define relations between syntactic objects (e.g. SO^1 contains SO^2) and syntactic derivations.

2.1.1 Syntactic objects

In general, there are two kinds of formalism in minimalist syntax: Some adopt a narrowly set-theoretic approach to syntactic representations (going forward, I'll drop the 'narrowly' for

convenience), and others adopt a graph-theoretic approach. In this section, I will provide a graph-theoretic formalism.[2] The formalism I posit is in many ways similar to previous graph-theoretic approaches (McCawley, 1968; Partee et al., 1993; Chametzky, 1996; Ermolaeva, 2021). There are, however, some significant ways in which my approach departs from the others. All previous formalisms posited the Single Root Condition: Every syntactic object (formally, tree or phrase marker) has a distinct node that dominates all other nodes. Because our investigation of syntactic structure building will lead us to analyze syntactic objects that don't have a single distinct root—for example, two-peaked structures that result from Parallel Merge (Citko, 2005) or counter-/non-cyclic Merge (Collins, 1997; Kitahara, 2011; Epstein et al., 2012)—we need a definition of graph-theoretic syntactic object that permits trees that are connected but not necessarily single-rooted graphs. One such definition is provided in (1).

(1) **Syntactic Object**

A *syntactic object* is a pair $\langle N, D\rangle$, where N is a finite set of nodes and D is a non-strict partial order (i.e. a reflexive, asymmetric, and transitive relation) in $N \times N$ $(= \{\langle x, y\rangle \mid x, y \in N\})$ such that the following condition holds:

(Connected Graph Condition) $N \times N = O^*$,

where $O^* = \{\langle x'_{\in N}, y'_{\in N}\rangle \mid \exists z'_{\in N}(\langle x', z'\rangle \in D \wedge \langle y', z'\rangle \in D)\}^*$.

According to the above definition, a syntactic object is a graph with many of the properties that syntacticians are familiar with. The graph has nodes. The nodes are connected by branches. The branches denote immediate dominance: If there is a branch from the bottom of a node x to the top of a node y, then x immediately dominates y. Because D is a transitive relation, if there is some downward path through branches from a node x to a node z, then x dominates z: $[(\langle x, y\rangle \in D \ \wedge \ \langle y, z\rangle \in D) \rightarrow \langle x, z\rangle \in D]$. If a node α only dominates itself,

2. In §2.3.2, we'll look very closely at a set-theoretic formalism (viz. Collins and Stabler's (2016) formalism).

then α is a *terminal node*; and if a node A dominates at least one node (other than itself), then A is a *nonterminal node*.

The unfamiliar part of the definition is the *Connected Graph Condition*. In short, the Connected Graph Condition requires that there is some path from every node to all other nodes in the graph. This condition is stated in terms of the relation O (for Overlap) and its transitive closure O^*. Two nodes x and y *overlap* if there is some node z such that both x and y dominate z.

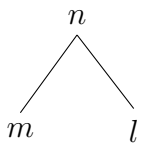
Overlap is a reflexive, symmetric, and intransitive relation. The reflexive property follows from the fact that overlap is derived from dominance. Dominance is reflexive, so overlap is reflexive: If $x = y$, then there is a z (namely, x and y) such that x and y dominate z. Unlike dominance, however, overlap is a symmetric relation. If x overlaps with y, then y overlaps with x (i.e. $\langle x, y\rangle \in O \rightarrow \langle y, x\rangle \in O$). Consider the following graph:

(2)

The graph in (2) has two nodes: n and m. Node n dominates node m. Dominance is reflexive, so m dominates m. According to the definition of overlap, n overlaps with m (i.e. $\langle n, m\rangle \in O$) iff there is a z' such that n dominates z' and m dominates z'. If $z' = m$, then node m dominates z' because m dominates itself. And n dominates z', because n dominates m. So n overlaps with m. Crucially, it's also the case that m overlaps with n (i.e. $\langle m, n\rangle \in O$), because there is a z' (m again) such that m dominates z' and n dominates z'. Overlap, then, is a symmetric relation, as the general logic of this discussion should make clear.

Overlap is not a transitive relation, (again) unlike dominance. To understand why, consider the graph in (3).

(3)



Note that n overlaps with m ($\langle n, m\rangle, \langle m, n\rangle \in O$) because both nodes dominate the node m. Next note that n overlaps with l ($\langle n, l\rangle, \langle l, n\rangle \in O$) because both nodes dominate l. However, m doesn't overlap with l ($\langle m, l\rangle \notin O$) because there is no node that both m and l dominate.

It is the case, though, that $\langle m, l\rangle \in O^*$—that is, the transitive closure of overlap (henceforth, m overlaps* with l). We find the transitive closure (denoted O^*) of overlap by adding to O the fewest possible pairs that are necessary to make O a transitive relation. A relation R is transitive if the following is true: $\forall a, b, c[(\langle a, b\rangle \in R \wedge \langle b, c\rangle \in R) \rightarrow \langle a, c\rangle \in R]$. We've established that (for the graph in (3)) $\langle m, n\rangle \in O$ and $\langle n, l\rangle \in O$, so O^* must include $\langle m, l\rangle$; and, because overlap is symmetric, it must include $\langle l, m\rangle$.

Most of the time we will look at cases where there is a difference between O and O^*, but there are some graphs that have no difference between the two sets. For any graph, there is no difference between O and O^* if and only if there are no branching nodes.

It will often be useful to refer to the root node(s) of a syntactic object, so I've given a precise definition of the *set of root nodes* of a syntactic object in (4).

(4) **Root nodes of a syntactic object**

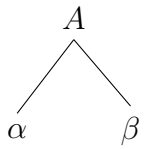
The *set of root nodes* of an SO $= \langle N, D\rangle$ is $roots^{SO} = \{x_{\in N} \mid \neg\exists y_{\in N}[\langle y, x\rangle \in D \wedge x \neq y]\}$.

The above definition says that every node in a syntactic object that isn't dominated by a distinct node (i.e. a node other than itself) is a root node. I'll sometimes refer to the set of root nodes of a syntactic object SO as $roots^{SO}$.

Consider the syntactic object in (5), which is defined as the pair $\langle N^1, D^1\rangle$. This object has a distinct node A that dominates all other nodes; in other words, A is the single root

node of the graph. When a graph-theoretic syntactic object has a single root node, I will refer to it with the symbol of the root node with a superscript SO. For example, the object in (5) is A^{SO}. The superscript SO reflects that I'm referring to the syntactic object with the single root node A, not the node A itself.

(5)

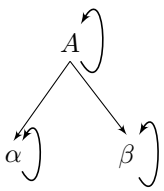


$$N^1 = \{A, \alpha, \beta\} \quad D^1 = \{\langle A, A\rangle, \langle \alpha, \alpha\rangle, \langle \beta, \beta\rangle, \langle A, \alpha\rangle, \langle A, \beta\rangle\}$$

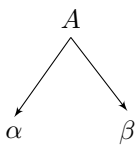
The syntactic object in (5) is defined as the pair $\langle N^1, D^1\rangle$, where N^1 is the set of the object's nodes and D^1 the set of its dominance relations. The set D^1 has five elements, each of which is a pair where the first term is understood to dominate the second (e.g. $\langle A, \alpha\rangle$ means 'A dominates α'). In accordance with the definition of syntactic object in (1), every dominance relation in D^1 holds between two nodes in N^1. There isn't, for example, a pair $\langle A, \delta\rangle$ in D^1: δ is not a node in the graph (i.e. $\delta \notin N^1$). Dominance (D^1) is a reflexive relation, so for each node n in N^1, there is a pair $\langle n, n\rangle$ in D^1; if it were the case that we substituted $D^{1'}$ for D^1, where $D^{1'}$ is identical to D^1 except that it doesn't contain $\langle \alpha, \alpha\rangle$, then the resulting graph would not be a syntactic object because $D^{1'}$ is not a reflexive relation.[3]

3. Syntacticians conventionally draw syntactic trees with branches that aren't arrows. It is probably more accurate to draw syntactic trees as directed graphs (i.e. trees with arrow branches) because the arrows denote an asymmetric relation: if A dominates a distinct node α, α doesn't dominate A. Additionally, when dominance is taken to be a reflexive relation (as it is here), then we can include cycles (or loops) in our (directed) graphs. By using cycles, we can draw distinct graphs for trees with reflexive and irreflexive dominance. Consider the examples in (i).

(i) a.



b.

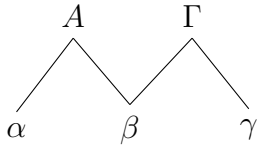


First note that (in both graphs) the arrows from A to α and β properly capture the fact that dominance is an

When we look at the drawn graph in (5), it's pretty easy to conclude that A^{SO} meets the Connected Graph Condition, but I will briefly discuss why it does. Recall that I use the transitive closure of the relation overlap (O^*) to express that two nodes are connected. It's trivially true that A overlaps with α and with β because A dominates both and both α and β dominate themselves. In fact, because D^1 is a reflexive relation, each node overlaps with itself. It's slightly more difficult to determine whether or not α and β overlap*—that is, that they are in the transitive closure of Overlap—because neither dominates the other. But in order to see that they do, in fact, overlap* we simply need to notice that α overlaps with A (recall that overlap is symmetric) and A overlaps with β. Therefore, α and β are in the transitive closure of overlap (O^*)—that is, $\langle\alpha,\beta\rangle, \langle\beta,\alpha\rangle \in O^*$.

When a syntactic object doesn't have a single root node, I will refer to the syntactic object with a ⛰ symbol. To take an example, the syntactic object in (6) has two nodes A and Γ that are not dominated by any node (but themselves).

(6)



$$N^2 = \{A, \Gamma, \alpha, \beta, \gamma\}$$
$$D^2 = \{\langle A,A\rangle, \langle\Gamma,\Gamma\rangle, \langle\alpha,\alpha\rangle, \langle\beta,\beta\rangle, \langle\gamma,\gamma\rangle, \langle A,\alpha\rangle, \langle A,\beta\rangle, \langle\Gamma,\beta\rangle, \langle\Gamma,\gamma\rangle\}$$

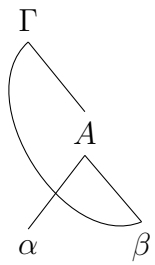
The graph in (6) is a syntactic object according to the definition in (1). From the discussions around (3) and (5)), it should be clear at this point all pairs of nodes in {α, β, A} as well as {β, γ, Γ} overlap*. Now we will ask whether α overlaps* with Γ and γ and whether A overlaps* with Γ and γ. It's surprisingly easy to see that all the nodes overlap* once you

asymmetric relation. Second, the loops in (a) connecting each of the nodes in the graph to themselves clearly show that the relation is reflexive, whereas the lack of loops in (b) denotes that the relation is irreflexive. Both (a) and (b) are well-formed directed graphs, but only (a) is a syntactic object according to the definition in (1).

realize that they all overlap* with β. α overlaps* with β and β with Γ; so by the transitive property of O*, α and Γ overlap*. The same logic can be used to prove that any two nodes in (6) overlap*: For any two nodes, they both (symmetrically) overlap* with β).

The syntactic object ▲ in (6) has a node β that is immediately (irreflexively) dominated by two nodes A and Γ. Neither A nor Γ dominates the other. In (7), the node β is immediately (irreflexively) dominated by both A and Γ, and in contrast to (6), Γ dominates A. The multidominance of a node (β, in this case) is the way I'll formalize a syntactic movement dependency,[4] so it's important to show that (7) is a syntactic object according to the definition in (1).

(7)



$$N^3 = \{\Gamma, A, \alpha, \beta\}$$

$$D^3 = \{\langle\Gamma,\Gamma\rangle, \langle A, A\rangle, \langle\alpha,\alpha\rangle, \langle\beta,\beta\rangle, \langle A,\alpha\rangle, \langle A,\beta\rangle, \langle\Gamma, A\rangle, \langle\Gamma,\alpha\rangle, \langle\Gamma,\beta\rangle\}$$

The graph in (7) is defined as the pair $\langle N^3, D^3\rangle$. I'll leave it to you to confirm that the set D^3 is a strict partial order. Of course, the other condition that a graph must meet in order to be a syntactic object is the Connected Graph Condition, which for (7), requires that every pair of nodes in N^3 overlaps*. I walk through how (7) meets this condition.

Note that every node but β dominates α and every node but α dominates β. We conclude from those facts that (i) $\alpha, A,$ and Γ overlap* and (ii) $\beta, A,$ and Γ overlap*. The only remaining question is whether or not α and β overlap*. The answer is yes: α overlaps with A and A overlaps with β, so by the transitive property of O^*, we can conclude that α and

4. See §2.2 and §3.2.3 for more on syntactic dependencies.

β overlap*.

2.1.2 Relations between syntactic objects

Now that I've defined what a syntactic object is and provided several examples of syntactic objects, we can move on to talking about the relations between syntactic objects, starting with the containment relation. A syntactic object can be said to *immediately contain* a syntactic object. Immediate containment is a relation between syntactic objects, unlike dominance, which is a relation between nodes in a graph. I've provided a graph-theoretic definition of immediate containment in (8).

(8) **Immediate Containment (graph-theoretic)**

A syntactic object $SO^1 = \langle N, D\rangle$ is *immediately contained* in a syntactic object $SO^2 = \langle N', D'\rangle$ iff

a. SO^1 is a subgraph of SO^2, and

b. $\forall x_{\in N} \forall y_{\in N'} [\langle x, y\rangle \in D' \rightarrow \langle x, y\rangle \in D]$, and

c. $\exists r_{\in roots^{SO^1}} \exists r'_{\in roots^{SO^2}} [\langle r', r\rangle \in D' \wedge$
$\neg \exists x_{\in N'} (x \neq r, r' \wedge \langle r', x\rangle \in D' \wedge \langle x, r\rangle \in D')]$[5]

The definition of immediate containment says that a syntactic object SO^1 is immediately contained in an object SO^2 iff it meets three conditions. Condition (a) requires SO^1 to be a subgraph of SO^2; I'm assuming the definition of a subgraph in (9).

(9) **Subgraph Relation**

A graph $G^1 = \langle N, D\rangle$ is a *subgraph* of a graph $G^2 = \langle N', D'\rangle$ iff (i) $N \subseteq N'$ and (ii) $D \subseteq D'$.

5. According to this definition, immediate containment is a reflexive relation (i.e every syntactic object immediately contains itself). I'll elaborate on this point in fn.7 below.

As we will discuss in more detail shortly, the second condition of Immediate Containment (8b) ensures that the containment relation between two objects aligns with the notion of constituent in syntactic theory, which is more restrictive than the notion of subgraph in graph theory. The third condition (8c) is what makes the relation *immediate* containment as opposed to simple containment.

We can derive the *containment* relation from the primitive immediate containment[6] relation as follows:

(10) **Containment (graph-theoretic)**

A syntactic object SO^1 is *contained* in a syntactic object SO^2 iff

a. SO^2 immediately contains SO^1, or

b. there is some syntactic object SO^3 such that SO^2 immediately contains SO^3 and SO^3 contains SO^1.

---

6. Picking immediate containment as the primitive relation and containment as the derived relation wasn't arbitrary. I noticed that the alternative—taking containment as primitive—led to different sets of immediate containment relations. To demonstrate, imagine we adopt the following definition of containment, which seemed to me at one time to be a reasonable and workable definition:

(i) **Containment (primitive)**
A syntactic object $SO^1 = \langle N, D\rangle$ is *contained* in a syntactic object $SO^2 = \langle N', D'\rangle$ iff
a. SO^1 is a subgraph of SO^2, and
b. $\forall x_{\in N} \forall y_{\in N'} [\langle x, y\rangle \in D' \rightarrow \langle x, y\rangle \in D]$.

The above definition is relatively simple; in fact, it is identical to the definition of immediate containment in (8) minus condition (c). Consider now the definition of immediate containment in (ii), which uses a standard strategy for deriving a more restrictive relation (namely, immediate containment) from a less restrictive relation (containment):

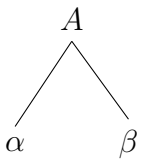
(ii) **Immediate Containment (derived)**
A syntactic object SO^1 is *immediately contained* in a syntactic object SO^2 iff (i) SO^1 is contained in SO^2 and (ii) there is no syntactic object SO^3 (distinct from SO^2) such that SO^1 is contained in SO^3 and SO^3 is contained in SO^2.

According to the definitions in this footnote, where containment is primitive, the syntactic object β^{SO} in (6)—the shared element in the two-peaked object ⛰—isn't immediately contained in ⛰, because Γ^{SO} is contained in ⛰ and β^{SO} is contained in Γ^{SO}. By the set of definitions in the main text, where immediate containment is primitive, β^{SO} is immediately contained in ⛰. We could adjust the definitions of these in such a way that the system has the same properties as the alternative (in the main text), but at least at the moment, I can only imagine very complicated adjustments.

Containment is just the transitive closure of immediate containment. The definitions of immediate containment and containment should be familiar to those who have worked within earlier graph-theoretic syntactic frameworks.[7]

As I mentioned previously, the containment relation is crucially more restrictive than the subgraph relation. Given that being a subgraph of a syntactic object is just one condition of being contained in that syntactic object, it's necessarily the case that the containment relation is stricter than the subgraph relation. Nonetheless, I think it will be fruitful to consider an example that demonstrates this fact. In (11), I've provided two graphs side-by-side. On the left-hand side is the graph $A^{SO\prime}$ (repeated from (5)), which is defined as the pair $\langle N^1, D^1\rangle$, and on the right-hand side is the graph $A^{SO} = \langle N^4, D^4\rangle$.

(11)

$\underline{A^{SO\prime}\text{ (A supergraph of }A^{SO})}$	$\underline{A^{SO}\text{ (A subgraph of }A^{SO\prime})}$
	
$N^1 = \{A, \alpha, \beta\}$	$N^4 = \{A, \alpha\}$
$D^1 = \{\langle A, A\rangle, \langle \alpha, \alpha\rangle, \langle \beta, \beta\rangle, \langle A, \alpha\rangle, \langle A, \beta\rangle\}$	$D^4 = \{\langle A, A\rangle, \langle \alpha, \alpha\rangle, \langle A, \alpha\rangle\}$

A^{SO} is a subgraph[8] of $A^{SO\prime}$, because (i) $N^4 \subseteq N^1$ and (ii) $D^4 \subseteq D^1$. But $A^{SO\prime}$ doesn't immediately contain A^{SO}, because condition (b) of the definition of Immediate Containment (8) is not met. A is a node in both A^{SO} and $A^{SO\prime}$. In $A^{SO\prime}$, node A dominates node β, but it doesn't dominate β in A^{SO}. With respect to this example, condition (b) says the following: If A dominates β in $A^{SO\prime}$, then A must dominate β in A^{SO}.

Although the object $A^{SO\prime}$ doesn't contain A^{SO}, it is a matter of fact that A^{SO} contains

7. As alluded to in fn.5, (immediate) containment is defined here as a reflexive relation (i.e. every SO (immediately) contains itself). We can make the relations irreflexive (if we needed or wanted to) by adding to the definition of immediate containment at (8c) the condition that $r \neq r'$.

8. A^{SO} is also a licit syntactic object.

α^{SO}, even in $A^{SO'}$. It's been assumed (at least implicitly) throughout the history of generative syntax that this containment relation between A^{SO} and α^{SO} should have no role in the theory of grammar. And when someone asks what contains α^{SO} in $A^{SO'}$, they mean to ask which constituents of $A^{SO'}$ contain α^{SO}, where a constituent of A^{SO} is a syntactic object contained in A^{SO}. For this reason, it will be useful to define the set of constituents of a syntactic object that contain a syntactic object, which I've done in (12).

(12) **Constituents containing SO**

The set of *constituents of* B^{SO} *containing* A^{SO} is $Con_A^B = \{SO \mid B^{SO}$ contains SO and SO contains $A^{SO}\}$.

The definition in (12) defines the subset of subgraphs that syntacticians are interested in when they want to know which syntactic objects contain another. On a similar note, when a syntactician says that a syntactic object is *uncontained*, they typically mean something like the following:

(13) **Uncontained Syntactic Object**

A syntactic object A^{SO} is an *uncontained syntactic object* iff there is no syntactic object B^{SO} such that

a. B^{SO} contains A^{SO} and

b. $B^{SO} \neq A^{SO}$.

The above definition simply says that a syntactic object A^{SO} is uncontained if there is no syntactic object distinct from A^{SO} that contains A^{SO}.

2.1.3 Derivations

Any framework that makes use of some notion of extension assumes that syntactic objects (at least those that aren't lexical items) are built over the course of a *derivation*. There have been several approaches to the formalization of syntactic derivations. I've elected to adopt

a definition that is very similar to the definition posited in Collins 1997 (p.82, ex.47). My version in (14) defines a derivation as a sequence of sets (called *stages*). The initial stage is a finite set of lexical items; I may sometimes refer to the initial stage as the *numeration* or *lexical array*. Apart from the initial stage, each stage S_i in the derivation is derived from the previous stage S_{i-1} by (i) taking a subset of S_{i-1} called Z, (ii) passing the elements of Z to a structure-building operation f,[9] (iii) removing the elements of Z from S_{i-1},[10] and (iv) adding the output of f to S_{i-1}.[11]

(14) **Derivation**

A *derivation* is a finite sequence of finite sets (called *stages*) $\langle S_1, \ldots, S_n \rangle$, for $n \geq 1$, such that

a. every element of S_1 (the initial stage) is a lexical item;

b. S_n is a singleton set; and

c. for all S_i, for $1 \leq i < n$, the following holds:

   i. There is a nonempty subset Z of S_i where $Z = \{z_1, \ldots, z_n\}$; and

   ii. no element of Z is an element of S_{i+1}; and

   iii. there is a syntactic object y such that y is not an element of S_i and y is an element of S_{i+1}; and

   iv. there is a structure-building operation f such that $f(z_1, \ldots, z_n) = y$.

As a shorthand, when a syntactic operation f is involved in the derivation of one stage S_i to the next S_{i+1}, we will say that S_{i+1} is derived (from S_i) by f. Also, if some stage of a derivation D is derived by an operation f, we will say that D involves f.

9. I'm making the simplifying assumption that the only syntactic operations are structure-building operations.

10. $S_{i-1} -$ Z.

11. $S_{i-1} \cup \{f(\ldots)\}$.

An *intermediate representation* of a derivation is any syntactic object that is an element of some stage of that derivation, excluding the syntactic object in the final stage of the derivation. We can formally define the set of intermediate representations of a given derivation D as the union of all stages up to and including the penultimate stage (see (15)).

(15) **Intermediate Representations**

The *set of intermediate representations* of a derivation $\mathrm{D} = \langle S_1, \ldots, S_n \rangle$ is $\bigcup_{i=1}^{n-1} S_i$.

If a syntactic object A^{SO} is an element of the set of intermediate representations of a derivation D, I'll say that A^{SO} is an intermediate representation of D. Note that the set of intermediate representations doesn't contain the sole member of the final stage in a derivation. Going forward, I'll refer to the sole member of the final stage of a derivation D as the *final representation of D*.

Let's assume that we know a unique corresponding final representation for every expression of every language. When we investigate an expression of a language, we can still only observe the properties of the final representation. We don't have direct access to any of the intermediate representations of a derivation. However, it may be possible to deduce from the final representation what the intermediate representations of a derivation are. If we can find out what the representations are, we'll say that the set of intermediate representations of a derivation D are *determinable* from the final representation of D:

(16) **Determinable**

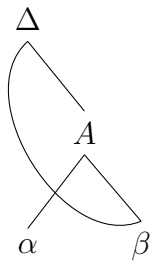
The set of intermediate representations of a derivation $\mathrm{D} = \langle S_1, \ldots, S_n \rangle$ is *determinable from* S_n iff there is an effective procedure (i.e. an algorithm) that determines the set of intermediate representations from the information given exclusively by the final representation of D.

We can only know if the set of intermediate representations of a derivation are determinable from the final representation if we know what the intermediate representations are. This is

really only the case when we are evaluating an analysis of an expression. So the concept of determinability is a useful tool when we are evaluating syntactic analyses. In particular, we will see that an analysis that is derivationally construed will always have a representational/declarative alternative when the proposed derivation is determinable—that is, its set of intermediate representations are determinable from its final representation. However, we have many more steps to make toward that conclusion.

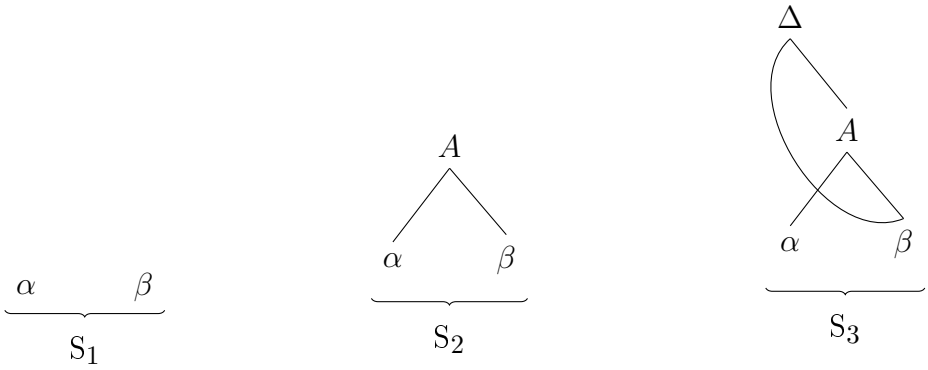
It will be useful to consider an example of how one derivation to a syntactic object is determinable while another to the same syntactic object isn't. Consider the syntactic object in (17), which we can call Δ^{SO}.

(17)



Because we haven't restricted in any way the set of structure-building operations, there are an infinite number of derivations to the syntactic object in (17). One derivation is the following:

(18)



The first stage (S_1) is a set set containing only the lexical items α^{SO} and β^{SO}. The sec-

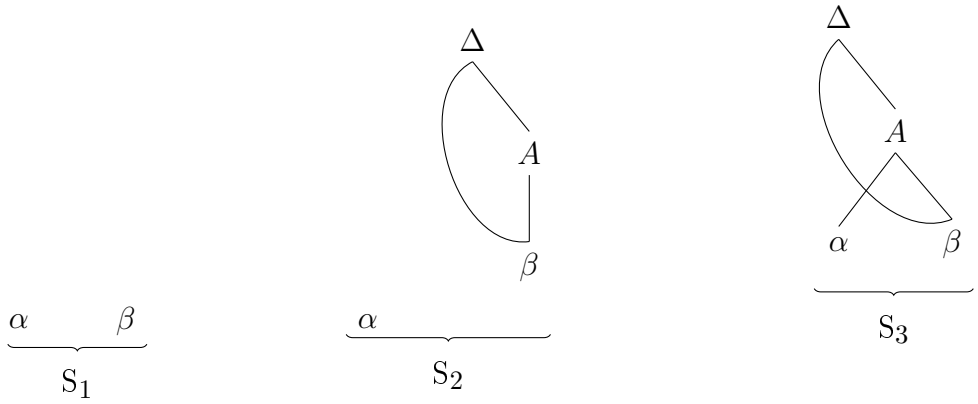
ond stage (S_2) is a singleton set whose member is A^{SO}; A^{SO} has a single root node A and immediately contains the lexical items α^{SO} and β^{SO}. The final stage (S_3) has the final representation Δ^{SO} as its sole member. The set of intermediate representations of the derivation is $\{\alpha^{SO}, \beta^{SO}, A^{SO}\}$. Is there some effective procedure that we can use to determine this set of intermediate representations from the final representation? Yes, any procedure that determines the set of constituents irreflexively contained in the final representation will return the set of intermediate representations. The following will do:

(19) **Containment Procedure**

1. Find the set of constituents contained in the final representation.
   ($Con^{final} = \{SO \mid$ final representation contains $SO\}$)
2. Remove the final representation from the result of Step 1.
   ($Con^{final} - \{SO^{final}\}$)

The Containment Procedure is stated in a coarse-grained manner, but it shouldn't really matter for our purposes. The upshot is that some final representations contain all and only the intermediate representations of their derivation. This is the case for the final representation of (18). Now consider the following three stage derivation to the same exact final representation:

(20)



This derivation begins with the same initial stage (S_1) that contains the two lexical items α^{SO} and β^{SO}. The second stage (S_2) has two members—the lexical item α^{SO} and a complex syntactic object $\Delta^{SO'}$. The final stage (S_3) contains the final representation in (17), which we can call Δ^{SO}. The set of intermediate representations of this derivation is $\{\alpha^{SO}, \beta^{SO}, \Delta^{SO'}\}$. In this case, it is unclear that there is any effective procedure we could use to determine the set of intermediate representations from the information given to us by the final representation Δ^{SO}. We certainly couldn't use the Containment Procedure because it would return the wrong set of representations. The procedure would both overgenerate by including the syntactic object A^{SO} from the second stage of the derivation in (18), and it would undergenerate by excluding the syntactic object $\Delta^{SO'}$ from the second stage of the derivation in (20). I call a intermediate representation of a derivation that is not determinable from the final representation of that derivation an *intermediate constituent*. The object $\Delta^{SO'}$ is an intermediate constituent of the derivation in (20).

The definition of a syntactic derivation in (14) doesn't specify what the set of structure-building operations are. In §2.2, I will investigate one possible class of structure-building operations. I call this class of operations the *Extension Condition Family* because it is the set of operations that are compliant with the spirit of the Extension Condition (Chomsky, 1993, 1995, 2000). I walk through several examples of operations in the Extension Condition Family in §2.2.1 and some operations outside the Extension Condition Family in §2.2.2. In §2.2.3, I argue that theories that only permit operations from the Extension Condition Family will necessarily have a strongly equivalent declarative (or representational) alternative theory. In§2.3.2, I review a commonly adopted formalization of a minimalist syntax (i.e. Collins and Stabler's 2016 formalization), show that the formalization only permits operations from the Extension Condition Family, and prove that it has a strongly equivalent declarative alternative.

2.2 Strict extension

As a direct descendant of the Strict Cycle Condition (Chomsky, 1977; Freidin, 1978; Müller, 2010, 2023), the Extension Condition (previously the Extension Requirement) is the quintessential minimalist cyclic mechanism (Chomsky, 1993, 1995, 2000). The Extension Condition is best understood as a meta-theoretical constraint on the definition of syntactic structure-building operations. Informally, the Extension Condition says that structure-building operations, however else you define them, must apply to and extend uncontained syntactic objects. Although the Extension Condition hasn't been given a very formal definition, its adoption has led to both compelling analyses of natural language phenomena and empirical discoveries. But a proper investigation into the properties of the condition requires a more rigorous formal approach with clearly stated definitions. As we begin to develop formal definitions of fundamental notions in our theories, we will find that there are a ton of implicit notions that are necessary to make our analyses tick.

Before introducing a formal definition of the Extension Condition, we need to define the notion *extension*. The Extension Condition requires structure-building operations to extend syntactic objects. We can't know if a structure-building operation extends a syntactic object without a grounded understanding of extension. In (21), I've provided what I believe to be a definition of extension that is consistent with the colloquial usage of the term. I've also included a definition of *extend by* n, where n is the degree to which an operation extends an object.

(21) **Extension (by n)**

Let f be a structure-building operation whose input(s) are $inputs^f = \{z_1, \ldots, z_n\}$ and whose output is y. Let A^{SO} be a syntactic object such that

a. the set of constituents that contain A^{SO} in $inputs^f$ is $C = \bigcup_{i=1}^{n} Con_A^{z_i}$, and

b. the set of constituents that contain A^{SO} in f's output y is $C' = Con_A^y$.

- Def. 1: f *extends* A^{SO} iff $C \subset C'$
- Def. 2: f *extends* A^{SO} *by* n iff
  i. f extends A^{SO} and
  ii. $|C' - C| = n$.

According to Def. 1 in (21), a mapping extends a syntactic object A^{SO} iff the set of constituents that contained A^{SO} in the input(s) is a proper subset of the set of constituents that contain A^{SO} in the output. A mapping only extends a syntactic object A^{SO} if every object that contains A^{SO} in the inputs also contains A^{SO} in the output. If A^{SO} is contained in B^{SO} in some input to f but A^{SO} isn't contained in B^{SO} in the output to f, then f doesn't extend A^{SO}. Furthermore, because a syntactic object must be contained in at least one new constituent in the output (because C must be a *proper* subset of C'), an identity mapping—for example, a mapping that takes A^{SO} as a sole input and returns A^{SO}—doesn't extend its input. Note that a mapping can extend a syntactic object that isn't contained in any input to the mapping. For example, if A^{SO} is contained in the output of f but not in any input to f, then f extends A^{SO}.

The second bulleted definition (Def. 2) in (21) determines the degree to which a mapping extends a syntactic object. Of course, in order for a mapping to extend a syntactic object by some degree, it must extend it, period; this is what the first condition ensures. The second condition says that a mapping extends a syntactic object A^{SO} by n iff there are n new syntactic objects that contain A^{SO} in the output of the mapping.

With a definition of extension (by n) in hand, we can define the *Extension Condition Family* as in (22). It is the case here, as it may be for all other structure-building operation families, that a structure-building operation is (i) defined as a mapping to exactly one syntactic object and (ii) as a mapping from at least one syntactic object.

(22) **Extension Condition Family**

A syntactic structure-building operation is a member of the Extension Condition Family iff it is a mapping f from n syntactic objects, for $n \geq 1$, to a single syntactic object SO' such that:

(Extension Condition) For every syntactic object SO that is either (i) the output of f or (ii) contained in an input to f, f extends SO by 1 and only 1; and

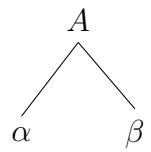
(Anti-Inclusion Condition) f extends no other syntactic object.

The Extension Condition Family, as defined in (22), is the set of structure-building operations that obey two conditions: *Extension Condition* and *Anti-Inclusion Condition*. The *Extension Condition* requires mappings to extend every input syntactic object (and the [other] syntactic objects it contains) and the output syntactic object by 1 (and only 1). The *Anti-Inclusion Condition* constrains mappings to extend only those syntactic objects required by the Extension Condition.

2.2.1 Binary Merge and other family members

In order to briefly demonstrate that these definitions work for most purposes, let's verify that run-of-the-mill Binary Merge (BinMerge), as in (23), is a member of the Extension Condition Family.

(23) **Binary Merge (BinMerge)**

$\alpha \qquad \beta \qquad$ =BinMerge⇒ $\qquad$ 

It should be clear that Binary Merge is Extension Condition compliant. The two inputs are immediately contained in the output A^{SO} and nothing else irreflexively contains the inputs.

Therefore the output A^{SO} is the only new constituent that contains the inputs. Binary Merge also extends the output A^{SO} by 1: It contains itself in the output and isn't contained in any input. The operation is also compliant with the Anti-Inclusion Condition because Binary Merge only extends its two inputs (plus anything contained in them) and its output.

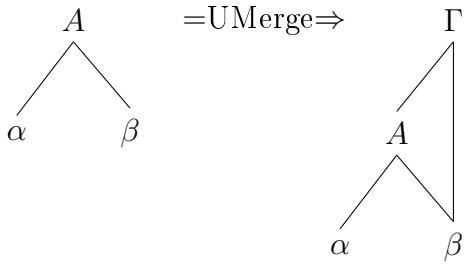
In fact, the only part of Binary Merge in (23) that doesn't follow from the definition of Extension Condition Family is that the operation takes exactly two inputs. Otherwise, it follows that the output syntactic object A^{SO} immediately contains α^{SO} and β^{SO}. If A^{SO} didn't contain either of α^{SO} or β^{SO}, then it would be the case that the mapping doesn't extend at least one of the syntactic objects contained in some input; in this case, it would be the input itself that isn't extended. Further, if A^{SO} contains but doesn't immediately contain both of α^{SO} and β^{SO}, then at least one of the syntactic objects in the input would have been extended by more than one. And because a mapping is only compliant with the Extension Condition if it extends all of its inputs by 1 and *only 1*, a mapping that extends an input by more than 1 is not a member of the Extension Condition Family.

Binary Merge is just one of a class of operations in the Extension Condition Family that I call the n-ary External Merge class. Each operation in this class takes n (for $1 < n$) inputs and returns an n-ary branching object that immediately contains all and only the inputs. Of this class, Binary Merge is the only operation that has been proposed in the minimalist literature—as far as I'm aware. However, other n-ary Merge operations (i.e. Ternary Merge) have their place in generative syntax, usually in the form of a phrase structure rule.

There are three Extension Condition Family members, outside of the n-ary External Merge class, that I would like to discuss. I've chosen to discuss these three particular operations because they each illustrate something important about the Extension Condition Family. First, let's consider a unary operation in (24) that bears a striking resemblance to Internal Merge (syntactic movement). Internal Merge is typically formulated as a binary operation, but by taking a formal approach to the restrictions on structure-building operations,

we reveal that syntactic movement can be modeled as a unary operation.

(24) **Unary Merge (UMerge)**



The output of Unary Merge is a multidominant structure. Multidominant structures are syntactic objects in which some object (β^{SO} above) is immediately contained in at least two objects (A^{SO} and Γ^{SO} above). One common approach to syntactic movement dependencies is to analyze them as the result of multidominance. Proponents of this view might say in an informal way that β^{SO} in (24) has moved to become sister to A^{SO}.

At least for now, I'm not concerned with the usefulness of the operation Unary Merge to syntacticians. Instead, I want to demonstrate that Unary Merge is a member of the Extension Condition Family. To start, it maps one syntactic object onto another (a 1-to-1 mapping). Furthermore, it's compliant with the Extension Condition. I'll run through why β^{SO}, the multiply contained object, is extended by 1 and only 1. You can verify that α^{SO}, A^{SO}, and Γ^{SO} are also extended by 1 and only 1. In the input to Unary Merge in (24), β^{SO} is contained in two objects (A^{SO} and β^{SO}), and in the output, β^{SO} is contained in three objects—in addition to A^{SO} and β^{SO}, it's contained in Γ^{SO}. We can also confirm that the Anti-Inclusion Condition is not violated by Unary Merge, because the only objects that are contained in the output (Γ^{SO}) are α^{SO}, β^{SO}, A^{SO}, and Γ^{SO}. The first three are contained in the input, and the last is the output. Because every syntactic object irreflexively contained in the output is contained in some input, it follows that Unary Merge doesn't violate the Anti-Inclusion Condition. If there were a syntactic object irreflexively contained in the output that is not contained in an input, then that object would have been

extended in violation of the Anti-Inclusion Condition. We've now established that Unary Merge is compliant with both the Extension and Anti-Inclusion Conditions, and is, therefore, a member of the Extension Condition Family.

The second operation I would like to discuss is an operation called Self Merge (25). It isn't very common to see an operation like Self Merge in the syntactic literature. Guimarães (2000); Kayne (2008)[12] both propose it to, among other things, resolve compatibility issues between Bare Phrase Structure and the Linear Correspondence Axiom.

(25) **Self Merge (Self)** (Kayne, 2008) [adapted]

α =Self⇒ A (with α as its daughter)

In brief, Self Merge takes a single input α^{SO} and outputs a new object A^{SO} that immediately contains the input. The input α^{SO} is extended by 1 and only 1 because the additional object that contains it in the output is the output, and the output A^{SO} is extended by one because only one additional object contains it in the output (namely, the output itself). So Self Merge complies with the Extension Condition. It also complies with the Anti-Inclusion Condition. The only two objects contained in the output are the output itself and the

12. Guimarães's (2000) formulation is slightly different than Kayne's (2008). Guimarães (2000) proposes a binary operation:

(i) **Self Merge** (Guimarães, 2000) [adapted]

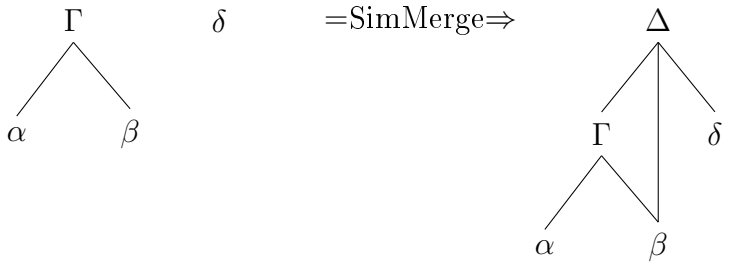
α α =Self Merge⇒ A (with α as its daughter)

A binary Self Merge is incompatible with my definition of derivation (14) because it requires that two identical objects are members of the same derivational stage. Because I model derivational stages using sets, there can't be two identical members of a single stage. However, as is usually the case, a definition can be redefined.

solitary input. It follows, then, that the only two objects that could possibly have been extended are the output and the input. We can therefore conclude that Self Merge is a member of the Extension Condition Family.

The final operation I'd like to discuss represents a class of operations that belong to the Extension Condition Family but look absolutely outrageous. I'm almost certain no one has ever proposed anything that resembles a member of the class, and despite some effort, I have failed to find any empirical support for a single one of them. Behold Simultaneous Merge:

(26) **Simultaneous Merge (SimMerge)**



Believe it or not, Simultaneous Merge is an Extension Condition Family member. It's a binary operation into a single syntactic object. The two inputs (Γ^{SO} and δ^{SO} above) are extended by one. Before the mapping they only contain themselves, and in the output, they are additionally contained in Δ^{SO}. The objects irreflexively contained in inputs (α^{SO} and β^{SO}) are also extended by one: Before the mapping they are both contained in two objects (themselves and Γ^{SO}), and after the mapping, they are also contained in Δ^{SO}. Further, Simultaneous Merge obeys the Anti-Inclusion Condition because nothing is extended but $\alpha^{SO}, \beta^{SO}, \Gamma^{SO}, \delta^{SO}$ and the output Δ^{SO}.

The reason that I've named the operation Simultaneous Merge is because it looks as if it applies External Merge of δ^{SO} and Internal Merge of β^{SO} simultaneously. In this class of simultaneous operations are an infinite number of operations that seem as if k (for $k \geq 1$) instances of Internal Merge and j (for $j \geq 0$) instances of External Merge have applied simultaneously. For example, there is an operation g that takes Γ^{SO} from (26) and

simultaneously applies Internal Merge to both α^{SO} and β^{SO}, and g is a member of the Extension Condition Family.

We've now taken a short tour through the halls of the Extension Condition Family home. We have found a very familiar character in Binary Merge. We've also seen some members that we may or may not have met before, like Unary Merge and Self Merge. And we've been lucky to meet Simultaneous Merge, the king of the family misfits. In the next section, we'll leave the Extension Condition Family alone and investigate some operations that aren't in the Extension Condition Family.

2.2.2 Operations outside the Extension Condition Family

We can't investigate all infinity of Extension Condition violating operations, but I've picked out five interesting cases:

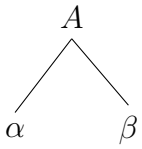
1. **Hyper Merge** violates the Anti-Inclusion Condition
2. **Super, Replacement, and Parallel Merge** violate the Extension Condition and the Anti-Inclusion Condition
3. **Identity** violates the Extension Condition

I've picked these operations because they exemplify the three classes of operations that don't fit into the Extension Condition Family. Given that there are two conditions that need to be met in order to be in the Extension Condition Family, every operation should fall into one of four classes. One class, the Extension Condition Family, is defined as the class that meets the two conditions. Hyper Merge represents the class that violates Anti-Inclusion but not the Extension Condition; Super Merge, Replacement Merge, and Parallel Merge represent the class that violates the Extension Condition and Anti-Inclusion; and the Identity operation represents the class that violates the Extension Condition but not Anti-Inclusion.

First, consider the unary operation *Hyper Merge* in (27), which is an example of an operation that obeys the Extension Condition but violates the Anti-Inclusion Condition.

(27) **Hyper Merge (Hype)**

α =Hype⇒

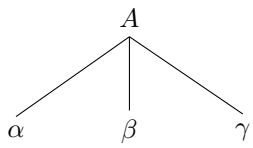


Hyper Merge takes one input and returns a single binary branching object, so it's a 1-to-1 mapping,[13] This operation is Extension Condition compliant because it extends its output and its input (and everything contained in them) by 1 and only 1. However, it violates the Anti-Inclusion Condition because it extends an object that isn't contained in the input (γ^{SO}). Therefore, Hyper Merge is not a member of the Extension Condition Family. As Karlos Arregi (p.c.) noted, several of the rules in Montague 2002 introduce operators syncategorematically, like the rule that attaches quantifiers and determiners to noun phrases. These rules look a lot like Hyper Merge. There are, however, an abundance of alternatives to syncategorematic rules in the literature; for example, in Dowty et al.'s (1981, p.194) treatment of Montague's grammar, the authors decide to abandon the original syncategorematic approach and adopt a non-syncategorematic rule for quantification of noun phrases.

13. I've defined Hyper Merge as a unary operation, but Hyper Merge stands in for a class of non-Extension Condition Family members. There is an n-to-1 Hyper Merge for any n greater than 0. For example, there is a binary Hyper Merge, which I've provided in (i).

(i) **Binary Hyper Merge**

α β =Hype⇒



Binary Hyper Merge, like every operation in this class of operations, complies with the Extension Condition but violates Anti-Inclusion.

I'd like to discuss another 1-to-1 mapping, called Super Merge (28), because it strongly resembles Self Merge. I showed that Self Merge is an Extension Condition Family member in §2.2.1. But Super Merge isn't an Extension Condition Family member because violates both the Extension Condition and the Anti-Inclusion Condition.

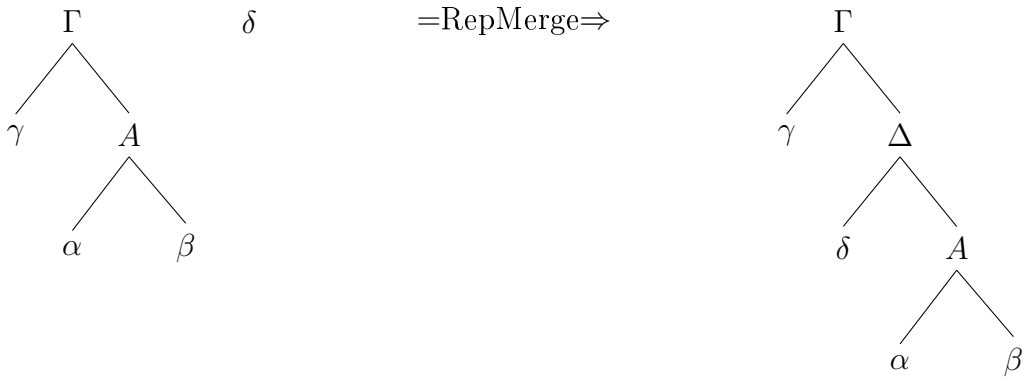
(28) **Super Merge (Super)**

α =Super⇒

Super Merge violates the Extension Condition because the input α^{SO} is extended by more than one. Before the mapping, α^{SO} is contained in one constituent—that is, itself. After the mapping, α^{SO} is contained in two additional objects (i.e A^{SO} and B^{SO}). Super Merge also violates the Anti-Inclusion Condition because A^{SO} is extended although it (i) isn't the output and (ii) isn't contained in the input.

Next we'll consider a kind of operation that has been proposed in various forms. I'll call the operation Replacement Merge (RepMerge).

(29) **Replacement Merge (RepMerge)**



Replacement Merge, as defined in (29), is a 2-to-1 mapping of syntactic objects to a syntactic object. Replacement Merge replaces a constituent irreflexively contained in an input (A^{SO}) with a new object (Δ^{SO}), and the new object immediately contains the object it replaces. This operation is essentially what is involved in countercyclic merge/movement into specifier positions and late adjunction. In Chapter 3, we will discuss two applications of Replacement Merge that have been proposed in the literature (namely, tucking-in and late adjunction).

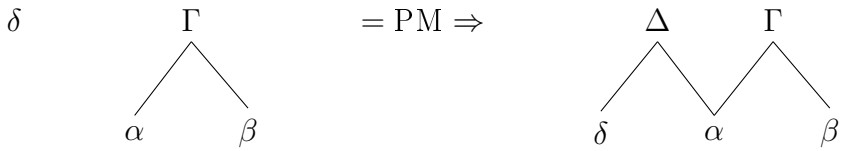
There are several counts of Extension Condition violations associated with Replacement Merge. To start, one of the inputs (Γ^{SO}) is not extended because it's not contained in the output. Furthermore, anything else that is contained in that input (i.e. A^{SO}, α^{SO}, β^{SO}, and γ^{SO}) is not extended because it was contained in Γ^{SO} before the mapping but not after; it is contained in an object (I'll call $\Gamma^{SO'}$) whose root node is Γ but $\Gamma^{SO'} \neq \Gamma^{SO}$. Recall an object SO is only extended by a mapping if the set of constituents that contain SO before the mapping is a proper subset of the constituents that contain SO after the mapping. The other input δ^{SO} is extended. Before the mapping it is only contained in itself, and after it is contain in itself as well as Δ^{SO} and Γ^{SO}. This means that Replacement Merge extends δ^{SO} by 2 in violation of the Extension Condition.

Replacement Merge also violates the Anti-Inclusion Condition. The object Δ^{SO}, which I call the new object because it isn't contained in any input, is extended (by 2) in violation of the Anti-Inclusion Condition.[14] The fact that Replacement Merge isn't a member of the Extension Condition Family is a good result because ruling out these kinds of operations was the original motivation for the Extension Condition (Chomsky, 1993).

The operation in (30), called Parallel Merge (PM), also violates the Extension Condition and the Anti-Inclusion Condition. Like the previous two operations, Parallel Merge takes two inputs and returns one (i.e. it's a 2-to-1 mapping of syntactic objects to a syntactic object).

14. The extension of the new object (Δ^{SO}) wouldn't violate the Anti-Inclusion Condition if it were the output of Replacement Merge, but it isn't the output.

(30) **Parallel Merge (PM)**



The output syntactic object is a graph with two root nodes (Δ and Γ), which I refer to as ▲. Parallel Merge extends ▲ by 1 because ▲ isn't contained in any input and contains itself in the output. One input (Γ^{SO}) is extended by one; it is only contained in itself in the input, and in the output, it is contained in itself and in ▲. The object β^{SO}, one constituent of Γ^{SO}, is also extended by one: Before the mapping β^{SO} is contained in Γ^{SO} and β^{SO}, and after, it's additionally contained in ▲.

Now for the violations. The other input δ^{SO} is only contained in itself before the mapping, but after the mapping, it's contained in two additional constituents (that is, Δ^{SO} and ▲). So one of the inputs to Parallel Merge is extended by more than one in violation of the Extension Condition. The same can be said of α^{SO}, which is a constituent of the input Γ^{SO}. Before the mapping α^{SO} is contained in itself and in Γ^{SO}, and after, the mapping it is also contained in Δ^{SO} and ▲. Parallel Merge also violates the Anti-Inclusion Condition because one constituent contained in the output is Δ^{SO}, but Δ^{SO} is neither the output nor is it contained in a input.

The way I've defined Parallel Merge is identical to the definition in Citko 2005. Citko (2005) also defines Parallel Merge as a 2-to-1 mapping, where the output is a two-peaked structure. In Citko's view and in her terms, the two rooted objects (in my terms, inputs) are combined by Parallel Merge in a way that is theoretically possible because it shares the properties of the commonly adopted operations External and Internal Merge. Citko (2005) doesn't mention the Extension Condition with respect to Parallel Merge. However, I think there is an an implied theory of licit syntactic operations in her proposal. The theory can be summarized (very briefly) as follows: a syntactic (structure-building) operation is any

mapping f such that (i) f is a 2-to-1 mapping and (ii) the output of f contains all of the inputs (i.e. f obeys the No-Tampering Condition). While (my interpretation of) Citko's theory works for her research goals, it seems to me that it permits operations that are ruled out by an intuitive conception of the Extension Condition. And because I'm concerned, in this chapter, with the properties of theories that adopt the Extension Condition in its strictest possible form, the extension-based theory I've developed so far can lead to insights that Citko's can't.

The Identity operation (Ident), which takes one input and returns that input, is interesting because it violates the Extension Condition but doesn't violate the Anti-Inclusion Condition.

(31) **Identity (Ident)**

$$\alpha \qquad = \text{Ident} \Rightarrow \qquad \alpha$$

The Identity operation doesn't extend any syntactic object. The single input (α^{SO}) is only contained in itself in both the input and in the output, so it isn't extended. And because the input is also the output, it follows that the output isn't extended either. Therefore, it's obvious that Identity violates the Extension Condition. Because Identity doesn't extend anything at all, it follows that it complies with the Anti-Inclusion Condition.[15]

2.2.3 Determinability and strict extension

In this section, I'll argue that a derivation is determinable by the Containment Procedure in (19), repeated below as (32), if and only if every operation included in the derivation is a member of the Extension Condition Family.

15. According to the definition of derivation that I gave in (14), the Identity operation can't actually derive a stage. If Identity derived stage S_i from S_{i-1}, then $S_1 = S_{i-1}$. This is ruled out by condition (iii) or by the conjunction of conditions (i) and (ii) of the definition.

(32) **Containment Procedure (repeated)**

1. Find the set of constituents contained in the final representation.
   ($Con^{final} = \{SO \mid$ final representation contains $SO\}$)
2. Remove the final representation from the result of Step 1.
   ($Con^{final} - \{\langle$final representation$\rangle\}$)

I'll call any derivation *strictly extending* if it is determinable (repeated in (33)) by the above procedure. Any operation that can derive a stage in a strictly extending derivation is a *strictly extending operation*. And a *strictly extending theory* is any theory that only permits the application of strictly extending operations.

(33) **Determinable (repeated)**

The set of intermediate representations of a derivation D = $\langle S_1, \ldots, S_n \rangle$ is *determinable from* S_n iff there is an effective procedure (i.e. an algorithm) that determines the set of intermediate representations from the information given exclusively by the final representation of D.

It's fairly easy to see that every derivation whose stages are derived by members of the Extension Condition Family are strictly extending. One property of Extension Condition Family members is that they never destroy constituents. A operation f destroys a constituent A^{SO} iff A^{SO} is contained in some input to f but A^{SO} is not contained in the output of f. If f destroys A^{SO}, then f doesn't extend A^{SO} and therefore violates the Extension Condition. So if a derivation D only includes members of the Extension Condition Family of operations, a syntactic object that is contained in the set of intermediate representations of D is contained in the final representation of D. We can conclude, then, that the Containment Procedure in (32) will never exclude an intermediate representation of D—that is, it will never undergenerate the set of intermediate representations. Put another way, a strictly extending derivation has no intermediate constituents.

The outputs of Extension Condition Family members only immediately contain themselves and their inputs; otherwise, they violate the Anti-Inclusion Condition, meaning they aren't, in fact, members of the Extension Condition Family. So if a derivation D only includes Extension Condition Family members, the final representation of D will not contain any object that was not, at some point, an input (henceforth, *never inputs*). And because inputs are always members of some derivational stage (i.e. intermediate representations), the final representation will not contain anything but itself and intermediate representations. In other words, given the final representation of D, the Containment Procedure will not return an object that is not in D's set of intermediate representations—that is, it will never overgenerate.

We have found that every derivation D that exclusively includes Extension Condition Family members derives a final representation that (i) contains every intermediate representation of D and (ii) doesn't contain any syntactic object that is not an intermediate representation of D. It follows from (i) and (ii) that every such derivation is strictly extending. Given any such derivation and its associated final representation, the Containment Procedure will accurately return the derivation's set of intermediate representations.

In contrast to members of the Extension Condition Family, all of which obey the Anti-Inclusion Condition, operations that don't obey the Anti-Inclusion Condition (e.g. Hyper, Parallel, and Replacement Merge) always create never inputs—objects that are not intermediate representations. If a derivation includes one of these operations and the never input is contained in the final representation, then the Containment Procedure will falsely return the never input. If a never input is created by some operation f and later destroyed by an operation g, then the intermediate representation that is the output to f is destroyed by g. As a result, the Containment Procedure will exclude one of the derivation's intermediate representations (i.e. it will have an intermediate constituent). We can conclude that any derivation that makes use of an Anti-Inclusion violating operation is not a strictly extending

derivation.

What if a derivation makes use of an operation that obeys the Anti-Inclusion Condition but violates the Extension Condition? The only operation that fits this bill is the Identity operation. Interestingly, it doesn't create never inputs and it never destroys constituents, so we may expect a derivation that makes use of it to be strictly extending. Whether or not such a derivation is strictly extending is moot, though. According to the definition of derivation that we are assuming, there can be no derivation that makes use of the Identity operation (because the input of an operation f can't be an element of the stage derived by f). If we amended the definition of derivation to allow for the possibility that the Identity operation could derive a stage from another, then we will get a boring result: if the final stage of a derivation is derived by Identity, then the derivation isn't strictly extending. If any number of non-final stages of a derivation (that otherwise only uses Extension Condition Family members) are derived by Identity, then the derivation is strictly extending. Therefore, if a derivation makes use of an operation that is not in the Extension Condition Family, it will either violate the Extension Condition or the Anti-Inclusion Condition. As a result, the set of objects returned by the Containment Procedure will not equal the set of intermediate representations. That is, the derivation will not be strictly extending.

I have shown that (i) every derivation that only includes Extension Condition Family members is strictly extending, and (ii) every derivation that includes an operation that isn't in the Extension Condition Family is not strictly extending. It follows from (i) and (ii) that a derivation is strictly extending if and only if it only makes use of Extension Condition Family members.

2.2.4 Interim summary

We've now developed and formalized a theory of syntactic structure-building that can clearly distinguish between operations that are strictly extending and those that are not: Only members of the Extension Condition Family are strictly extending. There is something striking about the set of strictly extending operations. Despite the fact that the conditions on inclusion into Extension Condition Family are extremely restrictive, several operations (in fact, an infinity of operations) are members of the Extension Condition Family that have never been proposed in the literature and whose existence hasn't been empirically supported. I'm thinking mostly of the class of simultaneous operations. It seems obvious that an adequate theory of syntax would rule these out, so it could be that my formulation of the Extension Condition Family isn't strict enough. However, there are operations that aren't members of the Extension Condition Family (e.g. Parallel and Replacement Merge) that have been proposed in the literature.

When it comes to the strictly extending operations that have never been proposed in the literature, it's probably the case that they should be ruled out by some independent principle(s) of grammar. In §3.1.3, I posit a constraint on the application of operations, which I call the *Single Selectee Constraint*, that rules out several of these operations (e.g Simultaneous Merge). As for the operations that appear in the literature but are not strictly extending, Chapter 3 will be devoted to cases in which these operations are used—with special attention paid to versions of Replacement Merge. I ultimately conclude that the use of non-strictly extending operations is useful to syntactic analysts who wish to maintain strict versions and longstanding versions of modules of syntactic theory such as binding theory and locality theory. However, I find that independent empirical support for the existence of non-strictly extending operations is lacking.

2.3 Strongly equivalent alternatives to strictly extending theories

I argued in the previous section that strictly extending derivations are determinable from their final representations by the Containment Procedure in (19). This means that, for every strictly extending derivation, the final representation (irreflexively) contains all of and only the intermediate representations of the derivation. Therefore, the following is true of every final representation *SO* of every strictly cyclic derivation: Every syntactic object contained in *SO* is either (i) a lexical item or (ii) an output of a Extension Condition Family member. What this means is that the set of final representations derivable in a strictly cyclic theory will always be definable without the use of derivations. In other words, every strictly derivational theory has a strongly equivalent declarative (or representational) alternative. For example, in §2.3.2, I will discuss a theory of syntax that permits a single operation Merge, which is a member of the Extension Condition Family. And then, in §2.3.4, I will prove in many more steps that the following is true of every derivable final representation *SO* of that theory: Every syntactic object contained in *SO* is either (i) a lexical item or (ii) the output of Merge. In §2.3.1, I'll provide a brief survey of derivational and declarative theories, and I'll establish exactly what I mean when I call a theory derivational or declarative. In §2.3.2 and 2.3.3, I'll introduce the details of Collins and Stabler's (2016) derivational theory, and I show that their theory is strictly extending because it only permits strictly extending operations. In §2.3.4, I'll provide a formal proof that there is a strongly equivalent declarative alternative to Collins and Stabler's (2016) derivational theory.

2.3.1 Derivational and declarative theories of syntax

Historically, classifying theories of syntax as derivational or declarative (also known as representational) has been difficult. The challenge is mainly due to disagreements in terminology. For some, a derivational theory is one that posits syntactic movement in order to account for kinds of dependencies (i.e. raising, topic/focus-dislocation, and so on). Syntactic movement

is an informal name for a kind of operation (or transformation) from syntactic representations to a representation of the same kind, where a subpart of the input representation is placed in a distinct structural position in the output representation. We can refer to this particular type of mapping as a *movement transformation*. Many theories posit mappings between syntactic representations that aren't movement transformations. For example, many formulations of Transformational Grammar defined insertion and deletion transformations that are distinguishable from movement transformations (Chomsky, 1956, 1957, 1965; Lasnik and Kupin, 1977). Declarative theories of syntax eschew movement transformations and resort to alternative mechanisms to syntactic movement for the purposes of modeling the relevant dependencies. To name a few of these mechanisms, we find feature sharing in Head-Driven Phrase Structure Grammar (Pollard and Sag, 1994), a larger set of syntactic categories and more powerful combinatory rules in Combinatory Categorial Grammar (Steedman, 1996), structure sharing in Lexical-Functional Grammar (Bresnan, 1982; Kaplan and Zaenen, 1989), and chains in Lexico-Logical Form Theory (Brody, 1995).

In a another sense of the term, a derivational theory is one that posits any kind of mapping from one syntactic representation to another of the same kind.[16] In these theories a derivation is a total or partial order of representations that form the outputs of successive mappings. The declarative counterparts of these theories don't define any mappings between syntactic representations of the same kind, which means there are no orders, and therefore no intermediate representations.

Because movement transformations are mappings between syntactic representations of the same kind, any theory that posits a movement transformation is a derivational theory in any sense of the word, but the opposite isn't necessarily true—a theory can define a set of mappings between representations that doesn't include any movement operations. Going

16. Some theories, like Lexical-Functional Grammar, posit multiple kinds of syntactic representation (i.e. A-structure, F-structure, and C-Structure, and so on), each of which encodes different types information about an expression. Additionally, there are mappings from one kind of representation to another. This not the kind of mappings that are essential to derivational theories in any sense of the term.

forward, I will consider a theory derivational if it posits any transformation at all. So, for example, Categorial Grammar defines mappings between syntactic categories (representations) (Bar-Hillel, 1953). Mappings between categories never displace arguments (because they apply to and replace contiguous strings) which means the theory doesn't define syntactic movement. Despite the fact that Categorial Grammar doesn't define a movement transformation, it is a derivational theory because it defines transformations.

To take another example, context-free grammars often form a subpart of syntactic theories of natural language (Chomsky, 1956; Hopcroft and Ullman, 1979). Context-free grammars are models that include mappings, called productions, from representations into representations. Productions aren't movement transformations, but if a theory is formulated as a context-free grammar, it is a derivational theory. Classical Transformational Grammars always had at least two components: a context-free grammar defining the representations of the base and a transformational component built on top of it. The transformational component is where syntactic movement is defined.[17]

2.3.2 A strictly extending theory

Collins and Stabler 2016 presents a formalization of minimalist syntax that is a strictly extending (derivational) theory of syntax. They define a derivation as a sequence of stages, each of which is derived from the previous by an operation (barring the initial stage, of course), and the only operation their formalization permits (Merge) is a member of the

17. Lexical-Functional Grammar also defines one level of syntactic representation, c-structure, with context-free productions (Bresnan, 1982, p.354). But Lexical-Functional Grammar doesn't have a transformational component, mapping c-structures (from the base) into c-structures; this precludes any movement transformations. If a process that generates c-structures from a context-free production rules generates intermediate syntactic representations, then I would argue that Lexical-Functional Grammar is nonderivational one sense of the term because it doesn't have syntactic movement, but I would also argue that it's derivational in another sense because it does define a mapping between syntactic representations of the same kind in the course of generating c-structures. However, Kaplan (1994, p.5) says that Lexical-Functional Grammar assumes a "descriptive, declarative, or model-based method" for assigning structures to expressions of a language. I believe that what Kaplan means is that Lexical-Functional Grammar does take the derivation of c-structures to be a part of natural language systems.

Extension Condition Family.

The article provides set-theoretic definitions of many notions that play a fundamental role in minimalist syntactic theory, and the authors show that many common principles, which are typically given as axioms, can be derived as theorems from a small set of definitions. The most salient aspect of their formalization for the purposes of this chapter is the operation Merge, which they take to be the only syntactic structure-building operation. Their definition of Merge is a member of the Extension Condition Family as defined in §2.2 with some minor changes.[18]

In addition to run-of-the-mill set-theoretic objects and operations (elements, sets, union, difference, *etc.*), the framework in Collins and Stabler 2016 includes Universal Grammar (34), which is a group of linguistic operations and objects. The operations are Select, Merge, and Transfer.[19] I'll provide definitions of Select and Merge, but since it won't be relevant, I'll omit their definition of Transfer. The kinds of primitive objects are phonological, syntactic, and semantic features, which are elements of finite sets PHON-F, SYN-F, and SEM-F, respectively. I'll assume that elements in PHON-F are phonological segments.[20] And semantic features won't play a role in this chapter, so I won't discuss them any further.

(34) **Universal Grammar** (Collins and Stabler, 2016, p.44, def.1)

Universal Grammar is a 6-tuple: ⟨PHON-F, SYN-F, SEM-F, Select, Merge, Transer⟩

Syntactic features, semantic features, and strings of phonological features are grouped together into objects called lexical items, as in (35), and a finite set of lexical items is called

18. Because Collins and Stabler's (2016) definition of immediate containment isn't reflexive, Merge does not extend its output. I'll discuss this more later when I introduce their definition of immediate containment, containment, and Merge. Importantly, the difference doesn't pose any problems to the major arguments of the chapter, which I believe are not intrinsically tied to the arbitrary choices we make when formalizing theories.

19. Collins and Stabler don't take this list of operations to be exhaustive. For example, they note that an operation Agree is likely a universal syntactic operation.

20. I'm simplifying for ease of exposition: "PHON-F may contain phonological features such as [+voiced] or [+ATR]. PHON* is the set of strings of segments, each of whose features are chosen from PHON-F" (Collins and Stabler, 2016, p.44).

a lexicon (36).

(35) **Lexical item** (Collins and Stabler, 2016, p.44, Def.2)

A *lexical item* is a triple = ⟨SEM,SYN,PHON⟩

where SEM and SYN are finite sets such that

SEM ⊆ SEM-F, SYN ⊆ SYN-F, and PHON ∈ PHON-F*.

(36) **Lexicon** (Collins and Stabler, 2016, p.44, def.3)

A *lexicon* is a finite set of lexical items.

A model of an individual grammar (or I-grammar (37)) is defined as Universal Grammar plus some lexicon. So each I-grammar has a finite set of lexical items and three operations. Technically, it also includes the sets of feature PHON-F, SYN-F, and SEM-F, but none of the operations in the model can take them or their elements as operands.

(37) **I-grammar**[21] (Collins and Stabler, 2016, p.45, def.4)

An I-grammar is a pair ⟨Lex,UG⟩ where Lex is a lexicon and UG is Universal Grammar.

The notion of lexical item token (38) is introduce to solve the problem of distinguishing copies and repetitions. This problem won't arise in this chapter, so I won't discuss it much here,[22]

(38) **Lexical item token** (Collins and Stabler, 2016, p.45, Def.5)

A *lexical item token* is a pair ⟨LI,k⟩ where LI is a lexical item and k is an integer.

In short, lexical item tokens are lexical items with an index. The index can distinguish two identical lexical items. For example, the two instances of *John* in the sentence $John_i$ *saw*

21. Collins and Stabler 2016 call these I-languages. I prefer the term I-grammar here because it leaves the term *language* for set of expressions that a grammar generates.

22. For more on copies and repetitions, I refer you to (Collins and Stabler, 2016; Collins and Groat, 2018). Occurrences (of syntactic objects) fall within the same general area as copies and repetitions.

$John_k$, where someone named John saw a different someone named John, are distinguished by their indices i and k. These indices tell the phonological and semantic modules of the grammar to interpret the two instances of *John* as distinct from each other.

A finite set of lexical item tokens is called a lexical array (39). In section §2.3.3, I'll argue that lexical arrays aren't a necessary concept and can therefore be eliminated. However, lexical arrays do play a role in Collins and Stabler 2016, so it's important to know the definition from the original article.

(39) **Lexical array** (Collins and Stabler, 2016, p.45, Def.6)

A *lexical array* (LA) is a finite set of lexical item tokens.

The recursive definition of syntactic object in (40) describes a kind of set-theoretic object that is similar to a syntactic tree (or graph-theoretic syntactic object). Condition (40a) says that a lexical item token is a syntactic object; it is an atomic unit that is, in many ways, similar to a graph with a single node. Condition 40b establishes that a syntactic object can be decomposed into other syntactic objects; this kind of object is analogous to a graph-theoretic syntactic object with multiple nodes.

(40) **Syntactic object** (Collins and Stabler, 2016, p.46, Def.7)

X is a *syntactic object* iff

a. X is a lexical item token, or

b. X is a set of syntactic objects .

You may have noticed that lexical arrays are syntactic objects, according to (40b), as a set of lexical item tokens is a set of syntactic objects.

There are two primitive relations on the set of syntactic objects. For one, there is immediate containment, which is set membership (41). The second relation is containment (the transitive closure of immediate containment) (42).

(41) **Immediate containment** (Collins and Stabler, 2016, p.46, Def.8) [adapted]

Let A and B be syntactic objects, then B *immediately contains* A iff $A \in B$.

(42) **Containment** (Collins and Stabler, 2016, p.46, Def.9)

Let A and B be syntactic objects, then B *contains* A iff

a. B immediately contains A, or

b. for some syntactic object C, B immediately contains C and C contains A.

As I mentioned in fn.18, this definition of immediate containment (and as a result, containment) is irreflexive. This sets it apart from the graph-theoretic definition of immediate contained presented in §2.1. As a consequence of this difference, the output of an operation will not contain itself, so all else remaining equal, an operation won't extend its output.[23] This isn't an issue, however. We can redefine the Extension Condition so it aligns with the arbitrary choice to define immediate containment as an irreflexive relation. In (43), I've provided a redefinition of the Extension Condition Family in order to bring it in line with a system that assumes irreflexive containment.

(43) **(Irreflexive) Extension Condition Family**

A syntactic structure-building operation is a member of the Extension Condition Family iff it is a mapping f from n syntactic objects, for $n \geq 1$, to a single syntactic object SO' such that:

(Extension Condition) For every syntactic object SO that is either (i) an input to f or (ii) contained in an input to f, f extends SO by 1 and only 1; and

(Anti-Inclusion Condition) f extends no other syntactic object.

23. This claim is complicated by the fact that Collins and Stabler (2016) define workspaces (see (45)) as sets of syntactic objects. So technically workspaces are syntactic objects. This is often an unwelcome consequence of their framework. I'll stipulate that workspaces are not syntactic objects, despite the fact that they meet the conditions for syntactic object-hood.

The only difference between the definition in (43), which I've named the Irreflexive Extension Condition Family, and the previous definition of the Extension Condition Family in (22) is the Extension Condition. An operation is a member of the Irreflexive Extension Condition Family iff the operation extends (i) all of its inputs and (ii) every constituent of its inputs by 1 and only 1. Both the reflexive and irreflexive definitions of the Extension Condition Family define the same set of operations, as long as you pair the reflexive definition of (immediate) containment with the reflexive definition of Extension Condition Family, and you pair the irreflexive definition of (immediate) containment with the irreflexive definition on Extension Condition Family.

For Collins and Stabler (2016), Merge is the only syntactic structure-building operation. It is a binary operation on syntactic objects (44). It takes any two distinct syntactic objects A,B ($A \neq B$) and returns a new set C such that $A,B \in C$. From (40b), this new set C is a syntactic object, because it's a set of syntactic objects.

(44) **Merge** (Collins and Stabler, 2016, p.47, Def.13)

Given any two distinct syntactic objects A and B, Merge(A,B) = {A,B}.

The distinctness condition is important to the formalization; if it wasn't included, then Merge could form a singleton set. Collins and Stabler (2016, p.57) define Merge this way to derive binary branching.

Merge is a member of the Extension Condition Family. It is a mapping from two syntactic objects (A and B) to one syntactic object ({A,B}). It obeys the Extension Condition: before the mapping, the inputs aren't contained in any syntactic object, and after, they are only contained in {A,B}; so the inputs are extended by one. Since containment is transitive, every object that is contained in the inputs is extended by one. It also obeys the Anti-Inclusion Condition: the output {A,B} contains only the inputs and the constituents of the inputs, so nothing else is extended, including the output itself. Given that Merge is the only syntactic structure-building operation, it follows that any derivation that only includes Merge will be

strictly extending.

I'll now introduce the the set of definitions that form the derivational component of the formalism in Collins and Stabler 2016. A derivational stage is an object in their formalism as it was in the graph-theoretic formalism in §2.1, but their stage is slightly more complex (45). It is a pair consisting of a lexical array (39) and a set known as a workspace. Informally, the operation Select derives a new stage by moving a lexical item token from the lexical array into the workspace (46). A syntactic object that is an element of a workspace (immediately contained in W) is a root in W (47).

(45) **Stage** (Collins and Stabler, 2016, p.46, def.10)

A *stage* is a pair S = ⟨LA,W⟩, where LA is a lexical array and W is a set of syntactic objects. We call W the *workspace* of S.

(46) **Select** (Collins and Stabler, 2016, p.47, def.12)

Let S be a stage in the derivation S = ⟨LA,W⟩.

If lexical item token A ∈ LA, then Select(A,S) = ⟨LA − {A},W ∪ {A}⟩.

(47) **Root** (Collins and Stabler, 2016, p.47, def.11)

For any syntactic object X and any stage S = ⟨LA,W⟩ with workspace W, if X ∈ W, X is a *root* in W.

A derivation is a sequence of stages such that each stage (apart from the initial stage) is derived from the previous stage by either Select or Merge (48). In the initial lexical array, the lexical item part of every lexical item token must be from the lexicon (48i), so a derivation is defined relative to some lexicon. A stage is derived by Select if it is exactly like the previous stage except that the new stage's workspace immediately contains a lexical item token from the previous stage's lexical array; and the new stage's lexical array doesn't have that lexical item token in it.

(48) **Derivation** (Collins and Stabler, 2016, p.49, def.14)

A *derivation* from lexicon L is a finite sequence of stages $\langle S_i,\ldots,S_n\rangle$, for $n \geq 1$, where each $S_i = \langle LA_i,W_i\rangle$, such that

(i) for all LI and k such that $\langle LI,k\rangle \in LA_1$, LI $\in$ L,

(ii) $W_1 = \{\}$ (the empty set),

(iii) for all i, such that $1 \leq i < n$, either

(derive-by-Select) for some A $\in LA_i$, $\langle LA_{i+1},W_{i+1}\rangle$ = Select(A,$\langle LA_i,W_i\rangle$), or

(derive-by-Merge) $LA_i = LA_{i+1}$ and the following conditions hold for some A,B:

(a) A $\in W_i$,

(b) either A contains B or W_i immediately contains B, and

(c) $W_{i+1} = (W_i - \{A,B\}) \cup \{Merge(A,B)\}$.

There are two ways that a stage can be derived by Merge. One, Merge can apply to a root and a syntactic object contained in *that* root (internal merge), as defined in the first disjunct of subcondition (iiib). Two, merge can apply to two roots in the workspace (external merge); this is defined in the second disjunct of subcondition (iiib).

The definition of derivation (48) rules out what Collins and Stabler (2016) call parallel merge. Their parallel merge is different from the operation Parallel Merge that I defined in §2.2.2. Their parallel merge is an instance of (binary) Merge that derives a workspace that immediately contains intersecting sets—that is, it doesn't output a single two-peaked object:

(49) **Parallel Merge**

Let W be a workspace, A $\in$ W, and B $\notin$ W. Merge(A,B) is an instance of parallel merge iff A does not contain B.

The operation Parallel Merge defined in §2.2.2 creates a graph-theoretic syntactic object with two root nodes. The differing properties of these two conceptions of Parallel Merge isn't deep. Really it follows naturally from the choice of primitive mathematical object (sets

vs. graphs). However, its important to realize that Collins and Stabler's (2016) Parallel Merge isn't an operation, as was the case in §2.2.2. Instead, it is an instance of Merge that meets certain conditions, and since it is an instance of Merge, which is a member of the Extension Condition Family, Parallel Merge doesn't violate the Extension Condition or the Anti-Inclusion Condition.

Collins and Stabler's (2016) definition of derivations has two properties that block workspaces from being derived by parallel merge. First, derivational operations apply in series, and (ii) conditions (48iiia) and (48iiib) are worded specifically to rule out instances of Merge in accordance with (49). As Collins and Stabler note, if we lift subcondition (48iiib) in the definition of derivation, parallel merge would be permitted (Collins and Stabler, 2016, p.50).

We have enough information at this point to see that Collins and Stabler 2016's formalization only permits strictly extending derivations: Every derivation is determinable from the Containment Procedure. But the task of proving that it has a strongly equivalent declarative alternative will be easier once I offer some refinements to their formalism.

2.3.3 Refining the derivational model

In the previous section, I present the derivational framework from Collins and Stabler 2016, and I argued that it is a strictly extending theory of syntax. The goal of Collins and Stabler 2016 is to provide formal definitions of common notions in minimalist syntax, not to argue for their usefulness in the framework. In this section, I argue that lexical arrays, workspaces, and Select are superfluous, and I adopt a new definition of derivation that is similar to Collins and Stabler's (2016). As you will see, the refined formalization is capable of defining all of the grammars/languages that the original can. It isn't absolutely necessary to refine Collins and Stabler's formalization to make my greater point—that is, to prove that Collins and Stabler's formalization has a strongly equivalent declarative alternative. But this version of their formalization makes the proof in §2.3.4 much simpler.

I start with the observation that there are no inherent constraints on the ordering of operations (i.e. Select and Merge). It's possible, for instance, for a derivation to begin by selecting each lexical item token from the lexical array one-by-one until the workspace is the initial lexical array: $\langle\langle \mathrm{LA}_1, \mathrm{W}_1\rangle, \ldots, \langle\emptyset, \mathrm{LA}_1\rangle\rangle$. So there is no clear reason why we need to have lexical arrays, workspaces, and Select.

Instead, we can redefine derivation as a sequence of stages such that (i) the first term in the sequence is a finite set of lexical item tokens and (ii) each stage is derived from the previous stage by Merge. In (50), I've provided a definition of derivation along these lines.

(50) **Derivation (without Select)**

A *derivation from lexicon L* is a finite sequence of stages[24] $\langle \mathrm{S}_1, \ldots, \mathrm{S}_n\rangle$, for $n \geq 1$, such that

(i) for all SO $\in \mathrm{S}_1$, SO is a lexical item token, and

(ii) for all LI and k such that $\langle \mathrm{LI}, k\rangle \in \mathrm{S}_1$, LI $\in$ L, and

(ii) for all i such that $1 \leq i < n$, the following conditions hold for some A,B:

(a) A $\in \mathrm{S}_i$

(b) either A contains B or B $\in \mathrm{S}_i$, and

(c) $\mathrm{S}_{i+1} = (\mathrm{S}_i$ - {A,B}) $\cup$ Merge(A,B)

With this refinement, we have attenuated the derivational component of the framework to derivations and stages.[25] And for now, the only syntactic operation is Merge.

It will do us some good to define the notion of a derivable syntactic object, since it will be important in discussions to follow. In short, a syntactic object is derivable if there is

24. A stage is a set of syntactic objects.

25. We can get rid of the notion of stages as well because all stages are syntactic objects. In other words, we can say that derivations are sequences of syntactic objects that meet the conditions stated in (50). Stage is simply a useful term we can use to refer to the syntactic objects in a derivation, just as lexical array is a useful term to refer to the first term in a stage.

well-formed derivation that ends in a stage that only immediately contains that syntactic object. I've stated this formally in (51).

(51) **Derivable from a lexicon**

A is *derivable from lexicon* Lex iff there is a derivation from Lex $\langle S_1,\ldots,S_n\rangle$, where $S_n = \{A\}$.

A derivable syntactic object is defined relative to some lexicon; you can also think of a syntactic object being derivable from an I-grammar = ⟨Lex,UG⟩. Going forward, D[Lex] denotes the set of syntactic objects derivable from some lexicon Lex. I'll refer to a derivable syntactic object A as the final representation of a derivation D if A is the element of D's final stage. I'll also say that a syntactic object A is derivable in n stages if there is a derivation D such that (i) A is the final representation of D and (ii) D has n stages.

2.3.4 A declarative alternative

In this section, I'll construct a declarative theory of syntax that generates the exact same set of syntactic objects that Collins and Stabler's (2016) theory derives, meaning the two theories are strongly equivalent. The redefinition of syntactic object—for the declarative theory—is provided in (52).

(52) **Representational syntactic object**

X is a *syntactic object* (SO) iff

a. X is a lexical item token, or

b. X = Merge(A,B) for some SO A,B such that

   i. A contains B, or

   ii. Neither A nor B contains the other, and there is no C such that A contains C and B contains C.

This definition of syntactic object makes use exclusively of definitions that Collins and Stabler (2016) define: Lexical item token (38), containment (42), and Merge (44). You may also have noticed that the definition of representational object borrows from Collins and Stabler's (2016) definition of (derivational) syntactic object (40) and their definition of derivation (48). A representational syntactic object, like a derivational syntactic object, is either a lexical item or set of syntactic objects. There is an important difference between the definition of representational syntactic object and the definition of a derivational syntactic object. The set of possible derivable syntactic objects is governed by the possible inputs to Merge in a well-formed derivation (e.g. at least one of the inputs must be a root; if some input isn't a root, then it must be contained in the other input; etc.). As for representational syntactic objects, the possible syntactic objects are determined by well-formedness conditions such as (52b-i) and (52b-ii). These well-formedness conditions are declarative counterparts to the derivational well-formedness conditions on possible inputs to Merge.

In order to relate specific sets of representational syntactic objects to particular languages, we'll assume that a syntactic object is only accepted by an I-grammar if the lexical item tokens in that syntactic object are from the lexicon of that grammar (53). What it means to be a lexical item token from a lexicon is defined in (54). Going forward, R[Lex] denotes the set of syntactic objects accepted by an I-grammar with a lexicon Lex.

(53) **Accepted syntactic object**

A syntactic object X is *accepted* by an I-grammar G = ⟨Lex,UG⟩ iff either

a. X is a lexical item token from Lex, or

b. for all lexical item tokens T contained in SO, T is from Lex.

(54) **Lexical item token from a lexicon**

X = ⟨LI,k⟩ is a lexical item token *from a lexicon* Lex iff LI ∈ Lex

Given any I-grammar, the set of syntactic objects derived from the lexicon (51) and the

set of representational syntactic objects accepted by the I-grammar are equivalent. This is stated more concisely in (55) (henceforth, the equivalence theorem).

(55) **Declarative/derivational equivalence theorem**

For any lexicon Lex, R[Lex] = D[Lex].

I prove the equivalence theorem in two steps. First, I prove that the set of syntactic objects derivable from an arbitrary lexicon Lex is a subset of the set of syntactic objects accepted by an I-grammar G = ⟨Lex,UG⟩ (D[Lex] ⊆ R[Lex]). Second, I prove that the set of syntactic objects accepted by an I-grammar G = ⟨Lex,UG⟩ is a subset of the set of syntactic objects derivable from Lex (R[Lex] ⊆ D[Lex]). If two sets are subsets of each other, then they are the same set.

(56) **Subtheorem 1:** For any lexicon Lex, D[Lex] ⊆ R[Lex].

Proof by induction on the number of stages in which a syntactic object A ∈ D[Lex] is derivable.

Base: The shortest possible derivation to a syntactic object is one stage long, so 1 stage is the base case. If A is derivable from Lex in 1 stage, then there is a derivation from Lex ⟨W_1 = {A}⟩; the initial and only stage has A as its sole member, so A is not formed by Merge. It follows that A must be a lexical item token from Lex. Any lexical item token from Lex is accepted by any I-grammar G = ⟨UG,Lex⟩; and thus, if A ∈ D[Lex] is derivable in one stage, then A ∈ R[Lex].

Induction: Let subtheorem 1 be true for all syntactic objects derivable in fewer than n stages, for $n \geq 2$. Let A ∈ D[Lex] be derivable in n stages.

I. Because lexical item tokens are always derivable in 1 stage and A isn't derivable in 1 stage, it must be the case that A = Merge(X,Y) (i.e. A is not a lexical item token), where X and Y are derivable syntactic objects. If a syntactic object α contains another β, then β is derivable in fewer stage than α. X and Y are contained in A;

therefore X and Y are derivable from Lex in fewer than n stages. Given that X and Y are derivable in fewer than n stages, X and Y are accepted representational syntactic objects (X,Y $\in$ R[Lex]).

II. There are two cases of Merge(X,Y) to consider (*i.e.* internal and external merge). It follows from each case that A is a representational syntactic object:

(Internal Merge) If X contains Y, then it follows from one well-formedness condition (52b-i) that A is a representational syntactic object.

(External Merge) If X does not contain Y, then A is a representational syntactic object iff there is no syntactic object α such that both X and Y contain α (i.e. it meets the other well-formedness condition (52b-ii), which is the declarative analog to the derivational ban on parallel merge). As was noted in §2.3.2, parallel merge is ruled out by the definition of derivation in (48) and the equivalent redefinition in (50).

III. All syntactic objects that are derivable from Lex contain only lexical item tokens from Lex (50). Because A is derivable from Lex (by the inductive hypothesis), all lexical item tokens contained in A are from Lex. Therefore A is a representational syntactic object that contains only lexical item tokens from Lex; that is, A $\in$ R[Lex].

We have shown for any syntactic object A and lexicon Lex, if A is derivable from Lex, then A is accepted by a grammar with Lex: For any lexicon Lex, D[Lex] $\subseteq$ R[Lex] ■

(57) **Subtheorem 2:** For any lexicon Lex, R[Lex] $\subseteq$ D[Lex].

Proof by induction on the number of syntactic objects contained in A $\in$ R[Lex].

Base: The base case is that A contains zero syntactic objects. If A contains zero syntactic objects, then it is a lexical item token from Lex; so A is derivable from Lex in one stage: $\langle W_1 = \{A\}\rangle$.

Induction: Let subtheorem 2 be true for all syntactic objects containing fewer than n syntactic objects, for $n \geq 2$.[26] Let A $\in$ R[Lex] contain n syntactic objects.

I. It must be the case that A = Merge(X,Y), where X and Y $\in$ R[Lex]. If an SO α contains another β, then β contains fewer syntactic objects than α. A contains X and Y; therefore X and Y contain fewer than n syntactic objects. Given that X and Y contain in fewer than n syntactic objects, X,Y $\in$ D[Lex].

II. There are only two licit classes of derivations to A = Merge(X,Y): one class of derivations that ends in an instance of internal merge [IA] and one class that ends in an instance of external merge [EM].

A is a representational syntactic object (by the inductive hypothesis). So it is either the case that Y is contained in X (52b-i), and therefore, A has a derivation in [IA] (i.e. the last stage in any derivation of A is derived by Internal Merge). Or, neither X nor Y contains the other and there is no syntactic object α such that both X and Y contain α (52b-ii), and therefore A has a derivation in [EA] (i.e. the last stage in any derivation of A is derived by External Merge).

We have shown for any syntactic object A and lexicon Lex, if A is accepted by a grammar with Lex, then A is derivable from Lex: For any lexicon Lex, R[Lex] $\subseteq$ D[Lex] ■

For any lexicon Lex, D[Lex] $\subseteq$ R[Lex] and R[Lex] $\subseteq$ D[Lex]. Therefore, for any lexicon Lex, D[Lex] = R[Lex]. In plain terms, the declarative theory is strongly equivalent to the strictly extending model.

26. There aren't any syntactic objects that contain only one syntactic object because there are no syntactic objects that are singletons.

2.4 When a declarative theory won't do

I hope I've achieved my ultimate goal of convincing you that strictly extending theories of syntax, at least as far as structure-building is concerned, are odd. To close out the chapter, I'd like to discuss what we should take away from the arguments presented throughout it, and then, I'll tease the next chapter, which will investigate non-strictly extending analyses of syntactic phenomena and ask whether positing non-extending operations are justified empirically.

The two major sections of this chapter were devoted to arguing the following:

1. Every derivation is determinable by the Containment Procedure if and only if the derivation only includes members of the Extension Condition Family.

2. Every syntactic theory that only permits strictly extending derivations has a strongly equivalent declarative alternative.

It follows from (1) and (2) that the derivational vs declarative debate is pointless if the Extension Condition Family of operations (or a subset of the family) are the only operations necessary in an adequate theory of syntax. However, if we find that our theory of syntax needs to permit nonmembers of the Extension Condition Family (i.e. non-extending operations), then we can conclude that a derivational theory is necessary. Of course, if we do find that non-extending operations are necessary, then we must conclude that non-strictly extending, derivational theories of syntax are necessary. As I note in §2.2.2, non-extending operations have been proposed in analyses of several empirical phenomena (e.g. anti-reconstruction effects, order-preserving movement, quantifier float, etc.). In the next chapter, I will discuss the reasons why non-extending operations are posited.

CHAPTER 3

ON THE USE OF NON-EXTENDING OPERATIONS

In Chapter 2, I argued that theories that only permit operations in the Extension Condition Family, which I call strictly extending theories, always have strongly equivalent non-derivational (or declarative) alternatives. I also demonstrated that one of the most utilized formalizations of a minimalist syntactic framework (namely, Collins and Stabler (2016)'s) (i) is strictly extending and (ii) has a strongly equivalent declarative alternative.

Not every minimalist framework is strictly extending. For example, the framework used in Richards 1997, 1999, 2001 to account for order-preserving syntactic movement permits instances of Replacement Merge, which is more widely known as *tucking-in* in the context of Richards' theory. Replacement Merge, as I show in Chapter 2 (§2.2.2), is not a member of the Extension Condition Family, so by definition, Richards' framework is not a strictly extending theory of syntax. Replacement Merge has also been proposed under the name *late adjunction* to account for anti-reconstruction effects (Lebeaux, 1991; Epstein et al., 1998), quantifier float (Bošković, 2004), and *exactly*-stranding (Zyman, 2022). Furthermore, structure-removing operations (e.g. Remove and Exfoliation) have been used in analyses of object shift (Heck, 2016) and restructuring phenomena (Müller, 2020; Pesetsky, 2021).

In this chapter, I argue that non-extending operations, like Replacement Merge, are posited when an empirical phenomenon appears to violate another architectural component of the theory of syntax. The analyst can maintain the simplicity of that other component of the theory by permitting a greater number of syntactic operations, some of which may be considered more complex.[1] As Chomsky (1970, p.186) writes, 'In general, it is to be expected that enrichment of one component of the grammar will permit simplification in other parts'. In §3.2, I consider the case of multiple wh-movement in Bulgarian. Bulgarian multiple wh-movement involves crossing dependencies. In general, crossing dependencies

1. See, for example, Freidin's (1999) criticisms of (what I call) Replacement Merge and Parallel Merge.

pose a problem for any strictly extending theory of syntax because they present a tension between extension and locality conditions on movement. In §3.3, I look at late adjunction and its function in an account of anti-reconstruction effects. Anti-reconstruction effects are puzzling within an strictly extending theory because they pose a problem for a longstanding theory of binding.

While the case studies in this chapter are meant to demonstrate challenges to strictly extending theories, it's clear that these challenges can be overcome. However, the point I would like to get across is that the two classes of theories will always lead an analyst to associate grammatical sentences with different syntactic derivations. This means that, in theory, a strictly extending analysis and a non-strictly extending analysis should make different predictions. Interestingly, for each of the case studies, I can think of no way of testing the differing predictions.

The chapter is organized as follows. In §3.1, I add to the graph-theoretic framework from Chapter 2 in order to restrict the set of possible derivations and final representations. Without these additions, the framework would lack most of the tools necessary for the development of analyses of empirical phenomena. The additions are constraints and conditions on the application and output of syntactic operations. They will play a crucial part in the case studies of multiple wh-movement and anti-reconstruction effects, which are presented in §3.2 and §3.3, respectively. In §3.4, I conclude the chapter with a summary of the arguments and a look ahead to the next chapter.

3.1 Assumptions and definitions

In this section, I will introduce the mechanisms necessary to account for local syntactic dependencies formed by structure-building operations and restricted by featural requirements. Then I will propose some locality constraints on the application of syntactic operations. Finally, I will conclude with some fairly informal thoughts on the nature of feature satisfaction

in syntactic theory. Every definition in this section is compatible with the graph-theoretic formalization I presented in §2.1.

3.1.1 Feature-based operations

As of yet, there is nothing in the framework I've adopted that places fine-grained, featural requirements on structure-building operations. That is, every operation we've considered has applied to any n syntactic objects without any further restriction on the category or featural specifications of those categories. Going forward, I'll assume conditions on the applicability of operations that refer to the formal featural specifications of the input object(s) and their constituents. In order to do this, I've included formal features to the definition of syntactic object (in bold).

(1) **Syntactic Object (with features)**

A *syntactic object* is a tuple $\langle N, D, FF\rangle$, where N is a finite set of nodes, D is a non-strict partial order (i.e. a reflexive, asymmetric, and transitive relation) in N x N ($= \{\langle x, y\rangle \mid x, y \in N\}$), and **FF is a pair** $\langle Basic, \ell\rangle$ such that the following conditions hold:

(Connected Graph Condition) $N \times N = O^*$,
where $O^* = \{\langle x'_{\in N}, y'_{\in N}\rangle \mid \exists z'_{\in N}(\langle x', z'\rangle \in D \wedge \langle y', z'\rangle \in D)\}^*$.

This definition of syntactic object is identical to the definition from Chapter 2, except a pair of formal features (FF) has been added. You can think of the formal features of a syntactic object as its *label*, as the term has typically been used since Chomsky 1995. FF is a pair that is defined as follows:

(2) **Formal Features**

The set of *formal features on syntactic object* A^{SO} (denoted FF^A) is a pair $\langle Basic, \ell \rangle$, where

a. *Basic* is a non-empty, finite set of syntactic features such that $Basic \subseteq$ SYN-F (i.e. the universal set of syntactic features), and

b. ℓ is a sequence of syntactic features $\langle F_1, \ldots, F_n \rangle$, for $n \geq 0$, such that every $F_i \in$ SYN-F.

The first term of FF, called *Basic*, is a set of features drawn from the universal set of syntactic features SYN-F. The features in *Basic* are the syntactic object's categorial features (i.e. N, V, D, T, ...), φ-features (i.e. person, gender, number), and any of its licensing features (i.e. WH, Top, Σ, ...). Every syntactic object contains a nonempty *Basic* feature set.

In addition to the set of *Basic* features, syntactic objects bear a (potentially empty) sequence of selectional features ℓ, following ideas of Stabler (1997); Adger (2003, 2010); Müller (2010); Merchant (2019); Zyman (2024). If a syntactic object A^{SO} bears a sequence ℓ, I'll refer to ℓ as A^{SO}'s *selectional list*. Following Heck and Müller (2007), I notate selectional features with bullets (•F•), but I assume that a selectional feature is a selectional feature by virtue of being a term of ℓ—that is, in SYN-F, there aren't two features F and •F•. If a syntactic object A^{SO} bears a Basic feature F (i.e. A^{SO}'s Basic feature set contains F), I indicate this with a subscript F (A^{SO}_F); and if the first term in A^{SO}'s' selectional list is •F•, I indicate this with a subscript •F• ($A^{SO}_{\bullet F \bullet}$). At times, for clarity, I might subscript an object's entire selectional list as a stack:

(3) $A^{SO}_{\begin{bmatrix} \bullet F \bullet \\ \bullet G \bullet \\ \bullet H \bullet \end{bmatrix}}$

In 3, you can see the shorthand for a syntactic object A^{SO} with a selectional list $\ell =$

$\langle F, G, H\rangle$, where •F• is the first term in A^{SO}'s selectional list (i.e. •F• is at the top of the stack) and •H• is the last term of A^{SO}'s selectional list (i.e. •H• is at the bottom of the stack).

Formal features on syntactic objects allow us to further restrict the conditions on the application of syntactic operations. In particular, it allows us to define operations that can only apply when the inputs contain objects with certain featural specifications. For example, we can capture the varying c(ategory)-selectional requirements on predicates, like the requirement that *devour* have a nominal argument (4a)—and not a prepositional, adjectival, or clausal complement (4b).

(4) a. The students devour the books.

b. *The students devour {on the teacher/happy/that it's raining}.

I assume that operations place a syntactic object bearing a selectional feature •F• (for any F ∈ SYN-F) in a sisterhood relation with one or more objects that bear a matching *Basic* F feature. I've stated this restriction on licit operations, which I call the Sisterhood Condition, in (5).

(5) **Sisterhood Condition**

Let f be a syntactic operation and $f(z_1, \ldots, z_n) = y$. f is licit iff

(i) there is some feature $\mathrm{F}_{\in \text{SYN-F}}$ and syntactic object α^{SO} such that (a) z_1 contains α^{SO} and (b) the first term of α^{SO}'s selectional list is F; and

(ii) for every input z_i ($\neq z_1$), there is a syntactic object β^{SO} such that (a) z_i contains β^{SO}, (b) β^{SO} bears the basic feature F, (c) α^{SO} and β^{SO} are sisters in y, and (d) for every input z_k (i.e. $z_1, \ldots, z_n$), α^{SO} and β^{SO} are not sisters in z_k; and

(iii) if $n = 1$, there is a syntactic object γ^{SO} such that (a) z_1 contains γ^{SO}, (b) γ^{SO} bears the basic feature F, (c) α^{SO} and γ^{SO} are sisters in y, and (d) α^{SO} and γ^{SO} are not sisters in z_1; and

(iv) for every syntactic object δ^{SO} that is contained in y, if δ^{SO} doesn't bear the basic feature F, then α^{SO} and δ^{SO} are not sisters in y.

According to the definition of derivation in (ch.2, ex.14), a syntactic operation is any mapping from n syntactic objects to exactly one syntactic object. The Sisterhood Condition further restricts the set of mappings to those that create new sisterhood relations between syntactic objects with matching selectional and basic features, where sisterhood is defined as follows:

(6) **Sisterhood**

Two syntactic objects A^{SO} and B^{SO} are *sisters in* C^{SO} iff there is a syntactic object D^{SO} such that

a. C^{SO} contains D^{SO}, and

b. D^{SO} immediately contains A^{SO} and B^{SO}, and

c. $A^{SO} \neq B^{SO} \neq C^{SO}$.

The notion of sisterhood as defined above should be very familiar, so I'll move on without considering examples.

Let's discuss the particulars of the Sisterhood Condition. Clause (i) of the Sisterhood Condition requires the operation's (f) first input (z_1) contains a syntactic object α^{SO} that bears a selectional feature •F•. I'll call α^{SO} *f's selector* and •F•, *f's trigger*. Clause (ii) says that every other input must contain a syntactic object that bears a *Basic* feature F. If a syntactic object meets these criterion, it is (one of) *f's selectee(s)*. In these cases, I might say that a *Basic* feature F *matches* a trigger •F•. In addition, (iic-d) requires that f's selector and selectee(s) are not sisters before the mapping and are sisters after the mapping.

Clause (iii) of the Sisterhood Condition covers the case of a unary syntactic operations (i.e. when f takes only one input z_1). In this case, the single input must contain both f's selector α^{SO} and f's selectee(s) γ^{SO}. For every γ^{SO} (i.e. selectee), α^{SO} and γ^{SO} are distinct from each other; and α^{SO} and γ^{SO} must be sisters after the mapping and must

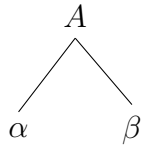
not be sisters before the mapping (iiic-d). It follows from the definition of sisterhood (in particular, (6c)) that the selector (α^{SO}) is distinct from its selectee(s) (γ^{SO}) because a syntactic object can't be a sister to itself (i.e. sisterhood is is not a reflexive relation). I'd also like to note that nothing in the definition of the Sisterhood Condition requires every syntactic object bearing the relevant *Basic* feature F must be a sister to the selector $\alpha^{SO}_{\bullet F \bullet}$ in the output of an operation. Given all other conditions are met, an operation f is licit if some input to f contains two objects bearing *Basic* feature F and only one of the two syntactic objects is sister to f's selector in the output of f.

There are several interesting consequences of clause (iv) that I won't explore here. The most important consequence for the purposes of this chapter is that clause (iv) rules out any mapping that creates sisterhood relations between the selector and syntactic objects that don't bear a matching *Basic* feature. Nothing in clauses (i)-(iii) rules out such a mapping.

In Chapter 2, I presented a syntactic operation that I call Binary Merge (repeated in 7).

(7) **Binary Merge (repeated)**

α β =BinMerge⇒

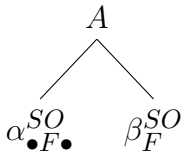


This definition of Binary Merge is featurally underspecified. To avoid any confusion, I'll make it clear when a syntactic object does or does not bear a feature. If an object has a subscript feature (e.g. •F• or F), then it bears that feature; otherwise, the object doesn't bear the feature. So it's clear that not every instance of Binary Merge, as it's defined in (7), meets the Sisterhood Condition. However, we can define Binary Merge, as follows, in order to make it clear which instances of Binary Merge will meet the Sisterhood Condition:

(8) **Binary Merge (with features)**

For some F $\in$ SYN-F[2],

$\alpha^{SO}_{\bullet F \bullet}$ β^{SO}_{F} =BinMerge⇒



According to this feature-based definition of Binary Merge, the operation applies to any two syntactic objects α^{SO} and β^{SO} as long as there is some syntactic feature F (from the set of universal syntactic features SYN-F) such that F is the first term of α^{SO}'s selectional list and F is one of β^{SO}'s *Basic* features. Binary Merge maps these two objects to a single syntactic object A^{SO} that immediately dominates both α^{SO} and β^{SO}. It follows that α^{SO} and β^{SO} are sisters (with matching features) in the output. And more, it follows from the fact that a syntactic object can't irreflexively contain itself that α^{SO} and β^{SO} are not sisters in any of the inputs. That is to say, if the input α^{SO} contains β^{SO}, then it's not possible for α^{SO} and β^{SO} to be sisters in α^{SO}; we arrive at the same conclusion if α^{SO} is contained in β^{SO}. Therefore, every instance of (feature-based) Binary Merge will meet the Sisterhood Condition.

Given that the output is a syntactic object, it must be a tuple $\langle N, D, FF \rangle$. We can read off the definition what N and D are (at least partially), but what is FF? What are the formal features of the output of Binary Merge? I'll assume a version of what-selects-projects. Because α^{SO} bears the selectional feature •F• that triggered Binary Merge, A^{SO} bears α^{SO}'s formal features minus the selectional feature •F•.

2. Allowing for any F ($\in$ SYN-F) to trigger Binary Merge (aka External Merge) is probably too liberal. We probably want to restrict the triggers of this operation to category features (c-selection) and lexical features (l-selection). As it stands, SYN-F isn't structured in any way.

(9) **Projection Condition**

Let f be a syntactic operation that is conditioned by a trigger •F• on f's selector α^{SO}, let α^{SO}'s formal features be $\langle Basic, \ell = \langle F_1, \ldots, F_n \rangle \rangle$, and let y be the output of f. For every A^{SO} that irreflexively contains α^{SO} in y, A^{SO}'s formal features are $\langle Basic, \ell' = \langle F_2, \ldots, F_n \rangle \rangle$.

In brief, the Projection Condition says that a licit operation f's selector α^{SO} passes its formal features to every irreflexively containing constituent A^{SO} in the output, except A^{SO}'s selectional list doesn't include f's trigger (i.e. the first term of α^{SO}'s selectional list). You can find similar versions of this idea in Merchant 2019; Zyman 2024.

In (8), we redefined Binary Merge so that it complies with the Sisterhood Condition, but we hadn't yet defined the Projection Condition. So let's consider how Binary Merge needs to be defined, in accordance with the Projection Condition, when the selector bears a selectional list with two. BinMerge's selector in this case will be $\alpha^{SO} = \langle N, D, FF = \langle Basic, \ell = \langle F, G \rangle \rangle \rangle$. The first term on α^{SO}'s selectional list is the feature •F•, so BinMerge's selectee β^{SO} has to bear the *Basic* feature F in order for the operation to apply.

(10) **Binary Merge (with Projection)**

For some F ∈ SYN-F,

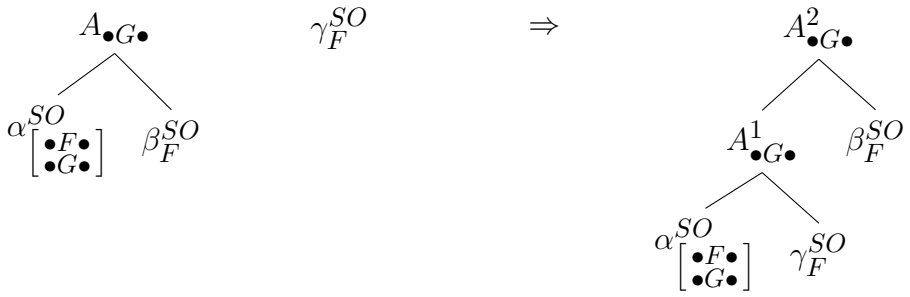
$$\alpha^{SO}_{\left[\begin{smallmatrix}\bullet F\bullet\\ \bullet G\bullet\end{smallmatrix}\right]} \qquad \beta^{SO}_{F} \qquad \Rightarrow \qquad \begin{array}{c} A_{\bullet G\bullet} \\ \alpha^{SO}_{\left[\begin{smallmatrix}\bullet F\bullet\\ \bullet G\bullet\end{smallmatrix}\right]} \quad \beta^{SO}_{F} \end{array}$$

The output of the above instance of Binary Merge object is A^{SO}. As is indicated by the subscript •G• on A^{SO}'s root node, the first term in A^{SO}'s selectional list is •G•, which is the second (and last) term of the operation's selector's (α^{SO}) selectional list. According to the definition of the Projection Condition, the FF of A^{SO} is $\langle Basic, \ell' = \langle G \rangle \rangle$—that is, the

Basic features of α^{SO} and the selectional list of α^{SO} minus the first term.

Binary Merge is member of the Extension Condition Family of operations, and for any member of that operation family, it's the case that the operation's selector will be contained in one and only one syntactic constituent (namely, the operation's output). So when in comes to Extension Condition Family members, the formal features of the operation's selector will only project to a single object. In contrast, several non-Extension Condition Family members result in projection of formal features to several syntactic objects. For clarity, I'll discuss an instance of Replacement Merge, which is not a member of the Extension Condition Family.

(11) **Replacement Merge (with features)**



The operation in (11) takes A^{SO} as its first input and β^{SO} as its second input. The operation's selector isn't the first input itself; it is α^{SO}, which is contained in the input. The operation's selectee is γ^{SO}—the second input itself. The operation's selector α^{SO} bears the selectional list $\langle F, G\rangle$. Two of the outputs constituents irreflexively contain the operation's selector; they are $A^{2^{SO}}$ and $A^{1^{SO}}$. According to the definition of the Projection Condition, α^{SO}'s formal features (minus the first term of its selectional list) need to project to each of these objects, so the formal of features of both $A^{2^{SO}}$ and $A^{1^{SO}}$ are projected from α^{SO}.

A potential issue can rise from the definition of the Projection Condition within a framework that allows for a fairly unrestricted set of syntactic operations. Consider the following derivation, which starts with an instance of Binary Merge in (12).

(12)

The instance of Binary Merge has a selector $\alpha^{SO}_{\bullet F\bullet}$ and a selectee is $\beta^{SO}_{F,\bullet G\bullet}$. The operation applies, and as a result, derives a stage S_i with the output $A^{SO}_{\bullet G\bullet}$ as a member. Crucially, the previous operation's selectee $\beta^{SO}_{F,\bullet H\bullet}$ bears a selectional feature •H•, which can trigger a operation. What happens if an operation is triggered by that feature on $\beta^{SO}_{F,\bullet H\bullet}$? You can see the result in (13).

(13)

When the Replacement operation above is applied, the formal features of the operation's selector $\beta^{SO}_{F,\bullet H\bullet}$ project to every containing constituent: Those constituents are $\mathrm{B}^{1^{SO}}_{F}$ and $\mathrm{B}^{2^{SO}}_{F}$. So the constituent immediately containing the latest operations selector has the same *Basic* features as the selector ($\beta^{SO}_{F,\bullet H\bullet}$), but also the constituent immediately containing the previous operation's selector $\alpha^{SO}_{\bullet F\bullet}$ has the same *Basic* features as $\beta^{SO}_{F,\bullet H\bullet}$.

I don't think this is a good result. It's easy to argue why. If a theory permitted such a configuration in such a situation, then the theory would likely overgenerate. For example, imagine that the verb *wonder* selects a $\mathrm{CP}_{\bullet wh\bullet}$ *Sue bought what*, and subsequently, wh-movement applies to the $\mathrm{CP}_{\bullet wh\bullet}$. In accordance with the Projection Condition, the result would be the C(P) in (14a). At this stage of the derivation, a verb, like *knows*, which bears a selectional feature •C•, could trigger Binary Merge, resulting in the (eventual) unacceptable

sentence in (14b).

(14) a. $\lfloor_{C(P)}$ wonders what Sue bought]

b. *John knows$_{\bullet C\bullet}$ $\lfloor_{C(P)}$ wonders what Sue bought].

Note that this overgeneration problem doesn't arise in strictly extending theories because the wh-movement (i.e. Replacement Merge) isn't permitted to apply after *wonder* and the C(P) combine. The syntactic object corresponding to the string *wonder what Sue bought* in (14a) will bear the *Basic* (category) feature V, and therefore *know* and *wonder what Sue bought* can't combine.

However, we aren't limiting ourselves to strictly extending theories; we are also considering non-strictly extending theories, so we need some way to curb this kind of overgeneration. I propose the constraint Embed Last in (15) that blocks the application of so-called embedding operations under certain conditions.

(15) **Embed Last Constraint**[3]

a. Let S_i be a derivational stage equal to $\{SO_1, \ldots, SO_n\}$; and

b. let F be the set of syntactic operations; and

c. let $f_{\in F}(SO, \ldots)$ be any operation that is defined over the elements of a subset of S_i where SO is the first input (i.e. SO must contain the selector); and

d. let $g_{\in F}(\ldots, SO, \ldots)$ be any operation that is defined over the elements of a subset of S_i where SO is not the first input (i.e. SO must not contain the selector).

For every SO in S_i, if there is an operation $f_{\in F}(SO, \ldots)$ that is defined at S_i, then for all $g_{\in F}(\ldots, SO, \ldots)$, $g_{\in F}(\ldots, SO, \ldots)$ doesn't derive S_{i+1} from S_i.

3. There are many ways in which the Embed Last Constraint is similar to the notion of Featural Cyclicity (Richards, 1997, 1999, 2001).

(i) **Featural Cyclicity** (Richards, 1999, p.127, ex.1)
A strong feature must be checked as soon as possible after being introduced into the derivation.

In fact, it may be the case that one subsumes the other or even that they are equivalent.

The constraint, which I call the Embed Last Constraint, prevents a syntactic object α^{SO} from being a selectee when it can be a selector. A little more formally, the constraint prevents the application of an operation f that embeds an object α^{SO} in β^{SO} if an operation g can apply to α^{SO} and g's selector is contained in α^{SO}. The constraint derives some ordering between operations.[4]

In our hypothetical derivation, $wonders_{V,\bullet C\bullet}$ merged with $\text{CP}_{\bullet wh\bullet}$; V projected, giving us $[_{\text{VP}}\ wonders_{\bullet C\bullet}\ \text{CP}_{\bullet wh\bullet}]$. Then •wh• on $\text{CP}_{\bullet wh\bullet}$ triggered wh-movement (i.e. Replacement Merge). The resulting object correlates to a string *wonder what Sue bought* in (14a) and the formal features of $\text{CP}_{\bullet wh\bullet}$ project up to every object that irreflexively contains $\text{CP}_{\bullet wh\bullet}$, resulting in an abstract representation $[_{C}\ wonders_{V,\bullet C\bullet}\ \text{CP}_{\bullet wh\bullet}]$. This derivation is ruled out by the Embed Last Constraint because the first operation—the operation that outputs $[_{V}$ *wonders* CP]—isn't permitted to apply. At that stage of the derivation, there is an operation (namely, wh-movement) that can be triggered by a feature on $\text{CP}_{\bullet wh\bullet}$, which is contained in CP, and therefore •C• on $wonders_{V,\bullet C\bullet}$ cannot trigger Binary Merge with $\text{CP}_{\bullet wh\bullet}$. What this means is that there can be no C(P) *wonder what Sue bought*, as in (14a).

The Embed Last Constraint doesn't require the operation triggered by the feature on C to apply; however, it may be the case that this operation does apply. If it does, and as a result, there is no possible operation at the next derivational stage where $\text{CP}_{(\bullet wh\bullet)}$ contains the selector, then $wonders_{V,\bullet C\bullet}$ can combine with $\text{CP}_{(\bullet wh\bullet)}$, resulting in (16a).

(16) a. $[_{V(P)}$ wonders what Sue bought]

b. *knows$_{\bullet C\bullet}$ $[_{V(P)}$ wonders what Sue bought].

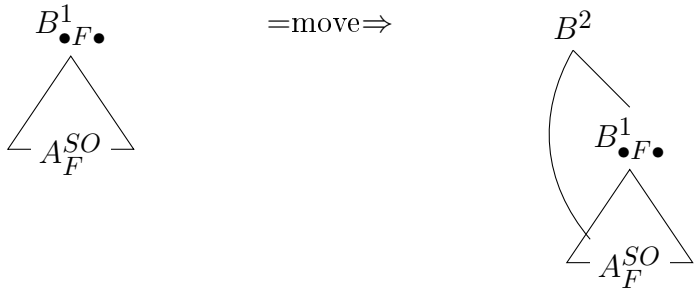
Because the output of this instance of Binary Merge is a V(P), the verb $know_{\bullet C\bullet}$ can't combine with it, as in (16b). Otherwise it would violate the Sisterhood Condition.

4. We could derive these ordering relations by requiring selectees to bear empty selectional lists. In fact, this is the typical approach (Collins and Stabler, 2016; Merchant, 2019). As I will show in §3.2, the two approaches aren't equivalent, and there are advantages to adopting the Embed Last Constraint over the alternative.

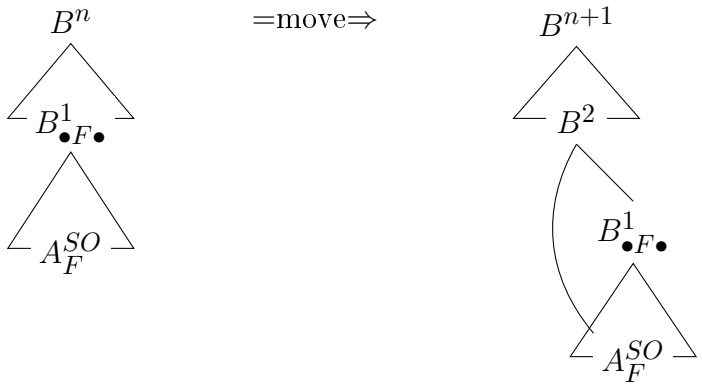
3.1.2 Syntactic movement

In a derivational theory, syntactic movement is any operation that creates a dependency between two positions in a syntactic representation, where the two positions are occupied by the same syntactic object. In a typical formulation, the two positions related by movement are in a c-command relation: one position α^2 is sister to an object β^{SO} that contains the other position α^1. Consider the operations in (17) and (18). Each operation has a selector is $B_{\bullet F \bullet}$ and selectee A_F.

(17) *Extending Move*



(18) *Non-Extending Move*



In both of the above cases, a syntactic object A^{SO}_{F} moves to become sister to some other syntactic object $B^{SO}_{\bullet F \bullet}$ that contains A^{SO}_{F}. As a result, the syntactic object A^{SO} is in (at least) two positions in the resulting object—that is, it is both sister to B^{SO} and contained in B^{SO}.

There is an important difference between the operations that derive the two objects above. The operation (17), which I'll call *extending move*, is a unary operation that is a member of the Extension Condition Family. The selector is itself the input ${B^1}^{SO}$, the selectee A^{SO} is contained in the selector, and every object contained in the input is extended by 1. The operation (18), *non-extending move*, is similar to extending move. It is a unary operation and the selector ${B^1}^{SO}$ contains the selectee A^{SO}. But non-extending move is not a member of the Extension Condition Family for several reasons. For one, the object ${B^2}^{SO}$ that immediately contains the selector and selectee in the output is (i) not the output and (ii) not in the input. If ${B^2}^{SO}$ were in the input, then the selector and selectee would necessarily be sisters in the input; and according to the Sisterhood Condition in (5), non-extending move would be an illicit operation. Since ${B^2}^{SO}$ is not the output nor in the input, its very existence in the output entails a violation of the Anti-Inclusion Condition. Furthermore, because ${B^2}^{SO}$ isn't contained in the input, every object X^{SO} that is irreflexively contained in ${B^2}^{SO}$ is not extended according to the definition of extension (ch.2, ex.21, def.1) because the set of objects that contains X^{SO} in the input is not a proper subset of the set of constituents that contain X^{SO} in the output.

Syntactic movement (often referred to as Internal Merge in minimalist circles) is usually formalized as a binary operation. The two objects are the thing moved (henceforth, the mover) and the thing moved to (henceforth, the attractor). I've conceived of syntactic movement as a (set of) unary operations because no other kind of operation is compatible within the framework I've developed so far. My definition of derivation in (ch.2,ex.14) only permits operations that apply to (i.e. take as inputs) members of the current derivational

stage S. And because mover(s) are never members of S (as can be seen for both extending move and non-extending move), it isn't possible for the mover to be an input to a syntactic movement operation. Furthermore, in theories that permit non-extending move, there are instances of syntactic movement where neither the attractor nor the mover are members of the stage at which the operations apply. In other words, it's only possible to define syntactic movement over a single object.

I'll refer to the input syntactic object to a unary operation as the *domain of syntactic movement*. As I mentioned at the top of this section, it is typically assumed that the mover (in canonical syntactic movement operations) targets a c-commanding position with the domain of movement. I capture this assumption by constraining licit unary operations to those where the selector contains the selectee(s), which I've named the Upward Movement Constraint (19).

(19) **Upward Movement Constraint**

For every unary operation f, f is licit only if f's selector irreflexively contains f's selectee(s) in the input to f.

In the terminology of syntactic movement, the Upward Movement Constraint says that the mover must be irreflexively contained in the attractor. Together with the Sisterhood Condition, the Upward Movement Constraint derives the c-command requirement on syntactic movement.

The Upward Movement Constraint is redundant in a strictly extending theory. Any operation that violates the constraint is not a member of the Extension Condition Family. This should also be clear from the diagram of extending move in (17). With extending move, the domain of movement is the attractor. In contrast, the domain of non-extending move irreflexively contains the attractor.

3.1.3 Locality constraints

It has been well-established for decades that there are constraints on syntactic movement (Chomsky, 1965, 1973; Van Riemsdijk and Williams, 1988; Rizzi, 1990; Fitzpatrick, 2002). In this section, I address some of these constraints. The way I've defined the constraints below are slightly different from the traditional conceptions, but they are certainly in line with the spirit of the traditional constraints. Furthermore, the constraints, as I've defined them, are not technically specific to movement operations. When relevant, I'll point out how the constraints affect non-movement operations (e.g. Binary Merge).

As far as I know, there is a universal assumption that an operation can move one and only one object at a time. If a trigger •F• matches features on more than one object—let's say α^{SO}_{F} and β^{SO}_{F}—then either α^{SO}_{F} or β^{SO}_{F} can be the selectee (at the current stage), but not both. We can capture this theoretic intuition with the *Single Selectee Constraint* in (20).

(20) **Single Selectee Constraint**

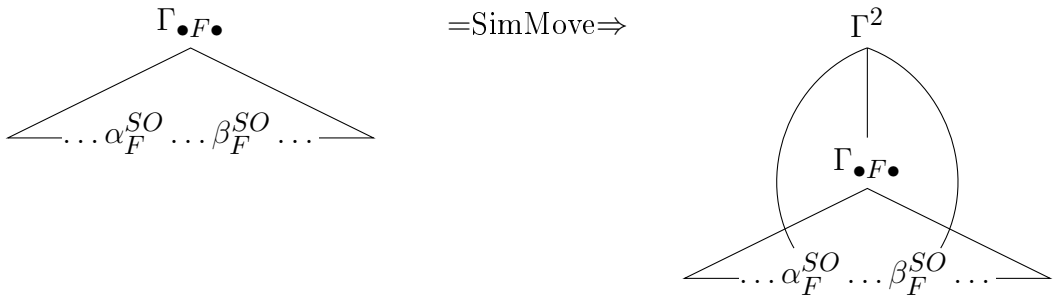
For every syntactic operation f, f is licit only if it has one and only one selectee.

The Single Selectee Constraint has several interesting consequences. For one, the constraint rules out operations in what I've called the Simultaneous Merge class (ch.2, ex.26); whether or not this is a wanted consequence is an empirical matter, but as I mentioned in §2.2.1, I haven't found any empirical support for the existence of Simultaneous Merge operations[5]. Second, because the Single Selectee Constraint is in effect for all syntactic operations, not just movement operations, it follows from the constraint that all syntactic objects will be binary branching—that is, the selector will have one and only one sister. Finally, at each stage, a movement operation can move one and only one mover.

5. It doesn't follow from anything that there has to be a single selector. For this reason it's possible for a single instance of a single operation to create multiple new sisterhood relations. Because it isn't necessary for the purposes of this chapter, I don't propose any constraint on the number of selectors an operation has, but I would like to note that any operation with multiple selectors will not be a member of the Extension Condition Family, and therefore it will not be in the Simultaneous Merge class. All such operations will still output a binary branching object.

Imagine a stage in a derivation where an operation f can apply, where f's selector is $A^{SO}_{\bullet F \bullet}$. If there are exactly two objects (e.g α^{SO}_F and β^{SO}_F) that can be f's selectee, then f can and needs to apply. The Single Selectee Constraint rules out an operation like Simultaneous Move in (21), where both α^{SO}_F and β^{SO}_F move, but how do we determine which of the two potential selectees moves to become sister to the attractor Γ^{SO}?

(21) *Simultaneous Move (SimMove)*



For empirical reasons, we want the operation to be able to apply to one and not be able to apply the other, when α^{SO}_F and β^{SO}_F are in a certain structural configuration; we don't want free variation between the two options. What we find is that α^{SO}_F is the selectee (i) when it irreflexively contains β^{SO}_F or (ii) when it asymmetrically c-commands β^{SO}_F; the opposite is true if the roles are reversed. Case (i) describes A-over-A effects, and case (ii) describes Relativized Minimality effects (Chomsky, 1973; Chomsky, 1981; Barss and Lasnik, 1986; Larson, 1988, 1990; Rizzi, 1990; Lasnik and Saito, 1993). I propose the Selectee Intervention Constraint (22) in order to account for A-over-A and Relativized Minimality effects.

(22) **Selectee Intervention Constraint**

Let the following hold:

a. S_i is a derivational stage;

b. $A^{SO}_{\bullet F \bullet}$ (for some $F \in$ SYN-F) is contained in some $\text{SO}_k \in S_i$;

c. $f(SO_k, \ldots)$ is defined over objects in some subset of S_i.

B^{SO}_F is **not** the selectee of $f(SO_k, \dots)$ at S_i if

(Activity Condition) there is some C^{SO}_F ($\neq B^{SO}_F$) such that for all $\text{SO}_{\in S_i}$, C^{SO}_F is not sister to $A^{SO}_{\bullet F \bullet}$ in SO, and

i. $A^{SO}_{\bullet F \bullet}$ irreflexively contains C^{SO}_F or

ii. $A^{SO}_{\bullet F \bullet}$ asymmetrically c-commands[6] C^{SO}_F; and

(A-over-A Condition) C^{SO}_F irreflexively contains B^{SO}_F, or

(Relativized Minimality Condition) C^{SO}_F asymmetrically c-commands B^{SO}_F.

The Selectee Intervention Constraint blocks a syntactic object B^{SO} from being a selectee under a certain conditions. These conditions always involve some other potential selectee C^{SO} that acts as a structural intervener. According to the Activity Condition, an object C^{SO} is a potential selectee if it bears a basic feature F that matches the operations trigger $\bullet$F$\bullet$, it is not already a sister to the selector $A^{SO}_{\bullet F \bullet}$, and is irreflexively contained in or asymmetrically c-commanded by the selector.[7] If an object meets the Activity Condition, it is a *potential intervener*. The A-over-A Condition says that B^{SO} is blocked from being a selectee if it is irreflexively contained in a potential intervener (that is irreflexively contained in the selector), and the Relativized Minimality Condition says that B^{SO} is blocked from being a selectee if it is asymmetrically c-commanded by a potential intervener (that is asymmetrically contained

6. We can derive the asymmetric c-command relation from the sisterhood relation:

(i) **Asymmetric C-command**
A syntactic object α^{SO} *asymmetrically c-commands* a syntactic object β^{SO} iff α^{SO} is in a sisterhood relation with a syntactic object γ^{SO} that irreflexively contains β^{SO}.

7. Karlos Arregi (p.c.) noticed that the Selectee Intervention Constraint doesn't rule out an operation that moves a selectee from inside a sister to the selector. He pointed to Richards's (2004) *Russian-doll questions* in Bulgarian. These questions involve multiple wh-movement, where one moved wh-phrase is contained in the other. Richards (2004) argues that the containing wh-phrase must move before the contained wh-phrase moves. This is exactly the order of operations predicted by the Selectee Intervention Constraint. However, the Selectee Intervention Constraint alone doesn't explain the full Russian-doll question paradigm. I believe the system proposed in this chapter can provide an account of all the Russian-doll questions if the contained wh-phrase sometimes undergoes scrambling out of the contained wh-phrase before any wh-movement occurs.

in the selector).[8] If there are two potential selectees C^{SO} and B^{SO} and C^{SO} blocks B^{SO} from being a selectee; then I will say that C^{SO} is an intervener for B^{SO}.

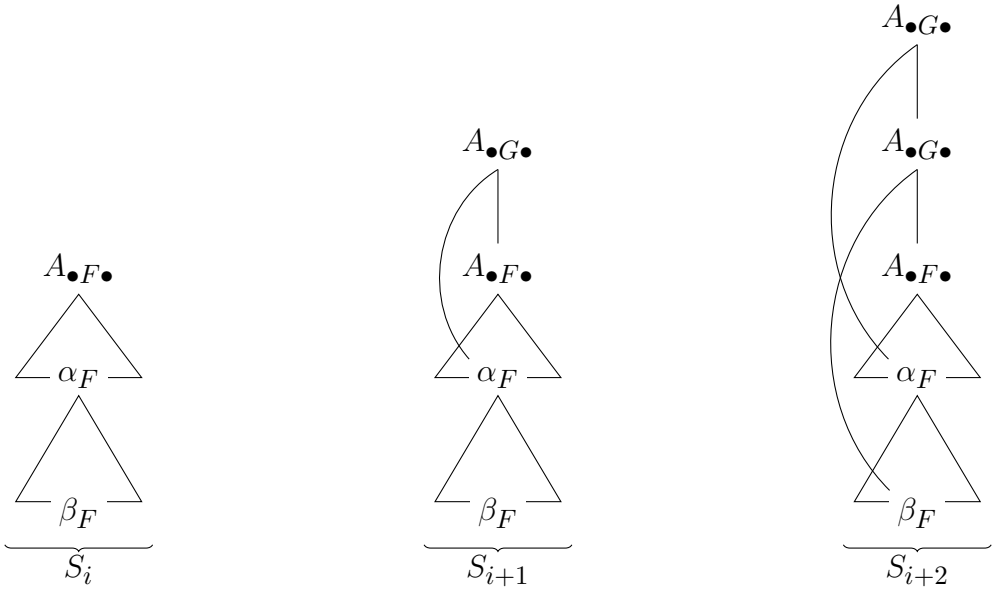
There are two properties of the Selectee Intervention Constraint that are worth discussing at this point. The first is related to the A-over-A Condition.[9] As stated in (22), the A-over-A Condition isn't just effective for movement operations; it also blocks certain selectees in non-movement operations. For example, if the derivational stage is $S = \{\mathrm{V}_{\bullet P \bullet}, [_{\mathrm{P}_P}\ \mathrm{P}_P\ \mathrm{DP}], \ldots\}$, where $\mathrm{V}_{\bullet P \bullet}$ is a verb that selects for a preposition (or a prepositional phrase) and [P P DP] (henceforth, PP^{SO}_P) is a prepositional phrase, then Binary Merge can apply to $\mathrm{V}_{\bullet P \bullet}$ and PP^{SO}_P. Interestingly there are two potential selectees (i.e. two objects that bear the basic feature P): the prepositional phrase itself PP^{SO}_P and object P^{SO}_P that is immediately contained in it (colloquially, the head of PP^{SO}_P). According to the A-over-A Condition, PP^{SO}_P is an intervener for P^{SO}_P.

I would also like to note the impermanence of selectee intervention. If a potential selectee is blocked by the Selectee Intervention Condition at one stage, it doesn't necessarily mean that it will be blocked from being a selectee of the same operation at a subsequent stage. Consider the derivation in (23).

8. According to the Selectee Intervention Constraint, a potential intervener is never a defective intervener in Chomsky's (2000) sense. Because defective intervention effects won't have any bearing on the arguments in this chapter, I've decided not to address the issue. With that said, I think that the system I'm proposing in this thesis can be easily adapted to account for defective intervention effects.

9. Van Riemsdijk and Williams (1988, §2.3) argue that the A-over-A Condition both over- and undergenerates, and therefore should be abandoned. Since then, some researchers, including Lasnik and Saito (1993, p.199, n.14), have presented arguments in favor of the A-over-A Condition. I don't find any of the arguments convincing when it comes to movement operations; however, the A-over-A Condition, or something like it, seems necessary to account for certain restrictions on non-movement operations. If necessary, one could redefine the A-over-A Condition so that it is only applicable to non-movement (i.e. binary) operations.

(23)



At stage S_i, an operation f can apply, where f's selector is $A^{SO}_{\bullet F\bullet}$ and there are exactly two objects (e.g α^{SO}_F and β^{SO}_F) that can be f's selectee. The potential selectee α^{SO}_F is an intervener for β^{SO}_F. As a result, β^{SO}_F is blocked from being f's selectee at S_i, and α^{SO}_F is the selectee. Therefore α^{SO}_F and $A^{SO}_{\bullet F\bullet}$ are sisters in the output of f. It follows that α^{SO}_F doesn't block β^{SO}_F from being f's selectee at stage S_{i+1}—α^{SO}_F doesn't meet the Activity Condition. So β^{SO} can be f's selectee at S_{i+1}, and if operations are obligatory, then f must apply with β^{SO}_F as its selectee. The output of the second instance of f would be S_{i+2} in (23).[10]

This state of affairs should raise some red flags, because it isn't always the case that an operation applies until every potential selectee has become sister to a selector, so some mechanism needs to be in place to prevent the overapplication of operations in this way. The mechanism I propose is the constraint, called the Trigger Activity Constraint, in (24).[11]

10. Informally, I'll assume that α^{SO}_F does not meet the Activity Condition at S_{i+2} (despite the fact that it's not sister to $A^{SO}_{\bullet F\bullet}$) because $A^{SO}_{\bullet F\bullet}$ doesn't irreflexively contain or asymmetrically c-command every position of α^{SO}_F. I'll discuss the notion of *position* in §3.2.3.

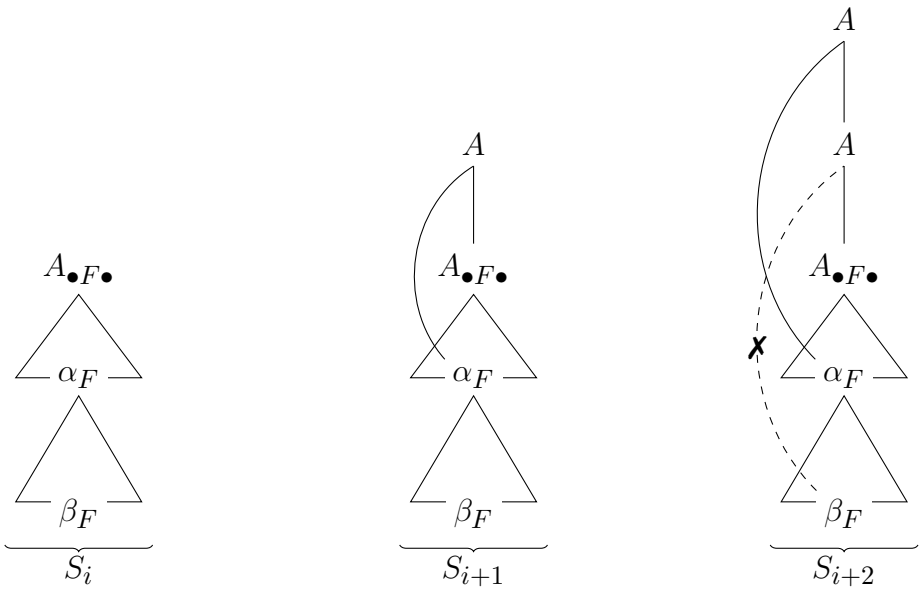
11. The Trigger Activity Constraint was inspired by the Edge Feature Condition in Müller 2010.

(24) **Trigger Activity Constraint**

Let A^{SO} bear an empty selectional list. For every syntactic object B^{SO} contained in A^{SO}, B^{SO} isn't a potential selector.

The Trigger Activity Constraint determines the derivational stage at which a trigger feature can no longer trigger an operation. Consider the derivation in (25).

(25)



At stage S_i, an operation f can apply, where f's selector is $A^{SO}_{\bullet F\bullet}$ and there are exactly two objects (e.g α^{SO}_F and β^{SO}_F) that can be f's selectee. The object α^{SO}_F blocks β^{SO}_F from being f's selectee at S_i, and f applies with α^{SO}_F as its selectee to derive S_{i+1}. The output of f, an element of S_{i+1}, is A^{SO}. A^{SO} contains (and is projected from) $A^{SO}_{\bullet F\bullet}$, and it bears an empty selectional list. The Trigger Activity Constraint therefore prevents f from applying to A^{SO} with $A^{SO}_{\bullet F\bullet}$ as its selector and β^{SO}_F as its selectee.

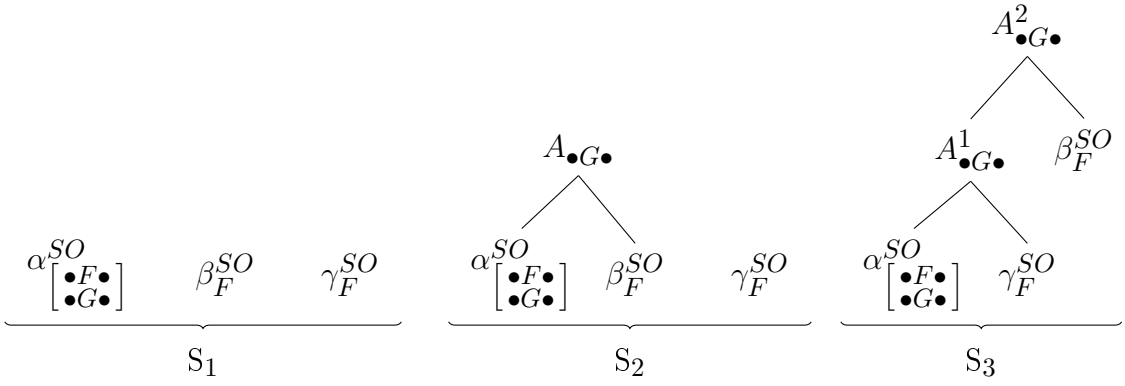
An interesting generalization follows from the Trigger Activity Condition: if a selectional feature •F• is the final term of a selectional list, then •F• can trigger at most one operation. If a selectional •G• feature is a non-final term on a selectional list, then •G• can trigger a potentially infinite number of operations. In §3.2.2, I'll show how this property can be

exploited to explain why some languages only permit one instance of (overt) wh-movement per clause (e.g. English) and why other languages require every wh-phrase to move overtly in each clause (e.g. Bulgarian).

3.1.4 Feature satisfaction

I've been operating under the assumption that a single selectional feature feature can trigger more than one operation. The following derivation is perfectly licit in a theory that permits at least some non-Extension Condition Family member(s) (e.g. Replacement Merge). This is true despite the fact that the selectional feature •F• on $\alpha^{SO}_{\bullet F \bullet}$ triggers both the operation that derives S_2 from S_1 and the operation that derives S_3 from S_2.

(26)



In a theory that only permits Extension Condition Family members (i.e. strictly extending theories), the derivation in (26) isn't licit because the operation that derives S_3 from S_2 isn't an Extension Condition Family member. It also follows from such a theory—that is, in conjunction with the Sisterhood Condition (5) and the Projection Condition in (9)—that a selectional feature will never trigger more than one operation. At S_2, no licit operation can create a sisterhood relation between $\alpha_{\bullet F \bullet}$ and another object, so none of α^{SO}'s selectional features are possible triggers; α^{SO} can't be any subsequent operation's selector. Furthermore, •F• (on α^{SO}) can't be the trigger for any subsequent operation because it previously triggered an operation. Given this set of facts, it appears that the selectional feature has

been *satisfied* (or *checked*), and it's clear that strictly extending theories predict that features are satisfied. Is this a virtue of strictly extending theories?

There is good reason to believe that some features should only be able to trigger one operation. For example, feature-driven approaches to c(ategory)-selection and l(exical)-selection must restrict features from triggering more than one operation. In (27a), we see that *hugs* selects for an object of category D. But as (27b) shows, *hugs*' selectional feature can only be said to trigger one operation; if *hugs* takes two argument DPs, the result is clearly unacceptable. The same pattern arises with *depends* in (28). The (a) example demonstrates that *depends* l-selects for an *on*-phrase (or a PP headed by *on*), but from the (b) example, we see that *depends* can't take two *on*-phrase arguments.

(27) When John's scared...

a. he hugs$_{\bullet D\bullet}$ pillows$_D$/Mary$_D$.

b. *he hugs$_{\bullet D\bullet}$ pillows$_D$ Mary$_D$. (int. 'John hugs pillows and Mary.')

(28) In times of crisis...

a. John depends$_{\bullet on\bullet}$ [$_{on}$ on Mary]/*[$_{for}$ for Mary].

b. *John depends$_{\bullet on\bullet}$ [$_{on}$ on Mary] [$_{on}$ on Paul]. (int. 'John depends on Mary and on Paul.')

On the basis of the above examples, we can conclude that at least some selectional features are satisfied when they trigger an operation, and satisfied features can't trigger operations. In fact, it appears that c-/l-selectional features have an additional property: They must be satisfied. From (29), we can conclude that the c-selectional feature •D• on *devours* must be satisfied (i.e. it must trigger an operation). The same is true of l-selectional features, as we can see in (30).

(29) When John's hungry...

a. he devours$_{\bullet D\bullet}$ his food.

b. *he devours$_{\bullet D\bullet}$.

(30) In times of crisis...

a. John depends$_{\bullet on\bullet}$ [$_{on}$ on Mary].

b. *John depends$_{\bullet on\bullet}$.

Although it appears that c- and l-selectional features must be satisfied, it could be that this is a special property of those features. There is evidence that some features can and (likely) must trigger multiple operations. To take just one example, the phenomenon of clitic climbing in Spanish has an all-or-nothing property. As noted in Aissen and Perlmutter 1976, it's either the case that every clausemate pronominal clitic is realized in an embedded clause, as in (31a), or they all raise into the embedding clause (31b). If only a proper subset of clitics raise into the embedding clause, the result is unacceptable, as we can see in (31c-d).

(31) *Clitic Climbing (Spanish)* (Aissen and Perlmutter, 1976, p.7)

a. Quiero mostrár-te-los.
want.1SG show.INF-2SG-3PL
'I want to show them to you'

b. Te los quiero mostrar.
2SG 3PL want.1SG show.INF
'I want to show them to you'

c. *Te quiero mostrar-los.

d. *Los quiero mostrar-te.

One could analyze the Spanish clitic climbing paradigm in (31) as involving a single occurrence of the feature •cc• (for clitic climbing) that obligatorily triggers the movement of a clitic, and because •cc• can't be satisfied, it triggers the same operation over and over again

until every potential selectee has moved, resulting in examples like (31b). If •cc• doesn't occur on the relevant head, then none of the clitics move, as in (31a). And because clitic climbing is obligatory when the conditions for its application are met, there is no situation where only one of two (or two of three) selectees undergo movement: If the condition for one instance of clitic climbing is met, then the conditions should be met for every clitic to climb. Therefore, the analysis correctly predicts the all-or-nothing property of clitic climbing.

Clitic climbing isn't the only operation that has this exhaustive application effect. As I will discuss in detail in §3.2, several languages that permit multiple wh-movement, in fact, require that all wh-phrases must undergo movement. And there is evidence that some of the languages that require exhaustive wh-movement (i.e. Bulgarian) involve a single feature that triggers every instance of wh-movement (Rudin, 1988; Richards, 1997, 1999, 2001).

All of this is to say that some selectional features need to be satiable (e.g. those involved in c- and l-selection) and some selectional features need to be insatiable (e.g. those involved in Spanish-like clitic climbing and Bulgarian-like multiple wh-movement). I won't add anything formally to the framework to deal with these differences, because any formal solution will need to be quite complex, given the generally sticky set of facts. Also, it has very little bearing on any arguments to come. That said, I'll make it clear in any of the following examples when a feature needs to be satisfied and when a feature can only trigger one operation. I have a strong suspicion that there is a negative correlation between the need to be satisfied and the ability trigger more than once. And more, it seems possible to me that satiable features are those that trigger binary operation (e.g External Merge), and insatiable features are those that trigger unary operations (e.g. Internal Merge).

3.2 Multiple wh-movement in Bulgarian

In this section I will present a non-strictly extending analysis of multiple wh-movement in Bulgarian[12]. The analysis is strongly influenced by Richards (1997, 1999, 2001)'s analysis of the same phenomenon. Multiple wh-movement in Bulgarian is order-preserving, which means that the movement dependencies are crossing in a sense that I'll make clear soon. Crossing dependencies (and therefore Bulgarian multiple wh-movement) pose a problem for strictly extending theories of syntax. Nonetheless, an account of the phenomenon is possible in a strictly extending theory by weakening constraints on the application of syntactic operations (namely, the Relativized Minimality Condition). It's interesting that the two classes of analyses of Bulgarian wh-movement (and crossing dependencies, in general) make different predictions about the derivations of crossing dependency constructions, and yet I haven't been able to find any empirical evidence that can lead us to prefer one kind of analysis over the other.

I argued that strictly extending syntactic theories have strongly equivalent declarative alternatives in Chapter 2.[13] So it must be the case that there is a declarative analysis of crossing dependencies, although I have yet to discover a clearly declarative analysis of Bulgarian multiple wh-movement.[14] And given that I haven't discovered a way to distinguish

12. In reality, I provide an analysis of a toy version of multiple wh-movement. My analysis will account for examples where there are exactly two animate wh-phrases. The full range of facts are complicated by a number of factors.

13. You may wonder, given the additions I've made to the framework in this chapter, if we need to reevaluate the claim that strictly extending theories always have strongly equivalent declarative alternatives. The additions made in this chapter restricted the set of possible final representations in a number of ways, but it isn't the case that the additions have fundamentally changed the way we classify operations into Extension Condition and Non-Extension Condition families. Because the keystone of the argument to the equivalency claim was the relation between final and intermediate representations (i.e. determinability of intermediate representations from the final representation) and because this relation hasn't been fundamentally changed by any of the conditions and constraints on applicability of operations, I don't believe that the additions have any bearing on the arguments in Chapter 2.

14. I could be convinced that Rudin (2013)'s analysis of word order in Bulgarian is a declarative analysis. She argues that word order, especially wh-word order, is governed by a set of speaker preferences, and what's more she argues that wh-word order can't be explained by principles governing the order of syntactic operations. While she does posit a syntactic wh-movement operation, it doesn't seem to play any role in

empirically between strictly and non-strictly extending analyses of Bulgarian multiple wh-movement, it must be the case that I haven't found a way to argue for a non-strictly extending analysis over a declarative analysis.

3.2.1 Word order in Bulgarian declaratives and wh-interrogatives

Word order in Bulgarian isn't fixed, but the consensus is that the basic surface word order is subject-verb-object (SVO). In (32), we see three (of six) possible orders,[15] which is enough to make the argument that the default word order is SVO.

(32) *Bulgarian Word Order* (Rudin, 2013, p.17, ex.2)

a. Knigata vidja Georgi.
book-the saw$_{3SG}$ Georgi
'Georgi saw the books.'

b. Decata vidja Georgi.
children-the saw$_{3SG}$ Georgi
'Georgi saw the children.'

c. Marija vidja Georgi.
Marija vidja$_{3SG}$ Georgi
'Marija saw Georgi.'

Examples (32a)-(32b) demonstrate that Bulgarian permits Object-Verb-Subject (OVS) word order. The preverbal arguments *knigata* ('the book') and *decata* ('the children') can't be interpreted as the logical subject of the verb *vidja* ('saw'). *Knigata* is inanimate, and since books (according to popular wisdom) can't see, the postverbal argument *Georgi*, which denotes a person, is understood as the logical subject. As for (32b), the preverbal argument is plural but the verb agrees with a singular argument. Bulgarian displays subject-verb agreement, so the sentence is interpreted with *Georgi* as the logical subject.

---

explaining the core desiderata of Richards's studies.

15. (Rudin, 2013, p.17, ex.1) shows that all six permutations of subject, object, and verb are possible in Bulgarian.

The preverbal argument *Marija* in (32c) is the name of a person (presumably). Interpreting *Marija* as the logical subject doesn't conflict with world knowledge, as with *knigata*, and it doesn't violate a grammatical requirement, as with *decata*. Therefore, it is possible, and in fact, strongly preferred for *Marija* to be interpreted as the logical subject. Neither common sense nor agreement conflicts prevent the postverbal argument *Georgi* from being interpreted as the logical subject, but interpreting *Georgi* as the logical subject is only possible 'given appropriate intonation and context' (Rudin, 2013, p.18, fn.3).

It is important that we've established the default word order because Bulgarian multiple wh-questions obligatorily involve multiple overt wh-movement, which is said to be 'order-preserving' (Rudin, 1988, 2013; Richards, 1997, 1999, 2001; Nissenbaum, 2000). For example, consider the examples in (33), in which there are two wh-phrases that undergo wh-movement.

(33) *Multiple Wh-movement in Bulgarian* (Rudin, 1988, pp.472-73, ex.54)

a. Koj kogo t_k vižda t_j?
who whom sees
'Who sees whom?'

b. * Kogo koj t_k vižda t_j?
whom who sees
lit. 'Whom sees who?'

In (33a), the subject wh-phrase precedes the object wh-phrase. This, of course, matches the relative precedence relations between subjects and objects in the basic declarative clause. If the order of the two wh-phrases is reversed, as in (33b), the result is unacceptable. It is in this sense that wh-movement is order-preserving.

Essentially, the following generalization holds for order-preserving movement dependencies in Bulgarian (and maybe all crossing movement dependencies). The generalization can be attributed to Rudin (1988) and Richards (1997), so I've named it the Rudin-Richards Generalization.

(34) **Rudin-Richards Generalization**

Two movement operations are order-preserving iff both operations are triggered by the same selectional feature on the same selector.

The Rudin-Richards Generalization says that order-preserving movement arises when multiple movers target specifiers of the same head. In other words, crossing dependencies always involve two or more movers to the same attractor.

In Bulgarian, the Rudin-Richards Generalization holds for examples where there are exactly two animate wh-phrases. However, it doesn't always hold for multiple wh-questions with more than two wh-phrases. If, for example, there is a subject (S/NOM), direct object (DO/ACC), and indirect object (IO/DAT) wh-phrase in a single clause, then there are two possible orders: (1) S-DO-IO or (2) S-IO-DO. Any example in which the subject doesn't linearly precede the rest of the arguments is unacceptable.[16]

(35) *Triple wh-questions (Bulgarian)* (Rudin, 2013, p.124, ex.74)

a. Koj kakvo na kogo e kazal?
who.NOM what.ACC to who.DAT AUX said
'Who said what to who?'

b. Koj na kogo kakvo e kazal?
who.NOM to who.DAT what.ACC AUX said

The free order of the direct and indirect object in triple wh-questions isn't due to the fact that they have no basic relative order (linearly or structurally). If that were the case, we

16. Rudin (1988; 1990; 2013) doesn't provide ill-formed examples where the subject/nominative wh-phrase doesn't precede the other two wh-phrases in a triple wh-question. But her discussion around examples (35) imply that a subject/nominative wh-phrase be initial. And more, Izvorski 1995 provides the following pair of examples, which show that a triple wh-question is acceptable when the subject/nominative wh-phrase is linearly first (among the wh-phrases) and unacceptable when it is linearly final.

(i) a. Koj na kogo kakvo beše kazal?
who.NOM to who.DAT what.ACC AUX.PAST said
'Who had said what to whom'

b. *Kakvo na kogo kojo beše kazal? (Izvorski, 1995, p.66,ex.37)

I haven't been able to find a full paradigm of triple wh-questions in Bulgarian involving a nominative, an accusative, and a dative argument.

might expect them to have a free order when there is no subject wh-phrase. However, double wh-questions with a direct and an indirect object wh-phrase must order the direct object before the indirect object, as you can see in (36).

(36) *Accusative and dative wh-questions (Bulgarian)* (Rudin, 2013, p.124,ex73)

a. Kogo na kogo e pokazal Ivan?
Who.ACC to who.DAT AUX pointed-out Ivan
'Who did Ivan point out to whom?'

b. *Na kogo kogo e pokazal Ivan?

In the acceptable example (36a), the direct object wh-phrase *kogo* precedes the indirect object wh-phrase *na kogo*. When the order is reversed in (36b), the resulting sentence is unacceptable.

Rudin 2013, §4.7.1 notes a number of other ordering 'rules-of-thumb' like inanimate wh-phrases, like *kakvo* ('what'), tend to follow human referring ones, except this tendency is weakened when the inanimate wh-phrase is D-linked. Also, adverbial wh-phrases tend to follow everything. But Rudin calls them rules-of-thumb because these tendencies are often violated. All of this is to say that the Bulgarian wh-questions are complex and often fuzzy. Any analysis that can account for the full range of facts needs to include sophisticated mechanisms that take time to develop, introduce, and justify. Because I'm concerned, in this thesis, with the kinds of theories that provide the means for syntactic analysis and not the fine-grained particulars of the analyses, I will focus on a toy version of multiple wh-questions—that is, the set of examples that involve exactly two animate wh-phrases, one subject and one object. For this set of examples, the Rudin-Richards generalization holds.

3.2.2 An account of multiple wh-movement

In order to understand my analysis of multiple wh-movement in Bulgarian, I need to establish how my framework accounts for languages without multiple wh-movement in multiple wh-questions. For example, English permits multiple wh-questions (i.e. constructions with multiple wh-phrases), but English only permits one wh-phrase to move within or outside of a clause. If the multiple wh-phrases move, the result is clearly unacceptable (37c).

(37) **Multiple wh-questions (English)**

a. Who do you think bought what?

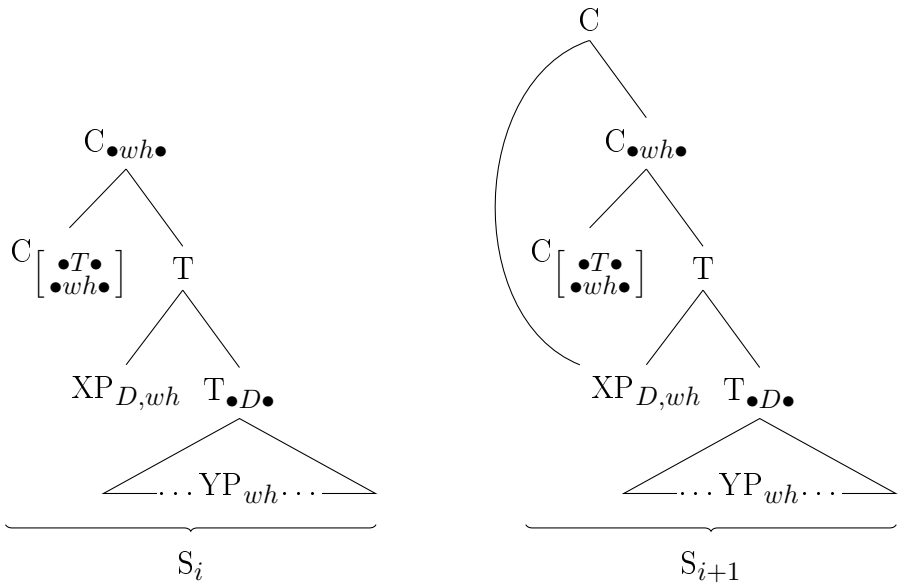
b. *What do you think who bought?

c. *Who what do you think bought?

Furthermore the movement operation that does occur must obey the Relativized Minimality Condition (22). In a simple transitive clause, the subject wh-phrase moves and the object wh-phrase remains in place, as in (37a). If the opposite occurs, the result is unacceptable (37b).

It's clear that the asymmetry between subjects and objects (and other wh-extraction asymmetries) are handled by the Relativized Minimality Condition. But what accounts for the lack of multiple wh-movement in English? I propose that multiple wh-movement in English is blocked by the Trigger Activity Constraint (24). Consider the derivation of a multiple wh-question with a single mover in (38).

(38) *Single wh-movement*



There are two crucial properties of the syntactic object at stage S_i. For one, the two wh-phrases are in an asymmetric c-command relation—$\text{XP}_{D,wh}$ asymmetrically c-commands YP_{wh}. Because they both bear the *Basic* feature wh, an attractor bearing a trigger •wh• can't, according to the Relativized Minimality Condition, attract YP_{wh} until after it attracts $\text{XP}_{D,wh}$; in other words, $\text{XP}_{D,wh}$ must move first in this derivation.

The second crucial property of this object at S_i is the complementizer C that bears the selectional list $\langle \bullet T \bullet, \bullet wh \bullet \rangle$. The first term is a c-selectional feature that had previously triggered an instance of Binary Merge between C^{SO} and T^{SO}, and the second, is a feature that triggers wh-movement. The operation that derives S_{i+1} from S_i is an instance of wh-movement, where $\text{XP}_{D,wh}$ is the mover. Because of the nature of feature projection, as defined in (9), the output of the operation is a syntactic object bearing an empty selectional list. Due to the Trigger Activity Constraint, which I've repeated below, every trigger feature in the containment domain of the object is inactive—that is, they can no longer trigger any subsequent operation in the derivation.

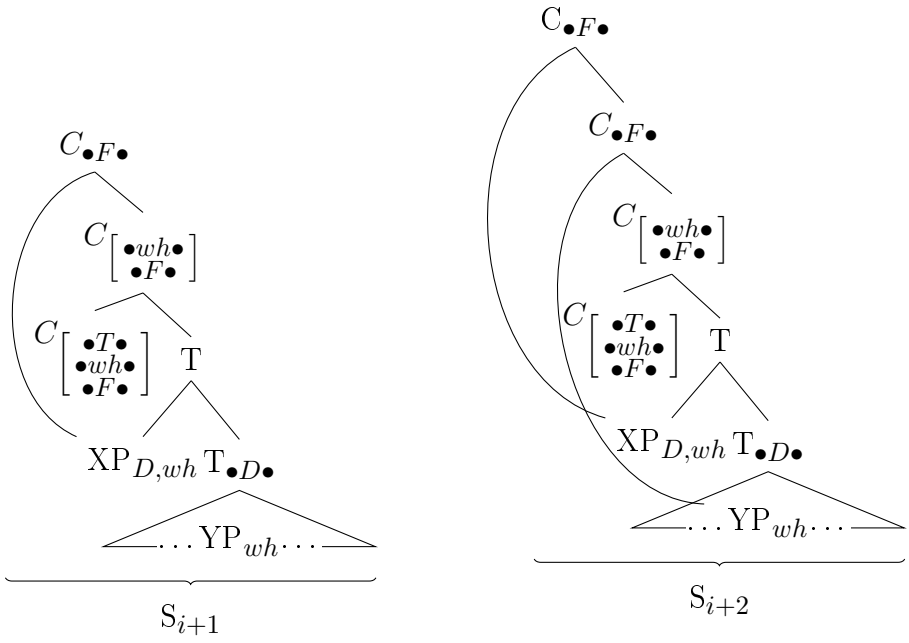
(39) **Trigger Activity Constraint** (repeated)

Let A^{SO} bear an empty selectional list. For every syntactic object B^{SO} contained in A^{SO}, B^{SO} isn't a potential selector.

As far as the derivation of multiple wh-questions in English is concerned, the Trigger Activity Constraint prevents the structurally lower wh-phrase XP_{wh} from moving because there is no derivational stage at which it can licitly move. It's necessary for the higher argument $\text{YP}_{D,wh}$ to move first in order to comply with the Relativized Minimality Condition, but once $\text{YP}_{D,wh}$ moves, the feature that would trigger movement of XP_{wh} is no longer active.

I propose that the difference between Bulgarian and English with respect to multiple wh-questions is a difference in the lexical inventories of the languages. I posited above that English has a complementizer with a two term selectional list $\ell^{Eng} = \langle \bullet T \bullet, \bullet wh \bullet \rangle$. Bulgarian, in contrast to English, has a complementizer with a three term selectional list $\ell^{Bul} = \langle \bullet T \bullet, \bullet wh \bullet, \bullet F \bullet \rangle$. The final selectional feature •F• creates a derivational window for multiple instances of wh-movement to occur. Consider the partial derivation of a multiple wh-movement construction in (40).

(40) *Double wh-movement*



Stage S_{i+1} was derived from the previous stage by wh-movement of $\mathrm{XP}^{SO}_{D,wh}$ to the attractor $C_{\left[\begin{smallmatrix}\bullet wh\bullet\\ \bullet F\bullet\end{smallmatrix}\right]}$. The output $C^{SO}_{\bullet F\bullet}$ bears a non-empty selectional list, and therefore the Trigger Activity Constraint doesn't block •wh• on $C_{\left[\begin{smallmatrix}\bullet wh\bullet\\ \bullet F\bullet\end{smallmatrix}\right]}$ from triggering another instance of wh-movement. In fact, the conditions are met for YP^{SO}_{wh} to undergo wh-movement because $\mathrm{XP}^{SO}_{D,wh}$ no longer intervenes. So this operation derives the next stage S_{i+2}.

Essentially, the difference between languages that permit multiple wh-movement and those that permit only one instance of wh-movement is the selectional feature •F•. Is there any independent evidence for •F•? The short answer is probably not. It could be that •F• triggers overt topicalization, in which case we could rename it •Top•. Example (41a) shows that topicalization and wh-movement can co-occur in a single clause in Bulgarian.

(41) *co-occurrence of topic- and wh-movement* (Rudin, 2013, p.99, ex.24)

a. Ivan na kogo dade knigite?
Ivan to who.DAT gave books.the
'As for Ivan_i, who did he_i give the books to?'

b. *Na kogo Ivan dade knigite?

The sentence initial nominal *Ivan* is interpreted as a topic, and the linearly second constituent *na kogo*, a wh-phrase, has presumably undergone wh-movement to a position below the topic. The sentence is unacceptable when these elements are reversed, as in (41b), so one potential analysis is that wh-movement derivationally precedes topicalization (movement) due to the order of features on the lexical complementizer. That is, the Bulgarian lexicon has a $C_{\left[\begin{smallmatrix}\bullet T\bullet \\ \bullet wh\bullet \\ \bullet F\bullet\end{smallmatrix}\right]}$. You can find additional examples of topicalization and wh-movement co-occurring in Bulgarian in Rudin 1990.

There are two potential arguments against the claim that •F• could be analyzed as •Top•. One of the predictions of the analysis is that topicalization should be able to block multiple wh-movement in Bulgarian. It could be, for instance, that •wh• triggers an instance of wh-movement, and then before another instance of wh-movement occurs, •Top• triggers an instance of topicalization. At that point, the Trigger Activity Constraint (39) would block $C_{\left[\begin{smallmatrix}\bullet wh\bullet \\ \bullet F\bullet\end{smallmatrix}\right]}$ from triggering another instance of wh-movement; so if there is an un-moved wh-phrase in the clause, we predict that it will in remain in place. I haven't been able to find any example of topicalization occurring in a multiple-wh clause in Bulgarian, so I can't confirm if the prediction is borne out. But if topicalization is possible in multiple wh-questions in Bulgarian, it would be interesting to know (i) if wh-phrases can remain *in situ* and (ii) what the word order facts are.

Since the •Top• is the last term in a selectional feature list, as soon as it triggers an operation, the object that bears it (i.e. $C^{SO}_{\bullet Top\bullet}$) is no longer a possible trigger due to the Trigger Activity Condition (39). So another prediction of the •Top• analysis is that multiple topicalization should be impossible. Rudin 1990 reports that multiple topicalization is acceptable in Bulgarian (42).

(42) Ot bazata jadene dali šte ni donesat?
from base-the food whether will to-us bring.3PL
'(I wonder) will they bring us food from the base?' (Rudin, 1990, p.434, ex.12a)

According to Rudin (1990), *Ot bazata* ('from the base') and *jadene* ('food') are interpreted as topics because they precede the complementizer *dali* ('whether').

It's fair to conclude that the purported •F• feature on the complementizer of multiple wh-movement languages, like Bulgarian, should not be analyzed as the feature that triggers topicalization. It may just be the case that the only function of •F• is to allow for cases of multiple movement. It isn't unprecedented to posit a feature that serves this kind of function: Edge features serve a similar function in Phase Theory (Chomsky, 2000, 2001, 2008; Müller, 2010).[17]

3.2.3 The problem of crossing dependencies

The order-preserving property of multiple wh-movement in Bulgarian, like several other phenomena that appear to involve crossing (movement) dependencies (e.g. object shift in several natural languages), presents a difficult puzzle for any strictly extending syntactic framework. This is why Richards (1997, 1999, 2001) proposed a non-strictly extending analysis of crossing dependency phenomena. In this section, I'll show you what the problem is, and I'll point out how non-strictly extending theories make different predictions about crossing dependencies than strictly extending theories (and by extension, declarative theories). Despite the fact that the two classes of theories make different predictions, I have yet to find any empirical evidence in support of one over the other.

The problem of crossing dependencies arises for strictly extending theories of syntax because of a tension between strict extension and locality conditions on dependencies. Before I can demonstrate the problem crossing dependencies pose for strictly extending theories, I need to define some terms, starting with *movement dependency*.

17. The term *edge feature* isn't adopted until Chomsky 2008, but I cite Chomsky's earlier work here because the same concept is used under the label EPP (Extended Projection Principle) feature.

(43) **Movement dependency**

A *movement dependency* is a pair $dep = \langle SO_1, SO_2 \rangle$, such that

a. $SO_1 = SO_2$ and

b. SO_1 c-commands SO_2.

When a syntactic object SO moves due to a syntactic movement operation, then SO occupies one more position than it did before the operation. We can understand movement dependencies as a list of positions that a single syntactic object is in as a result of a single movement. For example, in (44), the syntactic object α^{SO} has undergone syntactic movement to become sister to Γ^{SO}, and as a result a movement dependency has been created $dep^{\alpha} = \langle \alpha_1^{SO}, \alpha_2^{SO} \rangle$.

(44) $[\alpha_1^{SO} \; [_{\Gamma} \; \ldots \alpha_2^{SO} \; \ldots]]$

The first term of the dependency (α_1^{SO}) is the occurrence of α^{SO} that is sister to Γ^{SO}, and the second term (α_2^{SO}) is the occurrence of α^{SO} that is contained in Γ^{SO}. Sometimes it's the case that a single syntactic object will move more than once. In that case, we can assume that some mechanism can chain movement dependencies together into an n-tuple.[18]

What I mean when I say that two dependencies dep^1 and dep^2 are *crossing* is that some term of dep^1 is in between the two terms of dep^2 and some term of dep^2 is in between the two terms of dep^1. In this case, I'm defining *in between* in terms of asymmetric c-command:

(45) **In between**

A syntactic object x is *in between* syntactic objects y and objects z iff (i) y asymmetrically c-commands x and (ii) x asymmetrically c-commands z.

18. There's also a possibility that a syntactic object is the selectee in an instance of Parallel Merge and then undergoes (non-)extending movement. For discussion of the interaction between Parallel Merge and movement dependency chains, I refer you to Nunes (2001). Parallel Merge won't arise in any of the case studies I cover in this chapter, so it will suffice to stick to the basic assumption in the main text.

I've schematized crossing movement dependencies in (46). The two syntactic objects that have undergone movement are α^{SO} and β^{SO}, resulting in two movement dependencies: $dep^{\alpha} = \langle \alpha_F^{SO^1}, \alpha_F^{SO^2} \rangle$ and $dep^{\beta} = \langle \beta_F^{SO^1}, \beta_F^{SO^2} \rangle$. The schematic is in line with the Rudin-Richards Generalization (34) since both movement of β_F^{SO} and α_F^{SO} is triggered by the selectional feature •F• on Γ^{SO}.

(46) *Crossing Dependency*

$$[_{\Gamma^2}\ \beta_F^{SO^1}\ [_{\Gamma^1}\ \alpha_F^{SO^1}\ [_{\Gamma_{\bullet F\bullet}}\ \ldots [_{\mathrm{B}}\ \beta_F^{SO^2}\ \ldots [_{\mathrm{A}}\ \alpha_F^{SO^2}]]]]]$$

That dep^{α} and dep^{β} are crossing is visually clear from the schematic because the movement arrows cross. More importantly, we know that they are crossing dependencies because one term of dep^{α} (namely, $\alpha_F^{SO^1}$) is in between the two terms of dep^{β}, and one term of dep^{β} (namely, $\beta_F^{SO^2}$) is in between the two terms of dep^{α}.

Now for the reason why crossing (movement) dependencies are problematic for strictly extending theories of syntax. Given that we're making the commonly held assumption that only one operation can occur per derivational stage, there are two possible orderings of the movement operations in (46): either β^{SO} moves first or α^{SO} moves first. If β^{SO} moves first, then the operation that moves α^{SO} must be non-extending move, which is incompatible with a strictly extending theory. However, according to the Relativized Minimality Condition (22), because β^{SO} is an intervener to α^{SO}, movement of α^{SO} is blocked. If a strictly extending theory permitted non-extending move, it wouldn't be strictly extending. Therefore the only way to analyze crossing movement dependencies within a strictly extending framework is to abandon or weaken the Relativized Minimality Condition—that is, at least under certain circumstances. For example, one might propose that certain selectional features attract the structurally lowest potential selectee.

Even with strong and unconditionally inviolable constraints on movement, non-strictly extending theories don't have any issues accounting for crossing movement dependencies. If

β^{SO} moves first, it no longer intervenes between α^{SO} and the attractor $\Gamma^{SO}_{\bullet F\bullet}$, so α^{SO} can undergo non-extending move. The other order of operations is ruled out by the Relativized Minimality Condition. Movement of α^{SO} is blocked before movement of β^{SO} because β^{SO} intervenes between α^{SO} and the attractor $\Gamma^{SO}_{\bullet F\bullet}$.

What I've just argued is that both strictly extending and non-strictly extending theories can account for crossing dependencies. Non-strictly extending theories can account for crossing dependencies because they (by definition) have weaker restrictions on the set of possible syntactic structure-building operations, while strictly extending theories can account for crossing dependencies by weakening constraints on the application of operations (namely, the Relativized Minimality Condition). Furthermore, I've shown that the two kinds of theories are forced into different analyses of the derivation of crossing dependencies. Strictly extending theories must take it that the structurally lower object α^{SO} moves first, while the higher argument (β^{SO}) must move first in non-strictly extending theories. This all means that the strictly and non-strictly extending theories make different predictions about intermediate representations of the derivations to crossing dependency constructions. As for the strictly extending analysis, (47a) is an intermediate representation. The non-strictly extending posits (47b) as an intermediate representation.

(47) a. *Move α^{SO} first*

$[_{\Gamma^1_{\bullet G\bullet}}\ \alpha_F\ [_{\Gamma_{\bullet F\bullet}}\ \ldots [_{\text{B}}\ \beta_F\ \ldots [_{\text{A}}\ \alpha_F]]]]$

b. *Move β^{SO} first*

$[_{\Gamma^1_{\bullet G\bullet}}\ \beta_F\ [_{\Gamma_{\bullet F\bullet}}\ \ldots [_{\text{B}}\ \beta_F\ \ldots [_{\text{A}}\ \alpha_F]]]]$

In theory, it's possible that there's empirical evidence for one or the other intermediate representations. If the output of a movement operation projects selectional features to the output, as is shown in (47a-b) because $\Gamma^1_{\bullet G\bullet}$ bears a selectional feature •G•, then it should

be possible for some other operation, triggered by •G• on $\Gamma^1_{\bullet G\bullet}$, to apply. And more, if such an operation applies before the second movement operation triggered by •F•, then the second operation could be blocked by the Trigger Activity Condition.

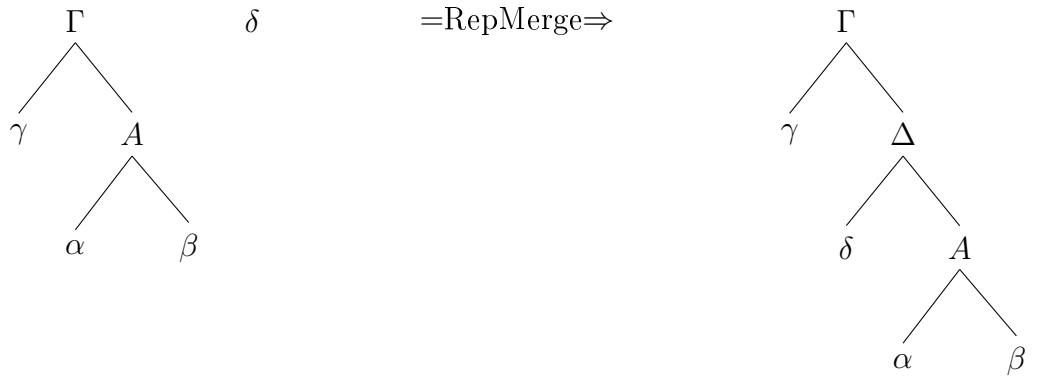
In less technical terms, the question is the following: Is it ever the case that a crossing dependency configuration is barred because another process blocks the creation of the derivationally second dependency, but not the first? As far as I know, no case exists, so I have to conclude, for now, that there is no empirical support for one kind of theory over the other. What's more, because every strictly extending theory has a strongly equivalent declarative alternative, there is no empirical basis for preferring any kind of derivational analysis over an adequate declarative analysis.

3.3 Late adjunction

In the previous section, I considered the problem that crossing dependency configurations pose for strictly extending theories of syntax. In short, the problem was that crossing dependencies force a staunch strictly extending analyst to weaken or qualify their locality conditions. While a non-strictly extending framework, like the one I adopted in (3.2.2), might have some conceptual advantages, there isn't, as far as I know, any empirical basis for preferring a non-strictly extending analysis to an adequate strictly extending analysis (or declarative analysis).

In this section, I explore some phenomena that have been analyzed using late adjunction. Late adjunction is a non-extending binary operation—that is, it's Binary Replacement Merge. If extending move is the unary analog of Binary Merge, then non-extending move is the unary analog of late adjunction. I'll assume that late adjunction is just any version of Binary Replacement Merge that isn't conditioned by selectional feature requirements. In fact, we may even want to go as far as to say that late adjunction is only permitted to apply to syntactic objects bearing empty selectional lists.

(48) **Replacement Merge (repeated)**



In the remaining sections, I'll present an empirical phenomenon that has been analyzed using late adjunction. In particular, we'll look at anti-reconstruction effects ('Lebeaux effects') that were first observed by Van Riemsdijk and Williams (1971, §III(6)). Then I'll show how anti-reconstruction effects present a problem for strictly extending theories.

3.3.1 Anti-reconstruction effects

Lebeaux (1991) presents a solution to the puzzle of anti-reconstruction. The puzzle of anti-reconstruction can be summarized as follows: why is it the case that some movement dependencies (specifically, $\bar{A}$-dependencies) don't reconstruct for the purposes of Condition C, while others do? To understand the puzzle, we need to establish (i) what Condition C is and (ii) what reconstruction is.

Condition C is a structural condition on co-referring syntactic objects (i.e. a binding condition) that requires an R-expression (for example, a name) to be un-c-commanded by a co-referring element. Consider the following example:

(49) $\text{He}_{i/*j}$ likes John_j.

If the subject pronoun *he* refers to someone other than the *John*—the one who is liked—then (49) is a perfectly acceptable sentence of English. However, if *he* and *John* refer to the same

person, the sentence is totally unacceptable. In the latter case, we say that the sentence violates Condition C or that the sentence displays a Condition C effect.

A movement dependency has reconstructed if the syntactic object that moved is interpreted in the lower position of the dependency (see (43) for a definition of dependency). Lebeaux (1991) noted that *picture* nominals always reconstruct for Condition C (50b).

(50) *Reconstruction for Condition C*

a. $\text{He}_{i/*j}$ likes those pictures of John_j.

b. ?*Which picture of John_j does he_j like t?

In (50a), where there is no movement dependency, we see that the familiar Condition C effect arises when the subject pronoun and the name *John* are co-referent. And it's clear that this Condition C effect isn't ameliorated by moving the wh-phrase *which picture of John* in (50). It's as if the wh-phrase is being interpreted in the structurally lower position of the movement dependency. In other words, the wh-phrase reconstructs for the purposes of Condition C.

Interestingly, Condition C effects can be ameliorated if the R-expression (e.g. *John*) is in a relative clause modifier, as can be seen in (51b). The lack of a Condition C violation is known as an anti-reconstruction effect.

(51) *Ant-reconstruction for Condition C*

a. *He_i likes those pictures that John_i took.

b. Which pictures that John_i took does he_i like?

As Lebeaux 1991 makes clear, it can't simply be that the complement of the preposition in *picture* nominals is visible for Condition C while relative clauses are not, because both are visible to Condition C if no movement occurs (50a) and (51a).

Lebeaux (1991) argues that the crucial difference between the two cases has to do with the argument/adjunct distinction. The PP's in *picture of* nominals are arguments, while

relative clauses are adjuncts. The generalization is supported by several paradigms, which I've included in (52)-(55).

(52) *Complements of* claim *vs. relative clauses* (Lebeaux, 1991, p.320, ex.3)

a. *He$_i$ denied the claim that John$_i$ made.

b. *He$_i$ denied the claim that John$_i$ likes Mary.

c. *Which claim that John$_i$ likes Mary did he$_i$ deny t?

d. Which claim that John$_i$ made did he$_i$ later deny t?

(53) *PP argument vs. PP adjunct* (Lebeaux, 1991, p.320, ex.4-5)

a. *He$_i$ looked at those pictures of John$_i$.

b. *He$_i$ looked at those pictures near John$_i$.

c. *Which pictures of John$_i$ did he look at t?

d. Which pictures near John$_i$ did he$_i$ look at t?

(54) *Process vs. result nominals* [inspired by (Lebeaux, 1991, p.320, ex.5)]

a. *He$_i$ is worried about the investigation of John$_i$'s office.

b. *He$_i$ is worried about the investigation near John$_i$'s office.

c. *Which investigation of John$_i$'s office is he$_i$ worried about?

d. Which investigation near John$_i$'s office is he$_i$ worried about?

(55) *Possessor vs. non-possessor PP* (Lebeaux, 1991, p.320, ex.6)

a. *He$_i$ likes those pictures of John$_i$.

b. *He$_i$ likes those pictures of John$_i$'s.

c. *Which pictures of John$_i$ does he$_i$ like?

d. Which pictures of John$_i$'s does he$_i$ like?

In each of the above paradigms, the arguments in the (c) examples reconstruct, and the adjuncts in the (d) examples don't reconstruct, for Condition C.

3.3.2 Lebeaux's non-strictly extending analysis

Lebeaux (1991)'s analysis is that the operations that compose arguments with their selectors must occur as early in a derivation as is possible. By contrast adjuncts can be inserted into the structure late—that is, after movement has occurred. When adjuncts are inserted late, the operation that inserts them is necessarily non-extending (i.e. binary Replacement Merge).

Let's walk through a derivation where reconstruction does, in fact, feed Condition C effects. These are the derivations that involve wh-movement of *picture* nominals. However, the derivations of all the (c) examples in (52)-(55) would be similar.

(56) *Reconstruction Derivation*

S_i: $[_{N \bullet of \bullet}$ picture $[_{\text{of}}$ of John$_j]]$

S_{i+n}: $[_C$ $\boxed{[_{wh}$ which $[_{N_{\bullet of \bullet}}$ picture $[_{\text{of}}$ of John$_j]]]}$ ←

$[_{C_{\bullet wh \bullet}} \ldots [_T$ he$_j \ldots [\ldots$ $\boxed{[_{wh} which [_{N_{\bullet of \bullet}}$ picture $[_{\text{of}}$ of John$_j]]]}$ $]]]]$

To derive stage S_i of (56), the (l-)selectional feature •of• on $picture_N$, $\bullet of \bullet$ triggers an instance of Binary Merge between *picture* and *of John*; the output is a N(P) *picture of John* that is contained in a wh-phrase at some later point in the derivation. At an even later stage in the derivation (S_{i+n}), the wh-phrase containing *picture of John* undergoes wh-movement across a pronoun *he* that is co-referent with *John* (in the picture nominal). However, moving the wh-phrase doesn't create a representations that meets Condition C: There is still an occurrence of *John* that is c-commanded by a co-referring pronoun, and therefore, Condition C is violated.

Now for a derivation of a sentence with an anti-reconstruction effect (57). This is a derivation of a sentence like (51b).

(57) *Anti-reconstruction Derivation*

S_i: $[_{wh}$ which picture$]$

S_{i+n}: $[_C$ $\boxed{[_{wh}\text{ which picture}]}$ $[_{C_{\bullet wh\bullet}} \ldots [_T$ he$_j \ldots [\ldots \boxed{[_{wh}\text{ which picture}]}]]]]$

S_{i+n+1}: $[_C$ $\boxed{[_{wh}\text{ which picture}}$ $[_{RC}$ that John$_j$ took$]]$

$[_{C_{\bullet wh\bullet}} \ldots [_T$ he$_j \ldots [\ldots \boxed{[_{wh}\text{ which picture}]}]]]]$

At some stage of the derivation (S_i), there is a wh-phrase *which picture* that is eventually (c-)selected by some predicate (e.g. *like*). At a later derivational stage ($S_i + n$), the wh-phrase undergoes wh-movement. Finally, the adjunct relative clause *that John took* is undergoes late adjunction to higher occurrence of the wh-phrase (or alternatively, to the noun *picture*). There is only one occurrence of the R-expression *John* in any of the representations of the derivation in (57): Neither *John* itself, nor any object *John* is contained in has undergone movement. As a consequence, there is nothing co-referent to *John* that c-commands it, which means that Condition C is not violated. This is a desirable result given that the associated sentence doesn't display the effects of a Condition C violation.

3.3.3 The problem of anti-reconstruction for strictly extending theories

Given that Lebeaux (1991)'s analysis crucially relies on a version of Replacement Merge, an operation that we know is not an Extension Condition Family member, we can definitively conclude that Lebeaux is working within a non-strictly extending theory of syntax. Positing late adjunction (i.e. binary Replacement Merge) is an analytical choice that has some clear

benefits in the context of anti-reconstruction data. Like with crossing dependencies, strictly extending theories run into problems with these data. It's easy to see what the problem is when we restate the facts in the terms of rule interaction.

If we assume that (i) binding relations are checked through some syntactic operation called BOUND? and (ii) movement can bleed BOUND?, then we must conclude from the examples with reconstruction effects, like (58b), that wh-movement counterbleeds BOUND?. That is, BOUND? is ordered before wh-movement.

(58) *Reconstruction for Condition C* (repeated)

a. $\text{He}_{i/*j}$ likes those pictures of John_j.

b. ?*Which picture of John_j does he_j like t?

Furthermore, we can conclude from cases of anti-reconstruction, like (59b) that adjunction counterfeeds BOUND?.

(59) Which pictures that John_i took does he_i like?

If this were the whole story, then a strictly extending analysis of the data would be simple to construct. We simply posit three operations in the Extension Condition Family: Binary c-/l-selection, Binary Adjoin, and BOUND?.[19] And we state an extrinsic order (within a syntactic domain) as follows: Binary c-/l-selection precedes BOUND? which precedes Binary Adjoin. This would be enough to account for the data in (58) and (59). However, it would predict that the following (very unacceptable) example would be perfectly acceptable:

(60) *He_i likes those pictures that John_i took.

The idea here is that the relative clause *that John took*, being an adjunct, wouldn't be in the relevant structure when BOUND? operates. And once the relative clause is Binary Adjoined into the structure, it is too late for BOUND? to raise the Condition C violation flag.

19. Given that BOUND? is clearly different in character from the structure-building operations I've been discussing throughout this thesis, it isn't clear if it could be an Extension Condition Family member. But for the sake of argument, let's assume it is compliant with a strictly extending theory.

There are, of course, always ways around these kinds of problems. Someone staunchly committed to a strictly extending theory can develop mechanisms that make certain syntactic objects invisible for the purpose of Condition C. But the important thing for the purposes of this thesis is that every strictly extending theory will posit a set of intermediate representations for anti-reconstruction derivations that are different from the intermediate representations in the Lebeaux-style, non-strictly extended analyses. For example, when it comes to the sentence in (61), the strictly extending derivation will involve something like the intermediate representation in (61a), and the non-strictly extending derivation will involve something like the intermediate representation in (61b).

(61) Which picture that John$i$ took did he$_j$ like?

a. [[which picture [that John took]]...[likes [which picture [that John took]]]

b. [[which picture]...[likes [which picture]]

Is there any independent empirical evidence that one of these intermediate representations exists and the other doesn't? Personally, I can't figure out any way to test that either of these intermediate representations must be arguments to some syntactic operation. I would, however, like to point out that some researchers have advocated for a lot of Replacement Merge, under the name Wholesale Late Merger (Takahashi and Hulsey, 2009; Stanton, 2016), on the basis of several empirical phenomena distinct from Lebeaux effects. Furthermore, Bošković (2004) argues that floating quantification is only possible when late adjunction is involved. Finally, Zyman (2022) argues that the distribution of stranded wh-phrase modifiers *exactly* and *precisely* follows if they are late adjoined to their associates at particular derivational stages.

3.4 Where we can look for evidence of derivations

Strictly extending theories of syntax appear to have a primacy in syntactic analysis, but there are certain puzzles that natural language presents the analyst with that can be easily solved by adopting operations outside the Extension Condition Family. In this chapter, I make the case that two commonly proposed non-extending operations (i.e. tucking-in movement and late adjunction) provide simple solutions to seemingly complex problems for strictly extending theories. Tucking-in (or Unary Replacement Merge) allows an analyst to account for crossing (movement) dependencies without abandoning or weakening the well-supported locality condition Relativized Minimality. As for late adjunction, it allows a syntactic analyst to account for asymmetrical movement patterns. For example, I show how Lebeaux (1991) accounts for reconstruction and anti-reconstruction effects without complicating a longstanding theory of binding.

I also hoped to demonstrate that strictly and non-strictly extending theories will often have to propose wildly different intermediate representations in derivations to the same natural language sentence. And more, despite the derivations being wildly different, it's surprisingly hard to find any empirical evidence in support of a strictly extending derivation over a non-strictly extending derivation (or vice versa). In fact, for each of the two case studies, I couldn't find any empirical evidence in support of the strictly extending or the non-strictly extending analysis.

The question remains whether or not it's possible to find evidence for strictly or non-strictly extending derivations in syntax or in some other component of grammar. In the next chapter, I argue on empirical grounds that prosodic structure-building is, in fact, derivational. I show that English intonational processes interact with syntax and information-structure in such a way that allows us to diagnose intermediate prosodic representations.

CHAPTER 4

DERIVING PROSODIC STRUCTURE

The investigation of strictly extending and non-strictly extending approaches to syntactic-structure building in Chapters 2 and 3 led to two interesting conclusions. For one, I argue that the two approaches are not strongly equivalent, and as a result, they have the capacity to make distinct sets of empirical predictions. And two, it is difficult (and perhaps impossible) to find empirical support for one approach over the other. It may be that the only arguments in favor of one approach over the other are conceptual. As I discuss in Chapter 3, non-extending operations are generally posited when the analyst prefers to weaken the theory of the kinds of syntactic structure-building operations in order to hold onto a stronger version of some other aspect of syntactic theory. In the case of multiple wh-movement in Bulgarian, we considered an analysis that posited a non-extending operation in order to maintain the prevailing theory of locality of movement. In the case of anti-reconstruction effects, we look briefly at an analysis that posits a non-extending operation in order to maintain a longstanding theory of binding.

In this chapter, I turn my attention from theories of syntactic structure building to those of prosodic structure building. For one, I argue against a common claim that the syntactic and prosodic structure of ditransitive verb phrases in English are always non-isomorphic. In fact I provide empirical arguments of both the syntactic kind (§4.1.2-4.1.4) and the phonological/prosodic kind (§4.2) that support the idea that the two structures are almost always isomorphic. I present novel data in §4.2.1-§4.2.3 that syntactic and prosodic structure of ditransitive verb phrases can be non-isomorphic under very particular sets of conditions.

Ultimately, the primary objective of this chapter is to argue for a non-strictly extending derivational theory of prosodic structure building. I show that cases of non-isomorphism between syntactic and prosodic structure as well as cases of ineffable expressions provide ev-

idence of intermediate representations that have been destroyed by non-extending structure-building operations. These arguments are distributed between §4.2.1-4.2.3.

4.1 Ditransitive mismatch and the syntax of verb phrases

Ditransitive mismatch is a term (coined by Kalivoda (2018), I believe) used to describe the purported non-isomorphism between the syntactic representations of ditransitive verb phrases and their corresponding prosodic representations. Ditransitive mismatch is typically stated as a puzzle. The alternative hypothesis is that syntactic and prosodic representations should be isomorphic. On the basis of some empirical arguments, researchers conclude that the alternative hypothesis is false, and they develop an analysis of the correspondence between the two structures that relies heavily on prosodic well-formedness constraints.

As with any claim about the relationship between syntactic and prosodic representations, the claim that ditransitive mismatch exists is only as strong as the empirical arguments in support of both the posited syntactic representation and posited prosodic representation. In §4.1.1, I will present a brief overview of the arguments for ditransitive mismatch, the kinds of ditransitive verb phrases in English, the possible prosodic phrasings of ditransitive verb phrases, and explain what it means for there to be a mismatch between the syntactic and prosodic representations of ditransitive expression.

What will be clear is that ditransitive mismatch is a puzzle if the only possible syntactic structure of ditransitive verb phrases is the descending (or right-branching) structure. If an ascending (or right-branching) structure is a viable syntactic analysis of ditransitive verb phrase, then ditransitive mismatch isn't a puzzle. In 4.1.2, I review Barss-Lasnik effects (i.e. asymmetries between internal objects of ditransitives). It has been argued that these effects support the descending verb phrase analysis of ditransitives (Larson, 1988, 1990). I will argue that Barss-Lasnik effects provide fairly compelling evidence against the view that the ascending verb phrase structure is the sole syntactic representation of ditransitive verb

phrases. But in §4.1.3, I will present arguments that support the existence of the ascending verb phrase structure. Furthermore, in §4.1.4, I will present evidence that shows either that (i) classical analyses if Barss-Lasnik effects are flawed or (ii) a single expression can have two structural analyses in parallel. I conclude that there isn't strong evidence that ditransitive verb have a single descending representation, and therefore that ditransitive mismatch doesn't exist: Ditransitive verb phrases have isomorphic syntactic and prosodic representations.

4.1.1 What ditransitive mismatch is

I'll define *ditransitive verb phrase* as any string comprising of a verb immediately followed by two consecutive internal arguments and any number of verb phrase modifiers (i.e. in head-initial languages).[1] Taking English as an example, we can see three different kinds of ditransitive verb phrases in (1).

(1) a. Nancy [gave]$_V$ [Tony]$_{IA_1}$ [the bike]$_{IA_2}$.

b. Nancy [showed]$_V$ [the bike]$_{IA_1}$ [to Tony]$_{IA_2}$.

c. Nancy [talked]$_V$ [about the bike]$_{IA_1}$ [with Tony]$_{IA_2}$.

The construction in (1a) is often referred to as a double object construction; the two internal arguments in double object constructions are noun/determiner phrases. For example, the ditransitive verb phrase *gave Tony the bike*, which is a string that can be analyzed as a verb *gave* followed by the determiner/noun phrase *Tony* and the linearly final determiner/noun phrase *the bike*. In (1b), we find an example of a prepositional dative construction. It involves a ditransitive verb phrase *showed the bike to Tony*, where the verb is *showed*, the linearly first internal argument is a noun/determiner phrase *the bike*, and the final internal argument is the prepositional phrase *to Tony*. A double prepositional phrase construction

1. The study of ditransitive mismatch has focused mostly on head-initial languages, so I'll continue this trend and put head-final languages aside for the remainder of this discussion.

is shown in (1c). The ditransitive verb phrase is the string *talked about the bike with Tony*, the verb is *talked*, the linearly first internal argument is the prepositional phrase *about the bike*, and the final internal argument is the prepositional phrase *with Tony*. Going forward, I'll call the linearly first internal argument in a ditransitive verb phrase IA_1 and the linearly second IA_2. As you can see, this terminological shorthand is reflected in the examples in (1).

Selkirk (1984, p.293) noted that English ditransitive constructions can have a variety of intonational phrasings. I'll cover intonational phrases in more detail in §4.2.1. For now, you can understand an intonational phrase as a pitch contour with one very prominent peak. Selkirk's observation is that English ditransitive constructions can have one or several intonational phrases. I've presented her data in (2). A pair of parentheses denotes an intonational phrase domain, and you can read these sentences as if a very prominent pitch peak occurs on the final stressed syllable in each intonation phrase.

(2) a. (Jane gave the book to Mary)

b. (Jane) (gave the book to Mary)

c. (Jane gave the book) (to Mary)

d. *(Jane gave) (the book to Mary)

e. *(Jane) (gave) (the book to Mary)

f. (Jane) (gave the book) (to Mary)

g. (Jane) (gave) (the book) (to Mary) (Selkirk, 1984, p.293)

If we take internal arguments as atomic units, we can see that almost every possible substring of the sentence *Jane gave the book to Mary* can be an intonational phrase.[2] The only exception is the substring *the book to Mary*, which is the substring consisting of (and only of) the two internal arguments (2d-e). When we zoom in on the verb phrase, which allows

2. Two points should be addressed here. One, for ease of exposition, I'm ignoring the possibility that intonational phrasing can be recursive. Two, each of the (possible) phrasings in (2) are only natural in certain information-structural contexts.

us to ignore the added complications of having the subject as a variable, we find that only one of four possible phrasings is unavailable in every information-structural context—that is, case where the two internal arguments are phrased together to the exclusion of the verb.

(3) a. (gave the book to Mary)

b. (gave the book) (to Mary)

c. (gave) (the book) (to Mary)

d. *(gave) (the book to Mary)[3]

Selkirk only presents this pattern for prepositional dative constructions, but the pattern holds for both double object constructions (4) and double prepositional phrase constructions (5) as well. I personally have a very difficult time judging the acceptability of many of these sentences, so I am presenting Erik Zyman's (p.c.) judgements here and throughout this chapter—unless examples are accompanied by a citation.[4]

(4) a. [Jane] (gave Mary the book)

b. [Jane] (gave Mary) (the book)

c. [Jane] (gave) (Mary) (the book)

d. *[Jane] (gave) (Mary the book)

(5) a. [Jane] (talked about the book to Mary)

b. [Jane] (talked about the book) (to Mary)

c. [Jane] (talked) (about the book) (to Mary)

d. *[Jane] (talked) (about the book to Mary)

3. Steedman (1991, p.288) reports that examples with this intonational phrasing are perfectly acceptable to his ear.

4. I recognize that Erik may be a biased participant because he is a member of this dissertation's committee. It gives me confidence that he has provided consistent judgements of sentences of the same kind but with different lexical items, and more, there were often large gaps of time between elicitation sessions.

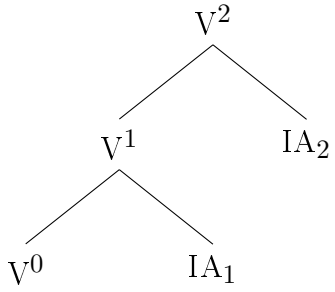
According to Kalivoda (2018), the generalization that the two internal arguments of ditransitives can't be phrased together (i) isn't unique to intonational phrasing, (ii) holds across kinds of ditransitive, and (iii) is robust cross-linguistically. For a very nice overview of cross-linguistic evidence of the generalization, I refer you to Kalivoda (2018, §3.2). I'll refer to this generalization as the Ditransitive Phrasing Generalization, which I've defined informally in (6).

(6) **Ditransitive Phrasing Generalization**

The two internal arguments of a ditransitive verb phrase cannot form a prosodic domain to the exclusion of their selecting verb: *(V)(IA_1 IA_2).

Any explanation of the Ditransitive Phrasing Generalization crucially depends on the syntactic analysis of ditransitive verb phrases. Let's assume that ditransitive verb phrases have an ascending (or left-branching) representation, as depicted in (7).

(7) *Ascending ditransitive*



The above ascending verb phrase has five constituents: the entire verb phrase ($V^{2^{SO}}$); the constituent that contains the verb and IA_1 ($V^{2^{SO}}$); the constituents corresponding to the verb (V^0), IA_1, and IA_2.

If the all verb phrases are analyzable as ascending syntactic structures, as in (7), then we can easily explain the Ditransitive Phrasing Generalization. The generalization follows from (what I call) the Constituent Mapping Conjecture, which I've defined in (8).

(8) **Constituent Mapping Conjecture**

A string w is a possible prosodic domain (e.g. an intonational phrase domain) iff w is analyzable as a syntactic constituent.

The Constituent Mapping Conjecture says that prosodic domains always correspond to some syntactic constituents, but not every syntactic constituent needs to be realized as a prosodic domain. Together with the assumption that (i) every substring of an expression must be part of some intonational phrase domain and (ii) ditransitive verb phrases have an ascending structure, all of (and only) the possible prosodic phrasings of ditransitive verb phrase follow from the Constituent Mapping Conjecture.

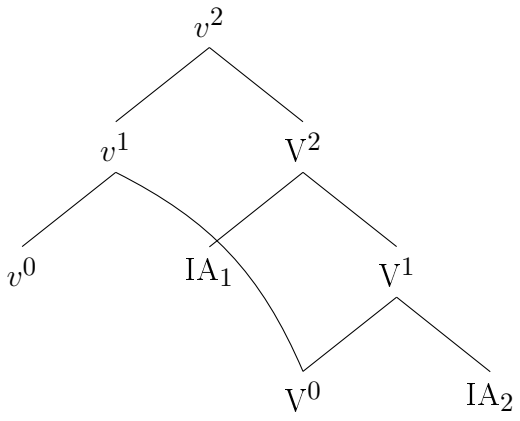
When the entire verb phrase is mapped to a single intonation phrase domain (e.g. the (a) examples in (3)-(5)), the intonational phrase phrase corresponds to the entire verb phrase constituent ($V^{2^{SO}}$). We see in the (a) examples that the verb and linearly first internal arguments are phrased together to the exclusion of the second internal argument, as in (V IA_1) (IA^2). In this case, the intonational phrase domain (V IA_1) corresponds to the syntactic constituent $V^{1^{SO}}$, and the phrase (IA_2) corresponds to the syntactic constituent IA_2^{SO}. In the (c) examples, each of the verb, IA_1, and IA_2 form their own independent intonational phrase domains—that is, (V)(IA_1)(IA_2). Each of these intonational phrase domains correspond to the syntactic constituents $V^{0^{SO}}$, IA_1^{SO}, and IA_2^{SO}, respectively. All of this is to say that each of the possible intonational phrase domains correspond to some constituent in the ascending verb phrase structure, which means they pose no problem for the Constituent Mapping Conjecture.

Now for the illicit (d) examples, where the two internal arguments are prosodically phrased together to the exclusion of the verb: *(V)(IA_1 IA_2). It follows from our set of assumptions that the verb should be able to form its own intonational phrase domain because to corresponds to the constituent $V^{0^{SO}}$ in (7). However, the phrase comprised of the two internal arguments (IA_1 IA_2) doesn't correspond to any constituent in the ascending

verb phrase. Therefore, we predict that it is not a possible intonational phrase domain. So given sufficient evidence for the ascending verb phrase structure in (7) for ditransitives, the Ditransitive Phrasing Generalization clearly follows from the Constituent Mapping Conjecture.

While there is evidence in support of the ascending verb phrase structure in (7), there is also plenty of evidence that ditransitive verb phrases should be analyzed as having a descending (or right-branching) representation (9). And almost as a rule, analyses of the relation of between syntax and prosodic representations assume that ditransitive verb phrases have a descending structure (Selkirk, 2011; Cheng and Downing, 2016; Kalivoda, 2018; Lee and Selkirk, 2022).

(9) *Descending Ditransitive*



The widespread adoption of the descending ditransitive verb phrase was inspired by Larson (1988, 1990). According to this kind of analysis,[5] the linearly second internal argument (IA_2) is a complement to the main verb (V^0), while the linearly initial internal argument

5. Some of the details of Larson have been abandoned over. For him, there are differences between the deep structures of double object constructions and prepositional dative constructions. He argues that IA_1 c-commands IA_2 in the deep structure of the prepositional dative construction, and that the structural relation (i.e IA_1 c-commands IA_2) is transformationally derived in the double object construction; before the transformation takes place, IA_2 asymmetrically c-commands IA_1. Generally, those who staunchly hold to the descending analysis of ditransitives assume that the there is no point in the derivation of any ditransitive verb phrase where IA_2 c-commands IA_1.

is a specifier to the main verb. In order to derive the correct linear order, it is posited that a functional head v that is structurally higher than the verb phrase. The main verb undergoes movement to adjoin to v, and as a result, is realized phonologically in the position of v. Because the main verb can only be pronounced in one position (by assumption), the phonologically realization of the vP (i.e. $v^{2^{SO}}$ in (9)) is the string V-IA$_1$-IA$_2$. Furthermore, the substring IA$_1$-IA$_2$ corresponds to the constituent V$^{2^{SO}}$, which is commonly referred to as the (Larsonian) headless VP.

If the descending verb phrase structure is the correct syntactic analysis of ditransitive verb phrases, then it follows from the Constituent Mapping Conjecture that the two internal arguments should be able to form an intonational phrase domain to the exclusion of the verb. After all, the string IA$_1$-IA$_2$ corresponds to a syntactic constituent. However, we know this isn't the case as there is empirical evidence that this string is not a possible intonation phrase domain. So in the view of those that adopt the descending verb phrase structure, there is a mismatch between the prosodic representation of ditransitive verb phrases and their corresponding syntactic representation because there is a syntactic constituent (namely, the headless VP) whose corresponding string IA$_1$-IA$_2$ isn't a possible intonational phrase domain. This mismatch is what Kalivoda 2018 calls ditransitive mismatch.

Mismatches between the syntactic and prosodic representations warrant a more complex theory of the relation between the two kinds of representation. Logically, it isn't possible to simply adopt the Constituent Mapping Conjecture (or something of its kind). Among the several theories that have been proposed to account for syntax-prosody mismatches are Align/Wrap Theory (Selkirk, 1981, 1986, 1996; Truckenbrodt, 1995, 1999; McCarthy and Prince, 2004), Match Theory (Selkirk, 2011; Kratzer and Selkirk, 2020), and Command Theory (Kalivoda, 2018). Each of these theories assume that there is an interaction between rules/constraints that (i) map syntactic constituents/relations directly to prosodic domains (i.e. faithfulness) and (ii) prosodic well-formedness constraints that are insensitive syntactic

structure (i.e. markedness).

We are now in a good position to ask whether or not ditransitive mismatch is real puzzle. Because the puzzle of ditransitive mismatch is contingent on the syntactic analysis of ditransitive verb phrases, the goal here is to establish whether or not there is a preponderance of evidence that ditransitive verb phrases can only be analyzed as having a descending structure, like the one in (9), or if there is enough evidence to argue that ditransitive verb phrases can be analyzed as having an ascending structure, like the on in (7). As we will see in §4.1.2, there is evidence that supports a descending representation and poses problems for ascending structures. However, there is just as much evidence that (i) supports an ascending analysis of ditransitive verb phrases and (ii) is seemingly incompatible with with a descending structure.

It isn't necessary in order to argue against the existence of ditransitive mismatch that ditransitive verb phrases have only one structure and that that structure is the ascending one. It could be that ditransitive verb phrases have two parallel structures—an ascending structure and a descending structure. In fact, this is what I will argue in §4.1.4, following Pesetsky 1995.

4.1.2 Evidence for the descending verb phrase

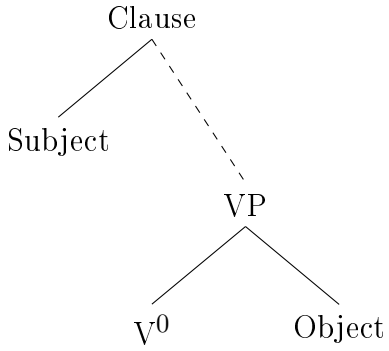
Larson (1988, 1990) presents several arguments in favor of the descending analysis of ditransitive verb phrases. Many of the arguments are based on observations from Barss and Lasnik 1986 that suggest the linearly initial internal argument of a ditransitive (IA_1) asymmetrically c-commands the second (IA_2); Barss and Lasnik present a number of binding and movement asymmetries.[6] Barss and Lasnik's observation don't directly support the claim that the two internal arguments in a ditransitive construction form a constituent to the exclusion of the

6. Barss and Lasnik (1986) only investigate double object constructions, but Larson (1988, 1990) shows that their arguments hold for prepositional dative constructions. This section won't discuss double prepositional phrase ditransitives, but for some observations about Barss-Lasnik effects in these double PP ditransitives, I refer you to Janke and Neeleman (2012, p.180).

verb. However, their observations are not expected if ditransitive verb phrases can only be analyzed as the ascending structure in (7).

All of the arguments in Barss and Lasnik 1986 assume that subjects asymmetrically c-command objects—that is, the structure of a monotransitive clause has the basic shape in (10).

(10)



The fact that subjects asymmetrically c-command their clausemate objects is important to all of their arguments because asymmetric c-command is central to their explanation of certain binding, licensing, and movement asymmetries. When we consider the example in (11a), we see that object reflexive pronouns (like *himself*) are licit when they co-occur with a co-referent subject (*John/he*). In contrast, subject reflexive pronouns are not made licit by the occurrence of a co-referent object clausemate (11b).

(11) *Condition A asymmetry (Baseline)*

a. $\{^{\text{John}}_{\text{he}}\}_i$ likes himself$_i$.

b. *Himself$_i$ likes John$_i$.

c. Sue expected $\{^{\text{John}}_{\text{him}}\}_i$ to like himself$_i$.

d. *Sue expected himself$_i$ to like John$_i$.

There isn't a general ban on subject reflexive pronouns. In exceptional case marking constructions, like (11c), the (so called) exceptionally case-marked subject can be a reflexive

pronoun; however, as is shown in (11d), its clausemate object doesn't license its occurrence.

Barss and Lasnik's explanation for these licensing asymmetries is the prevailing explanation to this day. They argue that reflexive pronouns need to be c-commanded by a co-referent clausemate; this requirement is known as (Binding) Condition A (Chomsky, 1981). Subjects asymmetrically c-command their clausemate objects, as clauses have the structure in (10); therefore subjects can bind objects (within a clause) for the purposes of Condition A, but objects can't bind subjects for Condition A.

Barss and Lasnik 1986 presents six arguments that that the linearly first argument (IA_1) in a double object construction asymmetrically c-commands the linearly second (IA_2). The arguments all have the same form. First, they show that some subject-object asymmetry exists,[7] which is assumed to be explained by asymmetric c-command, as with Condition A. Next, they show that an analogous asymmetry exists between the two internal objects of a double object construction—IA_1 can bind/license IA_2, and movement of IA_1 bleeds movement of IA_2.[8] As is the case with the subject-object asymmetries, it follows from the theories of binding, licensing, and locality of movement that IA_1 asymmetrically c-commands IA_2. We'll walk through a couple more of their arguments as well as Larson's extension to prepositional dative constructions.[9]

We've already seen the argument that a reflexive pronoun must be c-commanded by a co-referent clausemate (i.e. Condition A). Barss and Lasnik (1986) note that, in double object constructions, IA_1 can bind IA_2 for the purposes of Condition A (12a), but IA_2 can't bind IA_1 (12b). It follows, then, that IA_1 asymmetrically c-commands IA_2.

7. In most cases, it is more accurate to say that they assume the reader knows that the subject-object asymmetry exists.

8. See Chapter 3 for a discussion of movement asymmetries and the Superiority Condition.

9. Larson's arguments have the same form as Barss and Lasnik's.

(12) *Condition A asymmetry (Double Object)* (Barss and Lasnik, 1986, p.347)

a. I showed $\{^{\text{John}}_{\text{him}}\}_i$ himself$_i$ (in the mirror).

b. *I showed himself$_i$ John$_i$ (in the mirror).

Larson (1988, 1990) shows that the Condition A asymmetry also holds for prepositional dative constructions. In (13a), the linearly first internal argument (*Mary* or *her*) can bind the object of the preposition in the second internal argument for Condition A, but the opposite isn't true (13b).

(13) *Condition A asymmetry (Prepositional Dative)* (Larson, 1988, p.338)

a. I $\{^{\text{presented}}_{\text{showed}}\}$ $\{^{\text{Mary}}_{\text{her}}\}_i$ to herself$_i$.

b. *I $\{^{\text{presented}}_{\text{showed}}\}$ herself$_i$ to Mary$_i$.

It follows that the linearly first internal argument (IA_1) of prepositional ditransitive constructions asymmetrically c-commands the linearly second internal argument (IA_2).

Another of Barss and Lasnik's arguments is based on patterns of reciprocal binding. The examples in (14) demonstrate that a plural subject (e.g. *Two trainers*) can bind (or license) a reciprocal (e.g. *each other*) in an object, while a plural object can't bind a reciprocal in a subject.

(14) *Reciprocal binding asymmetry (Baseline)*

a. [Two trainers/*the trainer]$_i$ petted [each other]$_i$'s lions.

b. *[Each other]$_i$'s trainers petted [two lions]$_i$.

The above subject-object asymmetry is understood to be a result of the asymmetric c-command relation between the subject and object of the clause. On the basis of this theory of reciprocal binding, Barss and Lasnik (1986) argue that the linearly first internal argument (IA_1) in a double object construction asymmetrically c-commands the second (IA_2). This is because a plural IA_1 can bind a reciprocal in IA_2 (15a), but the opposite isn't possible (15b).

(15) *Reciprocal binding asymmetry (Double Object)*

a. I showed [two lions]$_i$ [each other]$_i$'s trainers.

b. *I showed [each other]$_i$'s trainers [two lions]$_i$.

Larson (1988, 1990) argued, in the same way that Barss and Lasnik did, that the first internal argument in a prepositional dative construction c-commands the linearly second (16).

(16) *Reciprocal binding asymmetry (Prepositional Dative)*

a. I sent [the boys]$_i$ to [each other]$_i$'s parents.

b. *I sent [each other]$_i$'s boys to [the parents]$_i$.

In (16a), the plural IA$_i$ binds the reciprocal in the prepositional phrase (IA$_2$), but as (16b) shows, it's impossible for a plural object of the preposition (in IA$_2$) to bind the reciprocal in IA$_1$.

We can arrive at the same conclusion about the relation between the two internal arguments of ditransitives on the basis of variable binding patterns. The pair of examples in (17) demonstrate that a quantified subject, like *every parent* in (17a), can bind a pronoun in the object (e.g. *their*). In other words, the sentence can be interpreted as 'parent x loves parent x's child, parent y loves parent y's child, ...'. This kind of interpretation is impossible if the subject contains a pronoun and the object is quantified, as we can see in (17b). The only possible interpretation is one where there is a single parent (e.g. Jan's parent) that loves every child—that is, Jan's parent loves every child.

(17) *Variable binding asymmetry (Baseline)*

a. [Every parent]$_i$ loves their$_i$ child.

b. *Their$_i$ parent loves [every child]$_i$.

As with Condition A, a variable binding interpretation of a pronoun is theorized to be sensitive to c-command; in particular the quantified phrase (e.g. *every x*) must c-command

the pronoun (e.g. *their y*). Barss and Lasnik 1986 showed that IA_1 of a double object construction can bind a pronoun in IA_2, but IA_2 can't bind a pronoun in IA_1, as can be seen in (18). Larson shows that the same is true for the prepositional dative construction (19); a quantified IA_1 (the DP) can bind a pronoun in IA_2 (the PP), but the reverse isn't possible.

(18) *Variable binding asymmetry (Double Object)* (Barss and Lasnik, 1986, p.348)

a. I denied [each worker]$_i$ his$_i$ paycheck.

b. *I denied its$_i$ owner [each paycheck]$_i$.

(19) *Variable binding asymmetry (Prepositional Dative)* (Larson, 1988, p.338)

a. I $\{^{\text{gave}}_{\text{sent}}\}$ [every check]$_i$ to its$_i$ owner.

b. ??I $\{^{\text{gave}}_{\text{sent}}\}$ his$_i$ paycheck to [every worker]$_i$.

In addition to Condition A and variable binding, Barss and Lasnik 1986; Larson 1988, 1990 argue that IA_1 c-commands IA_2 on the basis of Condition B, Condition C, weak crossover, reciprocal licensing, movement asymmetries, and negative polarity item licensing.

It has been noted (perhaps most famously by Pesetsky (1995)) that Barss and Lasnik- and Larson-style arguments lead us to conclude that the internal arguments in ditransitive constructions asymmetrically c-command verb phrase modifiers as long as the modifiers linearly follow the internal arguments. In (20) I've provided three pairs of examples. Each example is a prepositional dative construction. In the first of each pair, a plural internal argument or adjunct licitly binds a reciprocal in a modifier to its right.[10] The second example of each pair is there to show that a plural noun/determiner phrase can't bind a reciprocal to its left.[11]

10. As far as I know, binding possibilities between elements in separate verb phrase modifiers, like (20e-f), have never been reported. In Pesetsky 1995, it is only shown that a plural internal argument can bind a reciprocal in a modifier and that a plural modifier can't bind a reciprocal in an internal argument.

11. The patterns in the following data arise for all c-command-based diagnostics (i.e. Conditions A-C, variable binding, NPI licensing, etc.). I've used reciprocals here because, in contrast to many other binding phenomena, the c-command theory of reciprocal binding is fairly robust.

(20) *Binding into VP-adjuncts (Prepositional Dative)*

a. John showed [the teachers]$_i$ to the student in [each other]$_i$'s classrooms on Tuesday.

b. *John showed [each other]$_i$'s classrooms to the student for [the teachers]$_i$ on Tuesday.

c. John showed the student to [the two cleaners]$_i$ in [each other]$_i$'s classrooms on Tuesday.

d. *John showed the student to [each other]$_i$'s cleaners in [the classrooms]$_i$ on Tuesday.

e. The research assistant showed the photo to the participant for [the primary researchers]$_i$ in [each other]$_i$'s labs.

f. *The research assistant showed the photo to the participant in [each other]$_i$'s labs for [the primary researchers]$_i$.

In example (20a), a plural IA_1 binds a reciprocal in a following verb phrase modifier, but as (20b) shows, the reverse binding relationship isn't possible. Examples (20c-d) demonstrate that a plural in IA_2 can bind a reciprocal in an immediately following verb phrase modifier. Examples (20e-f) reveal that a plural in verb phrase modifier can bind a reciprocal in a modifier to its right, but not a reciprocal to its left.

The three pairs of examples in (21) demonstrate that, in the double object construction, rightward verb phrase modifiers can be bound into as well. I won't walk through all of the examples, but it should be easy to see the double object paradigm parallels the prepositional dative paradigm in (20).

(21) *Binding into VP-adjuncts (Double Object)*

a. John showed [the teachers]$_i$ the book about North American beavers in [each other]$_i$'s classrooms on Tuesday.

b. *John showed [each other]$_i$'s students the book about North American beavers for [the teachers]$_i$ on Tuesday.

c. John showed the student [the two teachers]$_i$ in [each other]$_i$'s classrooms on Tuesday.

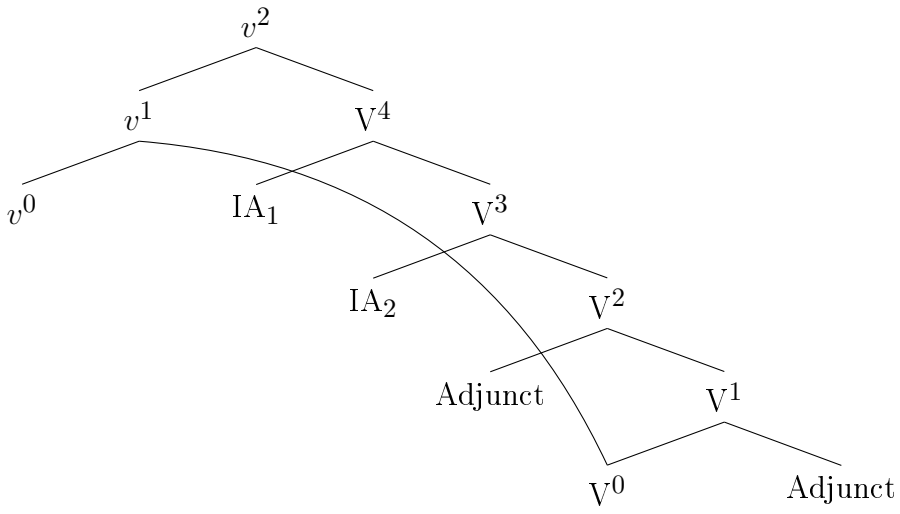
d. *John showed the student [each other]$_i$'s cleaners in [the classrooms]$_i$ on Tuesday.

e. The research assistant showed the participant the photo for [the primary researchers]$_i$ in [each other]$_i$'s labs.

f. *The research assistant showed the participant the photo in [each other]$_i$'s labs for [the primary researchers]$_i$.

As I mentioned above, if we take Barss and Lasnik's arguments to their logical conclusion, then it must be the case that internal arguments asymmetrically c-command (rightward) verb phrase modifiers. Several researchers have adopted this conclusion in some way or another (Pesetsky, 1995; Phillips, 1996, 2003). They take all of the binding/licensing evidence in this section, as well as in Barss and Lasnik 1986; Larson 1988, 1990, as evidence that rightward is downward (Kayne, 1994). The details of the various approaches differ, but I will represent this kind of analysis with the very descending tree in (22).

(22) *Very Descending Ditransitive (with adjuncts)*



Before we move onto the next section, which will present evidence in support of the ascending ditransitive verb phrase, I think it's important to mention one more of Larson's (1988; 1990) arguments because it directly addresses the question of whether or not the headless verb phrase is a constituent. It is typically assumed that only syntactic constituents can be coordinated. Under this assumption, if the headless verb phrase is a constituent, it should (barring some confounding factor) be coordinate-able. And as Larson notes, it is coordinate-able, as can be seen in (23).

(23) *Coordinated headless VP*

a. John sent [a letter to Mary] and [a book to Sue]. (Larson, 1988, p.345)

b. John sent [Mary a letter] and [Sue a book].

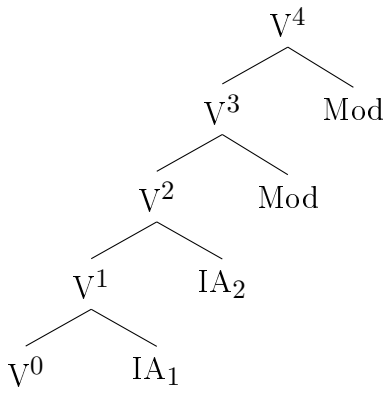
In (23a), we can see an example of Larson's where the two internal arguments of a prepositional dative construction are coordinated to the exclusion of the verb. I've also provided (23b) in order to show that the two internal arguments of a double object construction can be coordinated. Jackendoff (1990, pp.439-440), in his response to Larson 1988, argues that the examples are better analyzed as instances of a well-known kind of non-constituent coordination (called *gapping*). I think Jackendoff's argument are fairly convincing. I won't

go over his arguments here. If you're interested, I recommend reviewing both Jackendoff's arguments and Larson's 1990 response to Jackendoff.

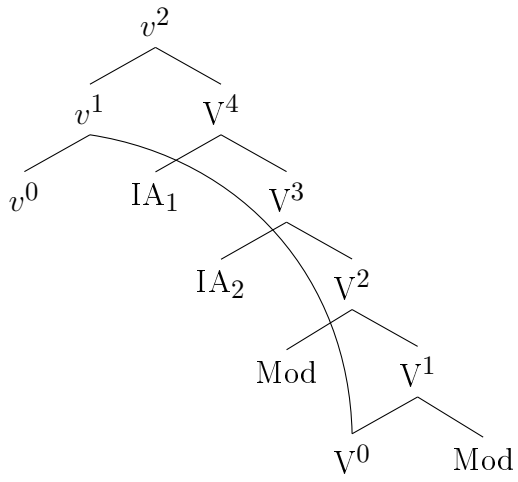
4.1.3 Evidence for the ascending verb phrase

In the previous section (§4.1.2), I presented arguments on the basis of binding/licensing asymmetries that the structure of ditransitives verb phrases should be analyzed as in (25). In this section, I will present arguments that support the idea that ditransitive verb phrases should be analyzed as having an ascending structure, like the tree in (24).

(24) *Ascending ditransitive*



(25) *Descending Ditransitive*



These arguments rest on the assumptions that (i) fronting around *though* involves movement of phrasal[12] syntactic objects with a *Basic* (i.e. category) feature V or v, and (ii) verb phrase ellipsis targets phrasal syntactic constituents with a *Basic* (i.e. category) feature V or v.

Starting with the prepositional dative construction, we will consider which substrings of the verb phrase can be fronted (26) around *though*. By fronting X around *though*, I mean

12. A syntactic constituent is phrasal as long as its not just a head. In the terms of the graph-theoretic formalization in Chapter 2, a phrasal syntactic object is any syntactic object that consists of more than one node. And in terms of the set-theoretic formalization in Collins and Stabler 2016, a phrasal syntactic object is any set of syntactic objects (that isn't a workspace).

that X is placed in a position immediately before the element *though*, which is assumed to be in the left periphery of a clause,[13] and X is not pronounced in its canonical position within the clause.

(26) a. *But [give] though he did the books to the children in the garden on Tuesday...

b. But [give the books] though he did to the children in the garden on Tuesday...

c. But [give the books to the children] though he did in the garden on Tuesday...

d. But [give the books to the children in the garden] though he did on Tuesday...

e. But [give the books to the children in the garden on Tuesday] though he did...

...they were disappointed not to have received toys (in the library) (on Monday).

In (26a), we see that the verb *give* can't be fronted around *though*. This is expect for both the ascending and descending structures because *give* corresponds to a non-phrasal syntactic object, and by assumption, only phrasal syntactic objects can be fronted around *though*. It is possible for the string *give the books* to be fronted around *though*, as you can see in (26); this string corresponds to the syntactic object $\mathrm{V}^{1^{SO}}$ in the ascending structure, but there is no syntactic object contained in the descending structure that corresponds to the string *give the books*. Therefore the ascending structure analysis makes the correct prediction with respect to (26), while the descending structure analysis doesn't make the correct prediction. Examples (26c-d) in the paradigm support the ascending structure analysis and pose problems for the descending structure analysis. The strings *give the books to the children*, and *give the books to the children in the garden*, which correspond to syntactic objects $\mathrm{V}^{2^{SO}}$, $\mathrm{V}^{3^{SO}}$, and $\mathrm{V}^{4^{SO}}$ (respectively) in the ascending structure in (24), can all be fronted around *though*. But there

13. There have been several analyses of fronting around *though*, and each analysis has assumed that the fronted string is a constituent. Chomsky (1973, p.112, fn.32) and Baltin (1982, §5.1) analyze the construction as a movement transformation, where the moved element has to be a constituent with certain formal features. Furthermore, Baltin (1982) explicitly analyzes *though* as a preposition that selects for a sentential complement, and therefore that the landing site of the fronted string is the specifier of a prepositional phrase. More recently, Thoms and Walkden 2019 analyze the string that has been fronted around *though* as a base-generated constituent that antecedes an ellipsis site. The details of the analysis don't really matter for the larger argument of this section. What is important is that the fronted string is, in fact, a constituent.

isn't a syntactic object contained in the descending structure in (25) that corresponds to any of those three strings. Both the ascending and descending structure analyses make the right prediction with respect to (26e), where the string *give the books to the children in the garden on Tuesday* is fronted around *though*, because the string corresponds to a syntactic object in each structure: in the ascending structure, it is $V^{4^{SO}}$, and in the descending structure, it is $v^{2^{SO}}$.

Fronting around *though* provides more testing ground with respect to the two structural analyses we are considering. It follows from the ascending structure analysis (and the assumption that only (V/v phrasal) syntactic objects can be fronted around *though*) that the examples in (30) should be unacceptable, and they are. The descending analysis (in conjunction with assumption about what can front) correctly predicts that (27a-b) should unacceptable; however, it incorrectly predicts that (27c) should be acceptable.

(27) a. *But [the books to the children] though he did give in the garden on Tuesday...

b. *But [the books to the children in the garden] though he did give on Tuesday...

c. *But [the books to the children in the garden on Tuesday] though he did give...

...they were disappointed not to have received toys (in the library) (on Monday).

With respect to the prepositional dative construction, patterns of acceptable and unacceptable instances of fronting around *though* provide a preponderance of evidence in favor of the ascending structure in (24). Although the descending structure in (25) makes some correct predictions with respect to the fronting patterns, it is very far from providing a comprehensive analysis of the entire paradigm.

Patterns of acceptable verb phrase ellipsis targets also provide us with a fruitful testing ground. Previous works that used verb phrase ellipsis to test the structure of ditransitive verb phrases include Radford 1988, Pesetsky 1995, and Janke and Neeleman 2012. As a reminder, we are assuming that a string can be a target of verb phrase ellipsis if and only if it is a phrasal syntactic object of category V or v. When a string is targeted by verb phrase ellipsis,

the string isn't pronounced.[14] The resulting silence can receive an interpretation from an appropriate linguistic antecedent in the discourse context, which is why the sentences in (28) are judge in the context of the preceding expression *John gave the books to the children in the garden on a Tuesday.* Consider the examples in (28).

(28) John gave the books to the children in the garden on a Tuesday...

a. *...long before Mary did ⟨give⟩ the DVDs to the seniors in the library on a Monday.

b. ...long before Mary did ⟨give the books⟩ to the seniors in the library on a Monday.

c. ...long before Mary did ⟨give the books to the children⟩ in the library on a Monday.

d. ...long before Mary did ⟨give the books to the children in the garden⟩ on a Monday.

e. ...long before Mary did ⟨give the books to the children in the garden on a Tuesday⟩.

As for (28a), where only the verb *give* is targeted by ellipsis, the example is unacceptable. This is predicted under any structural analysis because *give* can only correspond to a non-phrasal syntactic object and I am assuming that verb phrase ellipsis only targets phrasal syntactic objects. Example (28b-e) involve ellipsis targets that correspond to constituents of the ascending structure in (24): *give the books* ($V^{1^{SO}}$), *give the books to the children* ($V^{2^{SO}}$), *give the books to the children in the garden* ($V^{3^{SO}}$), and *give the books to the children in the garden on a Tuesday* ($V^{4^{SO}}$). Therefore, the ascending structure analysis makes the correct prediction for every example.[15] The last of these four strings (i.e. *give the books to the*

14. I place the target of ellipsis in angle brackets (e.g. ⟨TARGET⟩).

15. Erik Zyman (p.c.), whose judgements are presented throughout the chapter, noted that examples with stranded arguments are degraded without pitch accents on the stranded arguments. He also noted that this

children in the garden on a Tuesday) corresponds to the entire descending structure in (25), so the descending analysis correctly predicts that it should be a licit target of verb phrase ellipsis. However, none of the other three licit targets in (28) corresponds to constituents in the descending structure. Therefore the descending structure analysis makes several incorrect predictions with respect to possible targets of verb phrase ellipsis.

With respect to the prepositional dative construction, it is clear that the ascending structure analysis provides an explanation of the fronting around *though* data and the verb phrase ellipsis data, and it is also clear that the descending structure analysis doesn't. Now we can shift our focus to the double object construction. We will see that the fronting and ellipsis data parallel the prepositional dative data, and therefore provides evidence in favor of the ascending structure and against the descending structure for the double object construction.

In (29), you can see the part of the fronting around *though* paradigm for the double object construction that corresponds to the prepositional dative examples in (26). As before, fronting the verb alone is unacceptable. This follows from the assumption that only phrasal categories can be fronted. The remaining examples (29b-e) demonstrate that any verb initial string can be fronted, as was the case with the prepositional dative construction.

(29) a. **But [give] though he did the children the books (about North American beavers) in the garden on Tuesday...

b. But [give the children] though he did the books *(about North American beavers)[16] in the garden on Tuesday...

c. But [give the children the books (about North American beavers)] though he did in the garden on Tuesday...

---

fact could be related to John Bowers's observation that pseudogapping with two internal argument remnants is degraded unless both of the internal arguments receive 'the appropriate focus intonation' (Bowers, 2018, p.262, fn.2).

16. Erik finds it much more difficult to strand determiner phrases (relative to prepositional phrases) unless the determiner phrase is heavy.

d. But [give the children the books (about North American beavers) in the garden] though he did on Tuesday...

e. But [give the children the books (about North American beavers) in the garden on Tuesday] though he did...

...they were disappointed not to have received toys (in the library) (on Monday).

The fronted strings in (29b-e) all correspond to a constituent of the ascending structure in (24): *give the children* ($V^{1^{SO}}$), *give the children the books (about North American beavers)* ($V^{2^{SO}}$), *give the children the books (about North American beavers) in the garden* ($V^{3^{SO}}$), *give the children the books (about North American beavers) in the garden on Tuesday* ($V^{4^{SO}}$). Therefore, the ascending structure analysis provides a clear and simple explanation of fronting around *though* patterns.

The descending structure analysis correctly predict that the string *give the children the books (about North American beavers) in the garden on Tuesday* should be able to front around though. This is because the string corresponds to the entire descending structure ($v^{2^{SO}}$). However, none of the other three licit examples involve fronting of a string that corresponds to a constituent of the descending verb phrase in (25).

In addition to incorrectly predicting that (29b-d) should be unacceptable, the descending structure analysis incorrectly predicts that the example in (30) should be acceptable because the fronted string *the children the books (about North American beavers) in the garden on a Tuesday* is a constituent in the descending structure—that is, it corresponds to $V^{4^{SO}}$.

(30) **But [the children the books (about North American beavers) in the garden on Tuesday] though he did give...[he wanted to lend them toys in the library on Monday].

In contrast to the descending structure analysis, the ascending structure analysis correctly predicts that (30) should be unacceptable because the string *the children the books (about*

North American beavers) in the garden on Tuesday doesn't correspond to a constituent in the ascending structure.

In (31), you can see the paradigm for verb phrase ellipsis in double object constructions. The paradigm is in parallel with the prepositional dative paradigm. It is impossible for verb phrase ellipsis to only target the verb (31a). This is expected under either of the structural analyses we are considering because I assume that verb phrase ellipsis is restricted to applying to phrasal syntactic objects, and the verb on its own is not a phrasal syntactic object. It's possible, though, to elide any other verb initial string.

(31) John gave the children the books (about North American beavers) in the garden on a Tuesday...

a. ...**long before Mary did ⟨give⟩ the seniors the DVDs (about North American beavers) in the library on a Monday.

b. ...long before Mary did ⟨give the children⟩ the DVDs (about North American beavers) in the library on a Monday.

c. ...long before Mary did ⟨give the children the books (about North American beavers)⟩ in the library on a Monday.

d. ...long before Mary did ⟨give the children the books (about North American beavers) in the garden⟩ on a Monday.

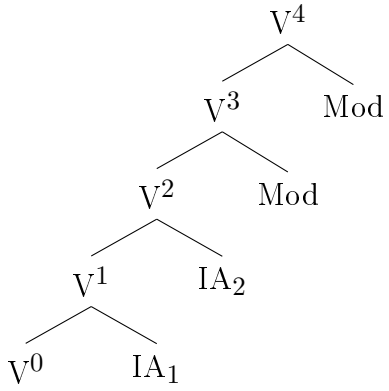
e. ...long before Mary did ⟨give the children the books (about North American beavers) in the garden on a Tuesday⟩.

The ascending structure analysis makes the correct prediction for all of the examples in (31) because every verb initial string (except for the string consisting of just the verb) corresponds to a phrasal syntactic object in the ascending structure. The descending structure analysis makes the correct prediction with respect to (31e) because the string *give the children the books (about North American beavers) in the garden on a Tuesday* corresponds to the entire

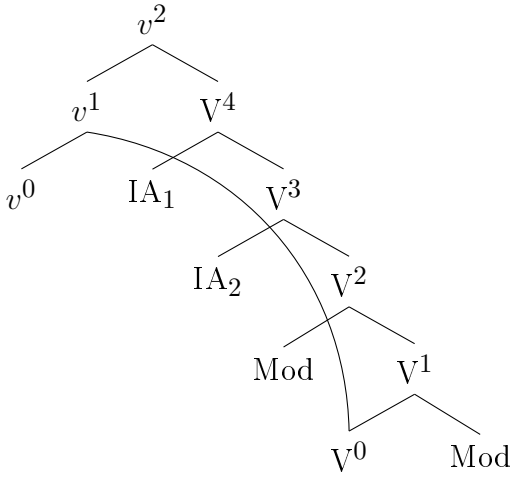
descending structure (i.e $v^{2^{SO}}$), but this analysis incorrectly predicts that examples (31b-d) should be unacceptable.

In this section, I've provided evidence from both patterns of syntactic movement and ellipsis that supports the an analysis of both double object and prepositional dative verb phrases where the structure is ascending, as in (32). Furthermore, I have shown that the fronting and ellipsis data are incompatible with the descending constituent analyses (33) of the two kinds of ditransitive verb phrases.

(32) *Ascending ditransitive*



(33) *Descending Ditransitive*



In the next section, I will present evidence from (or adapted from) Pesetsky 1995 that a single expression can simultaneously have an ascending structure and a descending structure. These examples form the empirical basis for what has become known as *Pesetsky paradoxes*. There have been several proposed solutions to Pesetsky paradoxes. I will briefly review some of the solutions to the paradoxes. Then, I will discuss some consequences the paradoxes have on ditransitive mismatch (in particular) and the correspondence between syntactic and prosodic representations (in general).

4.1.4 Pesetsky paradoxes

In §4.1.2-4.1.3, I have presented both arguments for and against the descending structure and arguments for and against the ascending structure of ditransitive verb phrases. On the basis of the data in those sections, it is a logical possibility that a single string corresponds to a descending structure when we are observing asymmetric binding effects and that the same string corresponds to an ascending structure when we observe partial verb phrase fronting/ellipsis. In this section, I will present data that rules out this logical possibility. In these data, we can simultaneously observe both asymmetric binding/licensing effects and partial fronting/ellipsis effects. The simultaneous occurrence of conflicting constituency evidence is known as a Pesetsky paradox.

In (34), we reveal that partial fronting around *though* can co-occur with reciprocal binding. And crucially, the binder is in the fronted string while the reciprocal isn't.

(34) a. But [show [the children]$_i$] though he did to [each other]$_i$'s mothers in the garden on Tuesday...

b. But [show [the children]$_i$] though he did to the mothers in the garden on [each other]$_i$'s birthdays...

c. But [show the child to [the mothers]$_i$] though he did in the garden on [each other]$_i$'s birthdays...

d. But [show [the children]$_i$ to the mothers in the garden] though he did on [each other]$_i$'s birthdays...

The linearly first internal argument (i.e. IA_1) is fronted around *though* with the verb in (34a,b), stranding IA_2 and the verb phrase modifiers. Under the assumption that only constituents can front around *though*, it follows that the string *show the children* is a constituent to the exclusion of IA_2 (and the modifiers). If that were the case, then it should be impossible for IA_1 to c-command the reciprocal in IA_2; as a result, we predict that the plural deter-

miner phrase *the children* (IA_1) can't bind the reciprocal in IA_2. However, it apparently can. So there is a clear conflict in a single expression between the structural requirements necessary to account for patterns of movement and binding. This is a Pesetsky paradox. The remaining examples in (34) are versions of the same paradox: The ascending structure appears to be needed in order to account for the partial fronting, but the reciprocal binding calls for a descending structure.

Paradoxical requirements also arise when we look at the interaction between reciprocal binding and verb phrase ellipsis (35).

(35) John showed the children to the mothers in the garden on Labor Day...

a. ...long before Mary did ⟨show [the children]$_i$⟩ to [each other]$_i$'s fathers in the library on May Day.

b. ...long before Mary did ⟨show [the children]$_i$⟩ to the fathers in the library on [each other]$_i$'s birthdays.

c. ...long before Mary did ⟨show [the children]$_i$ to the mothers⟩ in the library on [each other]$_i$'s birthdays.

d. ...long before Mary did ⟨show [the children]$_i$ to the mothers in the garden⟩ on [each other]$_i$'s birthdays.

If verb phrase ellipsis only targets syntactic constituents, then nothing in an ellipsis site should c-command anything outside the ellipsis site. And if reciprocal binding requires a plural binder to c-command its reciprocal associate, then nothing in a ellipsis site should be able to bind a reciprocal outside an ellipsis site. However, in each of the examples in (35), a plural determiner phrase in an ellipsis site binds a reciprocal outside the ellipsis site.

The examples in (34) and (37) demonstrate that the same exact paradoxes arise for the double object construction. Requirements for the patterns of syntactic fronting (36) and ellipsis (37) conflict with the requirements for reciprocal binding.

(36) a. But [give [the children]$_i$] though he did [each other]$_i$'s books (about North American beavers) in the garden on Tuesday...

b. But [give [the children]$_i$] though he did the books *(about North American beavers) in the garden on [each other]$_i$'s birthdays...

c. But [give [the children]$_i$ the books] though he did in the garden on [each other]$_i$'s birthdays...

d. But [give [the children]$_i$ the books in the garden] though he did on [each other]$_i$'s birthdays...

e. But [give [the children]$_i$ the books in the garden on [each other]$_i$'s birthdays] though he did...

...they were disappointed not to have received toys (in the library) (on Monday).

(37) John gave the children the books (about North American beavers) in the garden on Labor Day...

a. ...long before Mary did ⟨give [the children]$_i$⟩ the DVDs (about North American beavers) in the library on [each other]$_i$'s birthdays.

b. ...long before Mary did ⟨give [the children]$_i$ the books (about North American beavers)⟩ in the library on [each other]$_i$'s birthdays.

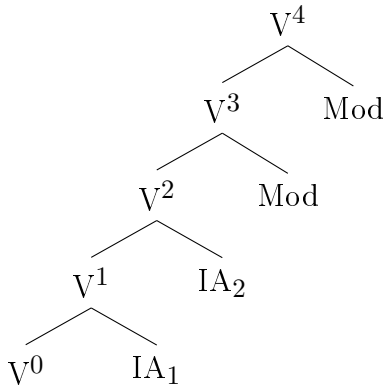
c. ...long before Mary did ⟨give [the children]$_i$ the books (about North American beavers) in the garden⟩ on [each other]$_i$'s birthdays.

d. ...long before Mary did ⟨give [the children]$_i$ the books (about North American beavers) in the garden on [each other]$_i$'s birthdays⟩.

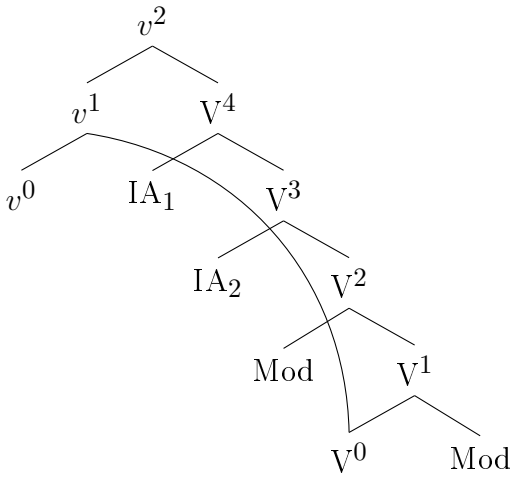
What solutions to Pesetsky paradoxes have been proposed? Pesetsky (1995) proposed that all linguistic expressions correspond simultaneously to two syntactic representations, which he calls cascade (our descending) and layered (our ascending) structure. That is, he

proposed that all ditransitive verb phrases should be analyzed as have both something like[17] the ascending and descending structure in (38) and (39).

(38) *Ascending ditransitive*



(39) *Descending Ditransitive*



In Pesetsky, binding, licensing, and coordination mechanisms are evaluated over the descending structure, while movement and ellipsis are evaluated over the ascending structure. Many may be skeptical of the idea that an expression can be assigned two parallel representations. Pesetsky (1995, p.248) himself wrote: 'Obviously a theory with one structural representation is simpler than a theory with two. The claim here is not that [the two parallel representation theory] is a null hypothesis for syntax, but that it is a *correct* hypothesis for syntax'. I'm not willing to debate the claim that a theory with one representation is simpler than a theory with two, because I have no clear framework for evaluating the simplicity of syntactic arguments. However, I don't necessarily understand why syntacticians generally assume that there can be only one syntactic representation per interpretation of each linguistic expression. As Jorge Hankamer put it,

17. Pesetsky's structures aren't identical to the structures I've been considering. However, the differences don't really matter to the topic of this chapter or thesis.

> [i]t has been a fundamental and, so far as I know, unquestioned methodological assumption in the theory and practice of syntactic investigation that for a given body of data there is one correct analysis[...]and our job is to find that analysis[...This assumption] is actually a very strong claim about the nature of this knowledge, and about the properties of the learning mechanism that allow a speaker to acquire it. Linguists have long marvelled at the apparent ability of a child to acquire the grammar of his language in so short a time on the basis of such incomplete and confusing data. I suggest he does not have such a putative ability at all. In the face of confusing and conflicting data, I believe he constructs conflicting analyses, and does not necessarily ever choose between them (Hankamer, 1977, pp.583-584).

A more recent strand of research done by Bruening, Janke and Neeleman, and others has challenged the theory of binding underlying the arguments in Barss and Lasnik 1986; Larson 1988, 1990. These researchers present evidence that, in many cases, c-command can't be the operative relation for binding—that is, it is far too strict. Bruening 2014 presents empirical arguments that c-command is far too strict to account for the full array of Condition C effects, and as a result, it is clear that Condition C effects can't be used to argue for a particular structural asymmetry between the two internal arguments of a ditransitive verb phrase. Hoeksema 2000 presents several examples from both English and Dutch that pose challenges to a theory which says that a licensor must c-command the a negative polarity item it licenses. This kind of evidence has led most of these researchers to conclude that a theory of binding based on linear precedence and some weaker structural relation goes further toward a unified account of binding/licensing phenomena. For Bruening (2001, 2014, 2018), the relevant structural relation is Phase-command, and for Janke and Neeleman (2012), it's m-command. In these theories, there are no conflicting requirements because the command domain for a binder/licensor is much larger (e.g. any phase-mate to the right), and as

a result, an ascending verb phrase structure is compatible with patterns of binding and licensing.

A third approach to solving Pesetsky paradoxes is proposed in Phillips 1996, 2003. In this work, Phillips proposes a theory of syntax where the representation of a linguistic expression is built string linearly from left to right. Most instances of structure-building operations in this theory are non-extending. Interestingly, he shows that building structure in this way provides a clear solution to Pesetsky paradoxes. Intermediate representations look something like the front-able/elide-able objects that we observe in §4.1.3, but as the derivation proceeds, these intermediate representations are destroyed, resulting in final representations that resemble the descending structure in (39). Phillips's approach is different from the other two approaches because there is no level of representation or intermediate representation that has the essential properties of the ascending structure in (38).

In any case, it seems reasonable to assume that ditransitive verb phrases have an ascending structure at some level of representation—that is, either in conjunction with a descending structure (Pesetsky, 1995) or not (Bruening, 2001, 2014, 2018; Janke and Neeleman, 2012). This, of course, has consequences for ditransitive mismatch. Recall that I argued in §4.1.1 that the existence of ditransitive mismatch as a real theoretical puzzle depends on the assumption that all ditransitive verb phrases have (i) one and only one representation and (ii) that that representation is the descending structure. If there is some level of representation at which ditransitive verb phrases have an ascending structure, which we have justified with several empirical arguments, then (ii) certainly doesn't hold and (i) may not hold.

In §4.2.1, I walk through (Erik Zyman's) English intonational phrasings of ditransitive verb phrases and show how the paradigms can be easily explained if the prosodic representation is mapped from an ascending syntactic structure. In other words, I will show that there is an isomorphism between prosodic representations and the syntactic representations that they correspond to (i.e. there is no ditransitive mismatch). In §4.2.3, I argue on the basis

of interactions between intonational phrasing and epithets that the mapping from syntactic representations to prosodic representation must be (i) derivational and (ii) non-strictly extending.

4.2 Ditransitives, epithets, and prosodic derivations

The puzzle of ditransitive mismatch—that is, the purported non-isomorphism between the syntactic representations of ditransitive verb phrase and their corresponding prosodic representations—is only, in fact, a puzzle if the posited syntactic and prosodic representation are supported by valid empirical arguments. As I note in §4.1.1, those who put forth the puzzle of ditransitive mismatch posit a descending (or right-branching) ditransitive verb phrase structure. The last few sections were devoted to reviewing evidence for two potential syntactic structures of ditransitive verb phrases. While there is some empirical support for the descending verb phrase (§4.1.2), we saw that there is also a preponderance of evidence (§4.1.3-4.1.4) in support of an alternative analysis: Ditransitive verb phrases have an ascending structure (at least at some level of representation). Furthermore, those who have presented the puzzle of ditransitive mismatch posit an ascending prosodic representation that is (essentially) isomorphic with the ascending syntactic structure. Therefore, if there is empirical evidence in support of the purported ascending prosodic structure, then the puzzle of ditransitive mismatch isn't, in fact, a puzzle. In this section, I investigate the prosodic structure of ditransitive verb phrases.

In §4.2.1, I argue that intermediate phrase domains are organized in such a way that supports the ascending prosodic representation. Given that prosodic representations are mapped from ascending syntactic structures, it follows that there is an isomorphism between two kinds of representations—that is, there is no puzzle of ditransitive mismatch. Furthermore, in §4.2.2 I review some cases where the intermediate phrasing is affected by (i) a special kind of narrow focus on ditransitive verbs and (ii) the GIVEN-ness of material

in the ditransitive verb phrase. I present evidence that the internal arguments in a ditransitive verb phrase can, in fact, be phrased together to the exclusion of the verb, but the phrasing only arises as a possibility when conflict(s) arise between independently supported prosodic requirements. In §4.2.3, I present evidence (from the interaction between epithets and intonation) that supports the claim that prosodic representations are built derivationally. Crucially, the argument relies on patterns of (i) ineffability and (ii) markedness to show that intermediate prosodic representations are (a) evaluated for well-formedness and (b) sometimes not extended.

4.2.1 Prosodic structure of English ditransitives

In this section, I will argue that the possible intermediate phrase domains of ditransitive verb phrases without any GIVEN material in (Erik Zyman's) English[18] follow from two assumptions. One, the prosodic representations of ditransitive verb phrases are mapped from ascending syntactic representations of those verb phrases. And two, intermediate phrases are always mapped from syntactic constituents. It follows from these two assumptions that a (sub)string of a ditransitive verb phrase can only form an intermediate phrase domain if the (sub)string correlates with a constituent in the ascending verb phrase structure.

Intermediate phrases are characterized by an intonational contour composed of a nuclear pitch accent (T*), a phrase accent (T_), and any number of prenuclear pitch accents (∼T).[19] The nuclear pitch accent is the final pitch accent in an intermediate phrase, and prenuclear pitch accents are non-final pitch accents within an intermediate phrase; nuclear pitch accents are never less prominent (i.e. a less extreme pitch excursion) than any of the prenuclear

18. When it comes to examples set in a wide focus, Erik Zyman's judgements are perfectly in line with the judgements reported in Selkirk 1984 and Selkirk 2000. I will also present data where a prosodic requirement on word emphasis interacts with a prosodic requirement on GIVEN material. In some of those cases, Erik's judgments differ from those presented in Selkirk 2000.

19. It has also been claimed that intermediate phrases are the domain for *nuclear stress* assignment (Ladd, 2008, pp.299-302).

pitch accents. Lisa Selkirk has referred to intermediate phrases by a different name in her work: *major phrases* (MaP) in Selkirk 2000 and *phonological phrases* (φ) in Kratzer and Selkirk 2020. However, I will continue to refer to the prosodic domain as the intermediate phrase, following the convention of those who work in the Autosegmental-Metrical framework (Pierrehumbert, 1980; Ladd, 2008; Ahn, 2015).

Selkirk (2000) notes that in an information-structural context where neither the verb phrase nor anything within it are GIVEN (as defined in Schwarzschild 1999), it is natural and neutral to produce a single phrase accent that falls somewhere near the right edge of the verb phrase, as in (40).

(40) *Ali:* Tell me what Molly wrote about in her journal. (Selkirk, 2000, p.241)

a. ∼H ∼H H* **L**_
(She loaned her rollerblades to Robin)$_{\text{iP}}$.

b. ∼H ∼H ∼H H* **L**_
(She pushed Sam's boat into the water)$_{\text{iP}}$.

c. ∼H ∼H H* **L**_
(She gave Zoe a backrub)$_{\text{iP}}$.

d. ∼H ∼H H* **L**_
(She sent her sincere regrets to Luis)$_{\text{iP}}$.

In (40), Ali asks someone to report something from a journal, and the speaker says something(s) about what Molly has done. As suggested by the parentheses and subscript iP (for intermediate phrase), the fact that there is only one phrase accent (by definition) means that the entire sentence forms a single intermediate phrase domain. Note that this kind of intonational contour (and prosodic phrasing) is possible with both double object constructions (40c) and prepositional dative constructions (40a,d). I've included (40b) because Selkirk (2000) does and to show that the patterns hold for other kinds of ditransitive constructions with prepositional phrase arguments.

In the same information-structural context as above, the speaker could respond with an

additional phrase accent near the right edge of the linearly first internal argument (IA_1). Selkirk's examples are in (41).

(41) *Ali:* Tell me what Molly wrote about in her journal. (Selkirk, 2000, p.242)

a. ∼H H* **L**_ H*L_
(She loaned her rollerblades)$_{iP}$ (to Robin)$_{iP}$.

b. ∼H ∼H H*L_ H* L_
(She pushed Sam's boat)$_{iP}$ (into the water)$_{iP}$.

c. ∼H H*L_ H* L_
(She gave Zoe)$_{iP}$ (a backrub)$_{iP}$.

d. ∼H H*L_ H* L_
(She sent her sincere regrets)$_{iP}$ (to Luis)$_{iP}$.

Importantly, this additional phrase accent doesn't add anything about the speaker's attitude toward Molly's actions—it is interpreted as a neutral report. It follows from the fact that there are two phrase accents that there are also two intermediate phrases: one intermediate phrase is composed of the verb and one internal argument (IA_1), and the other intermediate phrase is composed of the linearly second internal argument (IA_2). And again, this intonational phrasing is possible in both double object and prepositional dative constructions.

When a third phrase accent aligns with the right edge of the verb, examples are infelicitous in the same information-structural context (42).

(42) *Ali:* Tell me what Molly wrote about in her journal. (Selkirk, 2000, p.243)

a. # H*L_ H* L_ H*L_
(She loaned)$_{iP}$ (her rollerblades)$_{iP}$ (to Robin)$_{iP}$.

b. # H*L_ ∼H H*L_ H*L_
(She pushed)$_{iP}$ (Sam's boat)$_{iP}$ (into the water)$_{iP}$.

c. # H*L_ H*L_ H* L_
(She gave)$_{iP}$ (Zoe)$_{iP}$ (a backrub)$_{iP}$.

d. # H*L_ H*L_ H*L_
(She sent)$_{iP}$ (her sincere regrets)$_{iP}$ (to Luis)$_{iP}$.

However, the examples are felicitous if the the verb is emphasized (represented by italics in (43) and any subsequent examples). This emphasis is characterized by a pragmatic force (i.e. surprise, resentment, suspicion, etc.), as well as a prosodic prominence, which will most likely be realized phonetically as a more extreme pitch event—a larger pitch excursion, a stronger break, and/or a longer duration.[20] You might imagine someone uttering (43a) if they were shocked that Molly would loan anything to anyone, because the speaker has the impression that Molly doesn't trust people to return her items to her.[21]

(43) *Ali:* Tell me what Molly wrote about in her journal. (Selkirk, 2000, p.247)

a. H*L− H* L− H*L−
(She *loaned*)$_{iP}$ (her rollerblades)$_{iP}$ (to Robin)$_{iP}$.

b. H* L− ∼H H*L− H*L−
(She *pushed*)$_{iP}$ (Sam's boat)$_{iP}$ (into the water)$_{iP}$.

c. H*L− H*L− H* L−
(She *gave*)$_{iP}$ (Zoe)$_{iP}$ (a backrub)$_{iP}$.

d. H*L− H*L− H*L−
(She *sent*)$_{iP}$ (her sincere regrets)$_{iP}$ (to Luis)$_{iP}$.

As we proceed though the empirical possible intermediate phrasings, it will become very clear that emphasized prosodic words must be at the right edge of an intermediate phrase domain; we will see no exceptions to this requirement.[22] When there are three phrase accents, as is

20. This notion of prosodic *emphasis* is roughly the same as Ladd's 2008, pp.255-6. Selkirk (2000, pp.246-7) uses the term *Focus* (FOC) when describing the same phenomenon in these data. I prefer Ladd's term, *emphasis*.

21. Several researchers have noted that focus must occur at the right edge of a prosodic domain (Kanerva, 1989; Truckenbrodt, 1995, 1999; Selkirk, 2000). It's difficult to say whether or not the type of focus they observe is the same as the emphasis we see in (43). For one, in the works cited, the authors don't normally set their examples in any discourse context, so it may be that the distribution of focus and emphasis is quite different; for example, it's not clear from the context in (43) that the emphasized verb is contrastively focused, and it could be that the examples in the literature are all cases of focus contrasting with something in the linguistic context. Second, the right-edge requirement may be stronger with emphasis than with (at least some of) the focus construction in the works cited; in Selkirk 2000, the right-edge requirement is a violable constraint that is dominated by several other constraints, and as a result, the constraint is often violated.

22. You may wonder if the framing in the body text is inaccurate. Instead, it may be that emphasized

the case in (43), there are (by definition) three intermediate phrase domains. You can see in (43c) that it's possible to prosodically phrase double object constructions with three phrase accents, where the verb, IA_1, and IA_2 each form their own intermediate phrase domains. The same is true for prepositional dative constructions, which you can see in the other three examples.

It's possible to emphasize anything in this way. For example, in the same context, one could respond by emphasizing *Robin* (44a) or *rollerblades* (44b). Note that the context for these examples is unchanged; it's exactly the same as any of the examples we've seen so far.

(44) *Ali:* Tell me what Molly wrote about in her journal. (Selkirk, 2000, p.247)

a. (She loaned her rollerblades to *Robin*)$_{\text{iP}}$.
   ∼H ∼H H*L_

b. (She loaned her *rollerblades*)$_{\text{iP}}$ (to Robin)$_{\text{iP}}$.
   ∼H H* L_ H*L_

The emphasized words in (44) have an extra pragmatic force and phonetic prominence, but nothing in the verb phrase is GIVEN.

Because emphasized words must occur at the right edge of an intermediate phrase domain, emphasis allows us to construct examples where an intermediate phrase boundary must separate the verb from the linearly first internal argument (IA_1). As a result, emphasizing the verb allows us to test if the two internal arguments can phrase together to the exclusion of the verb. As it turns out, simply emphasizing the verb does not create the conditions for phrasing the two internal arguments together, as you can see in (45b), where the information

material must bear a nuclear pitch accent. It follows, then, that emphasized material will be at the right edge of an intermediate phrase domain if the immediately following linguistic material can't be deaccented for information-structural reasons. However, if everything following the emphasized material is deaccented, it could be that the emphasized material is not at the right edge of the intermediate phrase domain. As we will see shortly, phrase accents shift leftward if the nuclear pitch accent doesn't occur at the right edge of the intermediate phrase domain, so it is difficult to determine (without special instrumentation and software) if emphasized material is always at the right edge of the domain or if it always bears nuclear pitch accent. However, see §4.2.2 for evidence that the requirement is actually that emphasis must occur at the right edge of an intermediate phrase.

structure of the examples is the same as we've seen so far and the verb *loaned* has been emphasized.

(45) *Ali:* Tell me what Molly wrote about in her journal. (Selkirk, 2000, p.247)

H* **L**_ H* **L**_ H***L**_

a. (She *loaned*)$_{\text{iP}}$ (her rollerblades)$_{\text{iP}}$ (to Robin)$_{\text{iP}}$.

H* **L**_ ∼H H***L**_

b. * (She *loaned*)$_{\text{iP}}$ (her rollerblades to Robin)$_{\text{iP}}$.

As we saw before, when the verb is emphasized, it's perfectly acceptable to phrase each of the verb, IA_1, and IA_2 into separate intermediate phrase (45a). However, as Selkirk (1984, 2000) reported, example (45b) shows that it isn't acceptable to phrase the two internal arguments together to the exclusion of the verb.

So far in this section, I have run through two kinds of sentences. First, we looked at intermediate phrasing possibilities for ditransitive construction in broad focus and without emphasis. These sentences had two possible intermediate phrasings—that is, those in (46a,b). Second, we looked considered examples where the verb bears emphasis, and what we observed is that (i) the verb must occur at the right edge of an intermediate phrase[23] and (ii) emphasizing the verb requires each of the internal arguments to occur at the right edge of intermediate phrases (46d), and (iii) it isn't possible to phrase the two internal arguments together to the exclusion of the verb (46c).

(46) a. $\ldots \text{V IA}_1 \text{ IA}_2)_{\text{iP}}$ (40)

b. $\ldots \text{V IA}_1)_{\text{iP}}$ $(\text{IA}_2)_{\text{iP}}$ (41)

c. $*\ldots \text{V})_{\text{iP}}$ $(\text{IA}_1 \text{ IA}_2)_{\text{iP}}$ (45b)

d. $\ldots \text{V})_{\text{iP}}$ $(\text{IA}_1)_{\text{iP}}$ $(\text{IA}_2)_{\text{IP}}$ (43) & (45a)

In short, Erik Zyman's English appears to comply with the previously observed Ditransitive Phrasing Generalization (repeated in (47)).

23. We also observed in (42) that verbs apparently can't be at the right edge of intermediate phrases when they aren't emphasized.

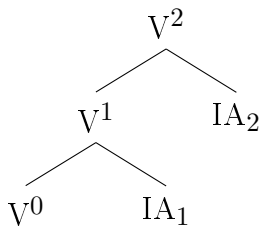
(47) **Ditransitive Phrasing Generalization (repeated)**

The two internal arguments of a ditransitive verb phrase cannot form a prosodic domain to the exclusion of their selecting verb: *(V)(IA_1 IA_2).

With respect to intermediate phrasing in English, the Ditransitive Phrasing Generalization captures the apparent fact that the two internal arguments (i.e. IA_1 and IA_2) can't form an intermediate phrase to the exclusion of the verb.

Given that I concluded in §4.1.4 that it is reasonable to assume that ditransitive verb phrases have an ascending structure,[24] like the structure in (48), an account of the Ditransitive Phrasing Generalization can be simple.

(48) *Ascending ditransitive (without adjuncts)*



If we assume that the ascending structure is the relevant syntactic structure for the purposes of determining a string's prosodic representation, then the Ditransitive Phrasing Generalization follows from the Constituent Mapping Conjecture, which I've repeated below in (49) from §4.1.1.

(49) **Constituent Mapping Conjecture (repeated)**

A string w is a possible prosodic domain (e.g. an intermediate phrase domain) iff w is analyzable as a syntactic constituent.

According to the Constituent Mapping Conjecture, the verb and two internal internal arguments should be able to form an intermediate phrase domain because the string correlates to

24. Either exclusively or in parallel with a descending structure.

a constituent in the ascending verb phrase structure (i.e $V^{2^{SO}}$). And we know from (40) that this prosodic domain is empirically attested. It also follows that the verb and first internal argument (IA_1) should be able to form an intermediate phrase to the exclusion of the second internal argument (IA_2) because the string V-IA_1 correlates with the syntactic constituent $V^{1^{SO}}$ (and IA_2 correlates with syntactic constituent IA_2). Importantly, it's predicted that the string IA_1-IA_2 shouldn't be able to form an intermediate phrase because that string doesn't correlate with any syntactic constituent in the ascending verb phrase structure. In other words, the Constituent Mapping Conjecture together with the ascending verb phrase syntactic analysis[25] capture the Ditransitive Phrasing Generalization in (49).

One fact that doesn't follow from the analysis as it stands is that the verb must be emphasized in order to occur at the right edge of an intermediate phrase—compare the unacceptable (42) and the acceptable (43). Because the verb is a constituent in the ascending verb phrase structure, it follows from the Constituent Mapping Conjecture that the verb on its own should able form an intermediate phrase. And although it can under certain conditions, it would be ideal to capture those conditions in the analysis. I know there are many avenues to take, but I'll elect to take the following tack: The Constituent Mapping Conjecture is defined such that prosodic domains must be mapped from phrasal syntactic

25. As Alan Yu (p.c.) noted, if a ditransitive verb phrase has two structures (namely, ascending and descending), it's not obvious why only one of the syntactic structures should be the input to mappings to prosodic representation. If it is the case that there are two structures, we might expect that the grammar of some speakers involve either (i) mappings from descending ditransitive structures to prosodic structure or (ii) mappings from both ascending and descending structures to a prosodic structure. Whether or not these grammars exist is an empirical question, and as far as I know, there is no empirical evidence in support of these grammars. If mappings of the kind in (i) and (ii) are impossible, then I see two possible conclusions to draw from the fact. One, it may be the case that there is only one syntactic representation of ditransitive verb phrase (i.e. the ascending one). The arguments in support of the descending structure must be reanalyzed in some way, and they have by Bruening (2001, 2014, 2018); Janke and Neeleman (2012). Another possible conclusion is that the ascending structure supersedes or outweighs the descending structure. The evidence for the ascending structure is more directly tied to syntactic constituents as a unit—that is, they are arguments about the properties of syntactic objects. The arguments for the descending structure, on the other hand, are about relations between syntactic objects. It could be argued that prosodic phrases are more like properties of strings and less like relations between strings; therefore, maybe we might expect a closer link between ascending structures and prosodic representation than we do between descending structures and prosodic representation.

object (as defined in fn.12):

(50) **Constituent Mapping Conjecture (Phrasal Version)**

A string w is a possible prosodic domain (e.g. an intermediate phrase domain) iff w is analyzable as a **phrasal** syntactic constituent.

The above conjecture predicts that it should be impossible for non-phrasal syntactic objects to map to prosodic phrases, like intermediate phrases. However, as we saw in (43) & (45), it's possible for a string w that correlates with a non-phrasal syntactic object can form a prosodic domain if w is emphasized. In §4.2.2, I will provide an account of this and other exceptions to the Constituent Mapping Conjecture

In this section, I have reviewed the data (and confirmed it with a new consultant) that supports the Ditransitive Phrasing Generalization for English intermediate phrasing. I then showed that the generalization follows nicely from (i) the assumption that prosodic representations of ditransitive verb phrases are mapped from ascending syntactic structures and (ii) the Constituent Mapping Conjecture (i.e. the relevant syntactic and prosodic representations of a string are isomorphic). In other words, I argued that there is no puzzle of ditransitive mismatch because there is no mismatch at all.

4.2.2 The effects of deaccenting

In this section, I briefly review the interaction between GIVEN information and prosody in English. We'll see that GIVEN information correlates with the phenomenon of deaccenting. We will then look at the interaction between emphasis and deaccenting. This interaction can lead to conflicts between prosodic requirements, and interestingly, there are two strategies to resolving the conflict. Which strategy is employed depends on which internal argument is GIVEN and which isn't. I argue that the distribution of resolution strategies provides evidence for intermediate prosodic representations and therefore prosodic derivations.

I assume, following Bolinger 1977; Rooth 1992; Schwarzschild 1999; Büring 2003, 2016; Ladd 2008; Arregi 2016, that information structure organizes an utterance into parts which do and do not have an antecedent in the prior discourse. When a piece of an utterance has an antecedent, we will say it is GIVEN, and when it does not have an antecedent, we will say it is not GIVEN (or *new*). In (44), we saw examples where the verb is GIVEN but it is followed by material that isn't GIVEN. In this situation is it is possible to associate a prenuclear accent with the GIVEN material. However, when GIVEN material is only followed by other given material or is sentence final, then it is deaccented.[26] We can interpret this pattern as a ban on associating a nuclear pitch accent with GIVEN material.

If the information-structural context is such that some elements of the utterance are GIVEN, those elements will not be able to licitly bear a pitch accent—they are *deaccented*. For example, Ali's utterance in (51) antecedes *Molly/she X-ed her rollerblades to Y* in Dani's response, where the elements that have no antecedent, and therefore, aren't GIVEN (in the response) are the verb *loaned* and IA_2 *Tony*. Because *loaned* and *Tony* are new elements in Dani's response to Ali, only *loaned* and *Tony* can bear a pitch accent (51b). The first internal argument IA_1 must be deaccented, as in (51b).[27] If IA_1 bear an accent, as in (51a), the utterance is infelicitous.

(51) *Ali:* I heard Molly gave her rollerblades to Robin.

Dani:

a. # (No,) (She loaned)$_{iP}$ (her rollerblades)$_{iP}$ (to Tony)$_{iP}$.
H*L_ H* L_ H*L_ L%

b. (No,) (She loaned her rollerblades)$_{iP}$ (to Tony)$_{iP}$.
H* L_ H*L_ L%

26. It is possible for material to be GIVEN and in focus. In those cases, the GIVEN material can (and typically, must) bear a nuclear pitch accent. We won't look at any examples of this kind in the chapter, so going forward, when I say something GIVEN, I mean GIVEN and not in focus.

27. As always in this chapter, I'm ignoring the subject.

Dani: (No,) She loaned her rollerblades to Tony. [equivalent to (51b)]

There is another interesting phenomenon that can be seen in (51). In the felicitous example (51b), the phrase accent associated with the first intermediate phrase occurs in a different position than we have seen in previous examples. The phrase accent shifts left to a position close to the peak of the nuclear pitch accent (Pierrehumbert, 1980; Beckman and Pierrehumbert, 1986; Barnes et al., 2010). There is a sharp drop from the peak to the the low point of the contour.

Recall from (43) & (45) that emphasized material must occur at the right edge of an intermediate phrase. If a constituent is emphasized, and hence, separated from following deaccented material, the result is unacceptable in any context (hence, the * in (52)). But a GIVEN constituent can be phrased as an intermediate phrase, and thus bear a nuclear pitch accent, if it resolves a phrasing issue, as in (53).

(52) *Ali:* I hear Molly gave her rollerblades to Robin.

H* **L**_ H*__L__ L%

a. * (No,) (She *loaned*)$_{\text{iP}}$ (her rollerblades to Tony)$_{\text{iP}}$.

b. * (No,) She loaned her rollerblades to Tony.

(53) *Ali:* I hear Molly gave her rollerblades to Robin.

H* **L**_ H* **L**_ H*__L__ L%

a. (No,) (She *loaned*)$_{\text{iP}}$ (her rollerblades)$_{\text{iP}}$ (to Tony)$_{\text{iP}}$.

b. (No,) She loaned her rollerblades to Tony.

Note that the only difference between the example (51) and examples (52) & (53) is that the verb is emphasized in the latter examples. Example (53), in particular, reveals that the

requirement that emphasized material occur at the right edge of an intermediate phrase is stronger that the requirement that GIVEN material not bear a nuclear pitch accent.[28] If emphasized material didn't absolutely have to occur at the right edge of an intermediate phrase, then we would predict that the following GIVEN material would be phrased into the same intermediate phrase as the emphasized verb, as is the case when the verb isn't emphasized (51b). However, as you can see in (54), this isn't a possible phrasing.

(54) *Ali:* I hear Molly gave her rollerblades to Robin.

H*L_ H*L_ L%
* (No,) (She *loaned* her rollerblades)$_{iP}$ (to Tony.)$_{iP}$

The availability of the repair in (53) demonstrates the robustness of the Ditransitive Phrasing Generalization. Generally speaking, (non-focused) anaphoric material can't bear nuclear pitch accent, so it's interesting that this requirement (or constraint) is trumped by whatever is responsible for the Ditransitive Phrasing Generalization. Interestingly, counterexamples to the Ditransitive Phrasing Generalization arise when the following holds: (i) the verb is emphasized, (ii) IA_1 isn't GIVEN, and (iii) IA_2 is GIVEN. This is the state-of-affairs in (55) and (56).

(55) *Ali:* I hear Molly gave her bike to Tony.

a. H* L_ H* L_ L%
(No,) (She *loaned*)$_{iP}$ (her rollerblades to Tony)$_{iP}$.

b. (No,) She loaned her rollerblades to Tony.

28. The relative strength of these prosodic requirements must be true for Erik Zyman's English. However, although her examples don't have the same information structure as the examples I present, Selkirk (2000, ex.36-37) concludes the opposite—that is, the requirement that emphasized material can occur at the right of an iP can be violated in order to comply with the deaccenting requirement on GIVEN material.

(56) *Ali:* I hear Molly gave her bike to Tony.

H* **L**_ H* **L**_ H* **L**_ L%

a. * (No,) (She *loaned*)$_{\text{iP}}$ (her rollerblades)$_{\text{iP}}$ (to Tony)$_{\text{iP}}$.

b. * (No,) She loaned her rollerblades to Tony.

When IA_2 is GIVEN and the verb is emphasized, then it's acceptable to phrase IA_1 and IA_2 together to the exclusion of the verb: We find a counterexample (55) to the Ditransitive Phrasing Generalization, which I've repeated in (57).

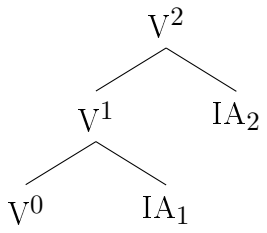
(57) **Ditransitive Phrasing Generalization (repeated)**

The two internal arguments of a ditransitive verb phrase cannot form a prosodic domain to the exclusion of their selecting verb: *(V)(IA_1 IA_2).

This is a surprising result for several reasons. From an empirical perspective, it's surprising because counterexamples to the Ditransitive Phrasing Generalization have gone almost entirely unreported in the literature; the only exception I know of is in Steedman (1991, p.288). From an analytical perspective, it's surprising because (i) it doesn't follow from the analysis I proposed at the end of §4.2.1, and (ii) we've already established that there is another way to resolve emphasis-GIVEN-ness conflicts (i.e. accenting GIVEN material), which as we see in (56), is not a possible resolution strategy under the conditions where Ditransitive Phrasing Generalization counterexamples arise.

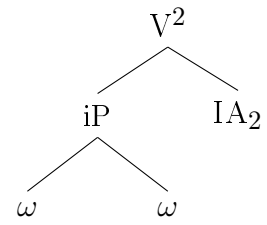
In the remainder of this section, I will argue that the analysis I presented at the end of the previous section is on the right track, despite the fact that it doesn't account for counterexamples to the Ditransitive Phrase Generalization. I propose that prosodic structure of ditransitive verb phrases are built derivationally from the ascending verb phrases (58).

(58) *Ascending ditransitive (without adjuncts)*



Each stage of the derivation is a mapping from a larger syntactic phrasal constituent (than the last) to a prosodic representation. For example, the first stage of the prosodic derivation will be the output of a mapping from V^{1SO}, and the the second will be a mapping from V^{2SO}. I will also assume that the syntactic constituent that is the input to the prosodic mapping is replaced by the the output of the prosodic mapping in the syntactic structure.[29] So the result of mapping V^{1SO} to a prosodic representation could be something like the following:

(59)



Each mapping complies with the Constituent Mapping Conjecture in (60) unless violating the conjecture would resolve a conflict between two or more prosodic requirements.

(60) **Constituent Mapping Conjecture (Phrasal Version)**

A string w is a possible prosodic domain (e.g. an intermediate phrase domain) iff w is analyzable as a **phrasal** syntactic constituent.

As you will see shortly, this system explains why the Ditransitive Phrasing Constraint is obeyed unless prosodic conflicts arise. And more, it explains the distribution of conflict

29. See Pak 2008; Collins and Stabler 2016 for a similar approach. Collins and Stabler (2016) aptly call this the *plug-back-in model*. In many ways, the plug-back-in model is like the cyclic approach to phonology established in Chomsky et al. 1956 and Chomsky and Halle 1968.

resolution strategies.[30]

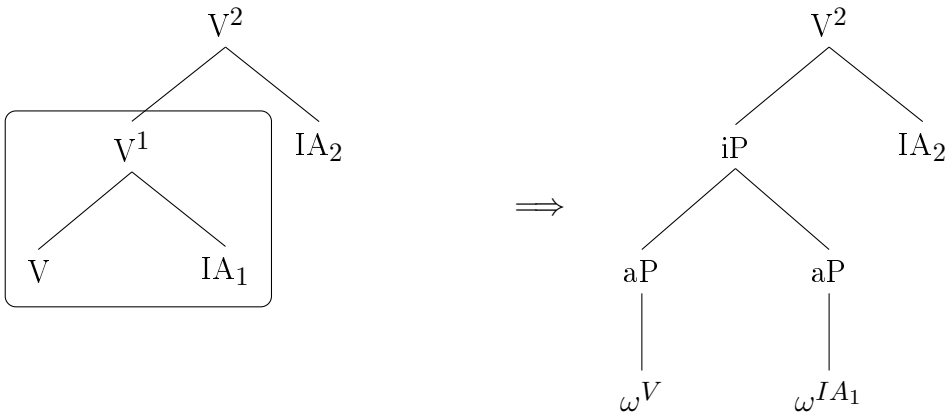
I'll start by showing how the prosodic representation of a ditransitive verb phrase is derived when nothing is GIVEN and nothing is emphasized. Recall that there are two potential phrasings under these conditions:

(61) a. $(\text{V IA}_1\ \text{IA}_2)_{\text{iP}}$

b. $(\text{V IA}_1)_{\text{iP}}\ (\text{IA}_2)_{\text{iP}}$

Of the two phrasings, the one that requires the fewest additional assumptions is (61b), so I'll walk through the derivation of that structure. The derivation begins by mapping the syntactic object $\text{V}^{1^{SO}}$ to an iP, as in (62).

(62)



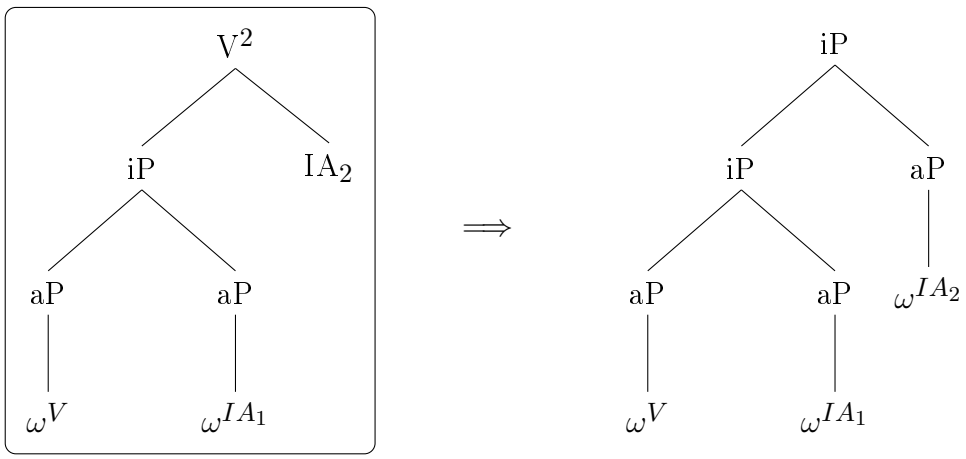
The boxed constituent in (62) indicates the input to the prosodic mapping. The output, which I've displayed inserted back into the syntactic presentation, is an intermediate phrase that immediately contains two accentual phrase (aP); and each accentual phrase immediately contains two prosodic words (ω), which are the verb and IA_1. I assume that every intermediate phrase must immediately contain at least one accentual phrase; we can understand accentual phrases as the domains for pitch accents—that is, every accentual phrase

30. This is very much a sketch of an analysis, and as a result involves several implicit assumptions about the relation between prosodic domains and suprasegmental associations. As a result, many of the details of the derivations will be brushed over.

is associated with one (and only one) pitch accent. This assumption will be necessary to explain some other derivations.

The next step of the derivation maps $V^{2^{SO}}$ to a prosodic structure. This step is presented in (63).

(63)

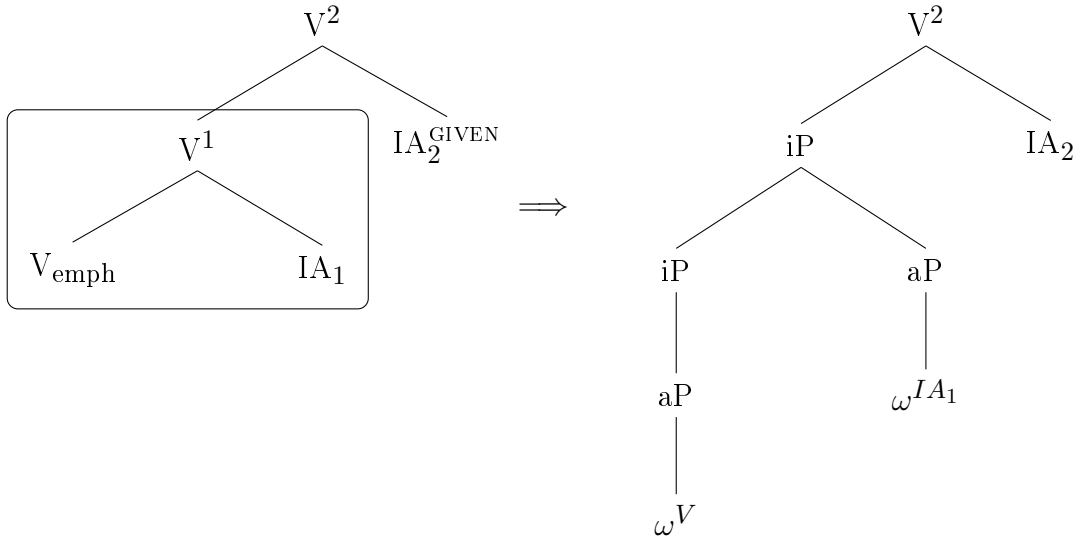


The second stage is the final prosodic representation of the derivation. It is categorized as an intermediate phrase that immediately contains (i) the intermediate phrase that was built at the previous derivational stage and (ii) an accentual phrase that correlates with the linearly second internal argument IA_2. If we were to display intermediate phrasing of this structure linearly, it would be $((V\ IA_1)_{iP}\ IA_2)_{iP}$. However, because I am assuming, with Ladd 2008, that intermediate phrases are the domain for nuclear pitch accent (i.e. final pitch accents in iP are nuclear), it will be the case that IA_1 bears nuclear pitch accent and the verb prenuclear pitch accent in the embedded intermediate phrase; and IA_2 bears nuclear pitch accent in the root larger intermediate phrase. So a simplified linear phrasing is $(V\ IA_1)_{iP}\ (IA_2)_{iP}$.[31]

Let's take an example of a string with the abstract form V_{emph}-IA_1-IA_{GIVEN}. This is the shape of a string that counterexeplifies the Ditransitive Phrasing Generalization. The smallest constituent in the string's corresponding syntactic structure is $[_{V_1}\ V_{emph}\ IA_1]$.

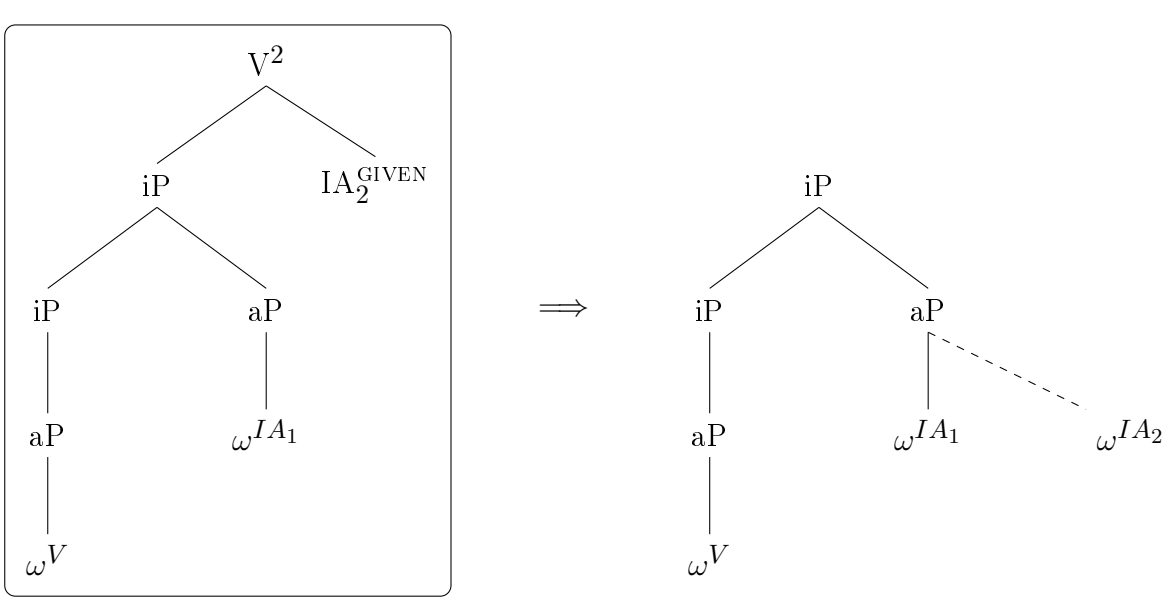
31. As I had in previous sections, when representing structures linearly, I'm ignoring the recursive/hierarchical aspects of the prosodic structure because it's easier to read and should have no bearing on the arguments. I'll call the flattened versions of the representations *simplified prosodic representations*.

(64)



Given that emphasis must be at the right edge of an iP, it must be the case that the syntactic constituent is mapped to the simplified prosodic constituent $(\mathrm{V_{emph}})_{\mathrm{iP}}(\mathrm{IA_1})_{\mathrm{iP}}$. The next step of the derivation will map the next larger syntactic object $\mathrm{V}^{2^{SO}}$.

(65)

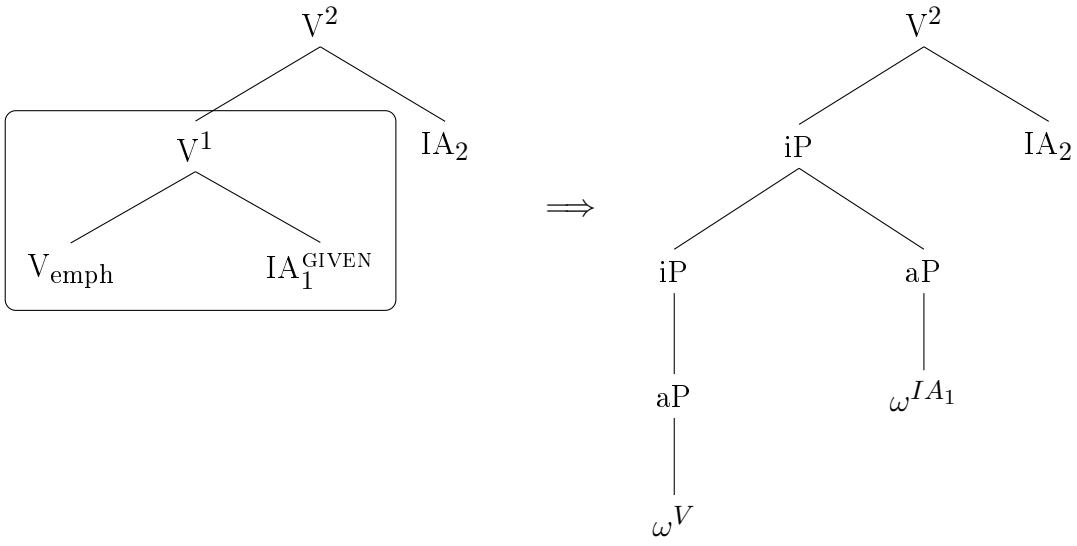


If (i) the prosodic representation at the previous stage were extended and (ii) the entire string is parsed into some iP, then we expect $\mathrm{V}^{2^{SO}}$ to be mapped to the simplified structure $(\mathrm{V_{emph}})_{\mathrm{iP}}(\mathrm{IA_1})_{\mathrm{iP}}(\mathrm{IA_2^{GIVEN}})_{\mathrm{iP}}$. This isn't what we observe empirically. An assumption of

the analysis I'm proposing is that the Constituent Mapping Conjecture is violated if such a violation would resolve a prosodic conflict. At the second stage of the above derivation, an emphasis-GIVEN-ness conflict arises that can be resolved if a non-extending operation parses IA_2 into the same intermediate phrase as IA_1, which is what I diagrammed in (65). The operation is non-extending because the prosodic word correlating with IA_2 (i.e. ω^{IA_2}) has been phrased together with ω^{IA_1}. In the input, there is an accentual phrase (I'll call it aP^{IA_1}) that immediately contains ω^{IA_1} (and only ω^{IA_1}). But in the output, aP^{IA_1} has been replaced by another accentual phrase that immediately contains two prosodic words: ω^{IA_1} and ω^{IA_2}. Essentially, the mapping in (65) is the prosodic analog to Replacement Merge.[32] The result of this non-extending operation is the simplified prosodic $(\text{V}_{\text{emph}})_{\text{iP}}(\text{IA}_1\ \text{IA}_2^{\text{GIVEN}})_{\text{iP}}$, which is exactly what we observe.

Now, I'd like to consider the case of the string V_{emph}-IA_{GIVEN}-IA_2. In particular, I would like to focus on why this string shape result in a conflict that is resolved by accenting GIVEN material as opposed to violating the Constituent Mapping Conjecture. The first stage of the prosodic derivation maps the syntactic object $[_{\text{V}^1}{}^{so'}\ \text{V}_{\text{emph}}\ \text{IA}_{\text{GIVEN}}]$.

(66)

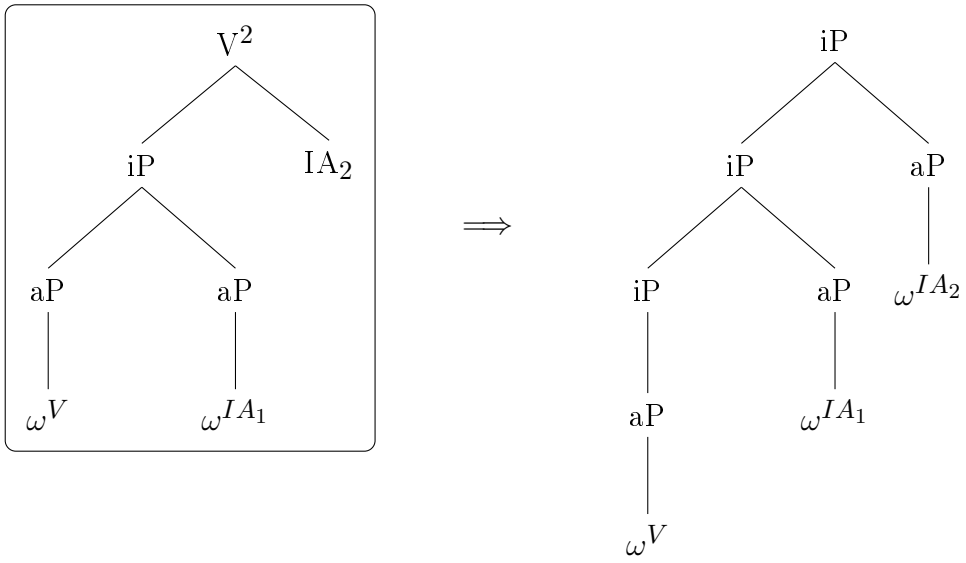


32. For a definition of Replacement Merge, see §2.2.1. For some applications of Replacement Merge in syntactic analyses, see Chapter 3.

At this stage in the derivation, it is impossible for a non-extending operation to assign the GIVEN IA_1 to the same intermediate phrase of the emphasized verb because of the apparently strict requirement that the emphasis is at the right edge on an intermediate phrase. Furthermore, there isn't a non-extending operation (i.e. an operation that would violate the Constituent Mapping Conjecture) that could resolve the conflict. As a result, the only possible licit prosodic representation is one where the GIVEN material forms its own intermediate phrase $(V_{emph})_{iP}(IA_{GIVEN})_{iP}$. I'm assuming, of course, that everything in the string needs to be parsed into some intermediate phrase, and because intermediate phrases need to contain at least one accentual phrase, IA_1 needs to be parsed into an accentual phrase.

Finally, at the next derivational step, the prosodic representation can be extended because IA_2 is not GIVEN and therefore can form its own intermediate phrase without violating any prosodic requirement.

(67)



The final representation is identical to the representation of a ditransitive verb phrase, where (i) none of the material is GIVEN and (ii) the verb is emphasized, like in (43). And what we find is that the simplified prosodic representation is $(V)_{iP}$ $(IA_1)_{iP}$ $(IA_2)_{iP}$.

Because the argument is complex and the analysis is very sketchy, a high level overview of the argument is warranted. To start, a derivational analysis that is by default extending (in sense of extending I've used throughout this thesis) can capture all of the data that complies with the Ditransitive Phrasing Generalization. I think this point can be made stronger by a fully fleshed out theory of prosodic well-formedness, which admittedly has not been given in this chapter. As for the counterexamples to the Ditransitive Phrasing Generalization, the idea behind my explanation of these is that they arise when prosodic conflicts present themselves and when a non-extending operation can resolve the conflict. A prosodic non-extending operation is defined exactly as a syntactic non-extending operation would be defined. Finally, we only observe accenting of GIVEN material when (i) a conflict arises between the prosodic requirements of GIVEN material and emphasized material and (ii) no non-extending operation could resolve the conflict. The only time a non-extending operation can't resolve a conflict of this kind is when there isn't any accessible intermediate phrase for the GIVEN material to be tucked into. As a matter of fact, only GIVEN material in position IA_1 can receive a nuclear accent in these conditions, and this fact follows very nicely from an analysis that posits step-by-step structure building from the bottom of an ascending syntactic representation to the top.

Alternative accounts may (i) calculate the output on the basis of the global syntactic structure or (ii) build prosodic structure in a strictly extending fashion. Theoretically, neither approach can posit the intermediate representation $(V_{emph})_{iP}(IA_{GIVEN})_{iP}$ in their analysis of strings of the form V_{emph}-IA_{GIVEN}-IA_2. In §4.2.3, I will present evidence from ditransitives verb phrases with epithetic internal arguments that supports the claim that $(V_{emph})_{iP}(IA_{GIVEN})$ is, in fact, an intermediate representation. This data presents compelling evidence, therefore, that non-extending prosodic structure-building operations are permitted in the derivation of prosodic representations. As a consequence, I conclude that prosodic structure building is derivational.

4.2.3 The effect of epithets on prosodic phrasing

In the previous section, I presented a sketch of a derivational prosodic-structure building theory, and I provided an analysis of some complex intermediate phrasings in (Erik Zyman's) English. The analysis accurately explains the strong tendency against the two internal arguments of a ditransitive verb phrase to be prosodically phrased together to the exclusion of the verb (i.e. what I call the Ditransitive Phrasing Generalization). It also explains why the same prosodic conflict is resolved in two different ways, and more, it explains the distribution of those resolution strategies.

In this section, I will use epithets to further support the analysis. Because epithets can absolutely never bear nuclear stress and one of the resolution strategies discussed in the previous section involves assigning nuclear stress to something that is typically resistant to nuclear stress, epithets provide us with a tool for testing whether or not intermediate prosodic representations are evaluated for well-formedness, or if instead, prosodic representations are evaluated globally. I will show you that conflicts that arise between epithets' accent-proofing and conditions on emphasized material are sometimes never resolved. I argue that this follows if prosodic representations are built derivationally, and at the derivational stage where such a conflict arises, there is no alternative resolution strategy. In other words, the interaction between epithets and emphasis provide evidence for intermediate prosodic representations.

Ladd (2008, pp.299-302) argues that nuclear and prenuclear pitch accents are categorically different because English epithets can bear prenuclear but not nuclear accent. For Ladd (like us), the domain of nuclear pitch accent is the intermediate phrase, which means that each intermediate phrase will necessarily contain a nuclear pitch accent. Recall that the rightmost pitch accent in an intermediate phrase is a nuclear pitch. If an epithet occurs in a position within an intermediate phrase where it would be nuclear, the epithet is deaccented, as in (68). (Small caps signifies nuclear stress/accent.)

(68) *Ali:* Everything okay after your operation?

Dani: Don't talk to me about it! I'd like to STRANGLE the butcher!

\# *Dani:* Don't talk to me about it! I'd like to strangle the BUTCHER! (Ladd, 2008, p.300, ex.22-23)

In Dani's felicitous response, where *butcher* is deaccented, it is clear that *the butcher* is interpreted non-literally (i.e. Dani isn't referring to a person whose profession is to butcher or sell meat). While Dani's infelicitous response is perfectly grammatical (and felicitous in other contexts), by placing a pitch accent on the initial syllable of *butcher*, *the butcher* can only be interpreted literally.

There isn't a general ban on pitch accenting epithets, though. They can bear prenuclear pitch accents. For example, the subject epithet *the butcher* in (69a) is aligned with a prenuclear pitch accent, and yet, the example is acceptable. We know that the accent on the epithet is prenuclear because there is no phrase accent that falls between it and the next pitch accent to its right. In (69b), where the subject aligns with the final pitch accent before a phrase accent—that is, a nuclear pitch accent—*the butcher* can only be interpreted literally; it has a non-epithetic reading.

(69) *Ali:* Everything okay after your operation?

Dani: Don't talk to me about it!

a. ∼H ∼H H*L_ (The butcher charged me a thousand bucks)$_{iP}$. (Ladd, 2008, p.300, ex.24)

b. \# H* L_ ∼H H*L_ (The butcher)$_{iP}$ (charged me a thousand bucks)$_{iP}$. (Ladd, 2008, p.300, ex.25)

It follows from Ladd's analysis that internal argument epithets are licit under certain conditions. First, if the epithet is deaccented, it cannot bear nuclear accent, and therefore, no violation arises (70a) & (70b).

(70) *Ali:* I heard Molly sold her bike to Tony.

a. (No,) (She loaned her rollerblades ∼H to the bastard H* **L**_)$_{iP}$.

b. (No,) (She loaned the piece of junk H* **L**_)$_{iP}$ (to Robin H*__**L**__)$_{iP}$.

Second, if the epithet aligns with a pitch accent, the epithet cannot be the final pitch-accented element in its intermediate phrase. For example, the direct object epithet in (71) licitly aligns with a prenuclear pitch accent.

(71) *Ali:* I heard Molly loaned her bike to Tony.

(No,) (She loaned the piece of junk ∼H to Robin H*__**L**__)$_{iP}$.

It's clear from Ladd's data and the additional data in (70) that epithets cannot align with nuclear pitch accents. But how does this ban on associating nuclear pitch accent interact with the other processes we covered in §4.2.1?

Remember that emphasized elements always occur at the immediate right edge of an intermediate phrase. In addition to bearing a nuclear pitch accent, emphasized elements are typically followed immediately by a strong prosodic break that is indicative of an intermediate phrase boundary. It's generally the case that GIVEN material can't bear nuclear pitch accent, and as a result, is deaccented in a post-nuclear position. Furthermore, there appears to be a ban on phrasing the two internal arguments of a ditransitive verb phrase together to the exclusion of the verb (*(V)(IA IA)).

Interestingly, if the verb is emphasized, and either IA_1 or IA_2 is GIVEN (but not both), there is a conflict that must be resolved. We observed in (53) that, when IA_1 is the GIVEN internal argument, the resolution strategy is to assign a nuclear pitch accent to the GIVEN IA_1. And in (55), we saw that the resolution strategy is to phrase the two internal arguments together into a single intermediate phrase, when IA_2 is the GIVEN internal argument.

In this section, we've uncovered another factor—the ban on associating epithets with nuclear pitch accent. If epithets are like (other) GIVEN material, then we predict that epithets can bear nuclear pitch accent when (i) it is IA_1 in a ditransitive verb phrase and (ii) the verb is emphasized. What we'll find is that epithets are not like other GIVEN material in this way. Instead, in the particular configuration we're considering, there is no licit way to assign a prosodic representation to the ditransitive verb phrase. So epithets don't behave like other deaccented material when they are in the IA_1 position. But when epithets are in the IA_2 position, they behave exactly like other deaccented material: They are deaccented and phrased together with IA_1 to the exclusion of the emphasized verb.

First, let's consider examples with an emphasized verb and an epithet in the IA_1 position of a prepositional dative construction.[33] If each of the internal arguments form separate intermediate phrases, in accordance with previously observed constraints on prosodic representations, then the epithet will align with a nuclear accent. The resulting (72a) is infelicitous in the provided context; *the piece of junk* is interpreted literally.

(72) *Ali:* What did Molly do with her bike?

a. # (She *loaned*)$_{\text{iP}}$ (the piece of junk)$_{\text{iP}}$ (to Robin)$_{\text{iP}}$.
   H* **L**_ H*L_ H*L_

b. * (She *loaned*)$_{\text{iP}}$ (the piece of junk to Robin)$_{\text{iP}}$.
   H* **L**_ ∼H H* **L**_

c. * (She *loaned*)$_{\text{iP}}$ (the piece of junk to Robin)$_{\text{iP}}$.
   H* **L**_ H* **L**_

This example also demonstrates that epithets in internal argument positions can't be accented to repair an ill-formed utterance, which contrasts with GIVEN non-epithets, as we saw in (53). Another possible strategy is considered in (72b) & (72c). In those examples, the two internal arguments are phrased together to the exclusion of the verb. This isn't

33. I haven't included any double object construction data, but I can confirm that the patterns are exactly the same as with the prepositional dative construction—that is, at least for Erik Zyman.

possible for GIVEN non-epithets either (55), so in this way, epithets behave like other GIVEN material.

There is another way in which epithets behave like GIVEN non-epithets. When the verb is emphasized, IA_1 isn't GIVEN, and IA_2 is an epithet, it's possible to parse the (deaccented) epithet into an intermediate phrase with the preceding IA_1, as you see in (73).

(73) *Ali:* Why was Molly over at Robin's today?

Dani: I can't believe it!

H* **L**_ H* **L**_
(She *loaned*)$_{iP}$ (my bike to the bastard)$_{iP}$.

One may object to the claim that the the epithet is in the same intermediate phrase as IA_1, and offer as an alternative that the epithet is not parsed into an intermediate phrase at all. If this alternative were true, then we would predict that epithets could occur in the IA_2 position when IA_1 bears emphasis (and therefore occurs at the right edge of an intermediate phrase). This prediction is not borne out. We can see in (74) that an epithet in the IA_2 position can't follow an emphasized IA_1.

(74) *Ali:* Why was Molly over at Robin's today?

Dani: I can't believe it!

∼H H* **L**_ H* **L**_
\# (She loaned my *bike*)$_{iP}$ (to the bastard)$_{iP}$.

We can explain (72b), (72c), (74) if we posit a constraint against epithets being right-adjacent to emphasis, but this constraint undergenerates. An epithet can be right adjacent to emphasis if the epithet is at the left edge of a complex constituent (75).

(75) *Ali:* Why was Molly over at Robin's today?

H* **L**_ ∼H H*L_
Dani: (So she could *borrow*)$_{iP}$ (the bastard's bike)$_{iP}$.

What we've discovered in this section is that epithets have the same properties as other GIVEN material, except that epithets can't bear a nuclear pitch accent under any circumstance. While it's possible for GIVEN non-epithets to bear nuclear pitch accents in order to resolve some conflict, it's not the case for epithets. Interestingly, despite the fact that there is a demonstrable alternative resolution strategy—phrasing the two internal arguments together to the exclusion of the verb—that resolution strategy is only available to epithets in the environment where it's available for GIVEN non-epithets (i.e. in position IA_2). That resolution strategy is not available in position IA_1, where the resolution strategy for GIVEN non-epithets is to assign a nuclear pitch accent.

At the start of this section, I said the epithet data in this section will provide evidence for the existence of an intermediate representation. In particular, I said that the non-strictly extending theory predicts that $(\text{V}_{\text{emph}})_{\text{iP}}(\text{IA}_{\text{GIVEN}})_{\text{iP}}$ is an intermediate representation in the derivation of a ditransitive verb phrase of the form V_{emph}-IA_{GIVEN}-IA_2, while the alternative theories do not. Evidence for intermediate representations supports non-strictly extending (derivational) analyses of prosodic structure building over alternative theories like (i) globally evaluative theories or (ii) strictly extending derivational theories.

I believe that the ineffable string V_{emph}-$\text{IA}_{\text{EPITHET}}$-$\text{IA}_2$ provides evidence for the intermediate representation $(\text{V}_{\text{emph}})_{\text{iP}}(\text{IA}_{\text{GIVEN}})_{\text{iP}}$. Recall that, in the proposed non-extending theory, the first step in the prosodic derivation of a ditransitive verb phrase is the mapping of the syntactic object $[_{\text{V}^1}\ \text{V}\ \text{IA}_1]$. In the previous section, I noted that IA_1 can exceptionally bear a nuclear pitch accent in the output of the mapping iff the verb is emphasized. The exceptional nuclear pitch accenting resolves an apparent conflict between several prosodic requirements. The gist of the analysis is that, in the face of conflict, something has to give.

We have discovered in this section that epithets can't bear nuclear pitch accent under any set of conditions. So when a syntactic object $[_{\text{V}^1}\ \text{V}_{\text{emph}}\ \text{IA}_{\text{EPITHET}}]$ is mapped to a prosodic representation, every possible prosodic representation is ill-formed: There is no

way to resolve the conflict; nothing will give.

Furthermore, this explanation captures an even stronger generalization:

(76) **The GIVEN IA_1 Generalization**

There aren't any counterexamples to the Ditransitive Phrasing Generalization where IA_1 is GIVEN.

Under the non-strictly extending derivational theory, the reason that GIVEN material is never in position IA_1 in counterexamples to Ditransitive Phrasing Generalization follows from the fact that a conflict arises and is resolved before IA_2 is accessible to the prosodic derivation. When IA_1 is GIVEN, and therefore, contributes to the only available resolution strategy is exceptional pitch accenting. If IA_1 is an epithet, the exceptional pitch accenting isn't possible, so the derivation crashes; if IA_1 is a GIVEN non-epithet, exceptional pitch accenting occurs, the conflict is resolved, and there is no conflict to resolve at the next stage of the derivation, when IA_2 is accessible.

In a global evaluation framework, the entire syntactic structure is accessible, and thus, we predict (at least on the face of it) that the conflict can be resolved by phrasing the epithet together with IA_2. This, as we have seen, is not possible. As for the strictly extending derivational framework, it's actually very unclear how such a theory can account for basic cases of non-isomorphism between syntax and prosody.

4.3 What prosodic derivations can tell us about syntax

The takeaway from Chapter 3 is that finding evidence for purported intermediate representations of syntactic derivations is extremely difficult. The promise of this chapter was that I would present evidence for intermediate representations of prosodic derivations. The strategy was to show that certain kinds of prosodic representations could only be built by a non-extending structure-building operations. I argued that counterexamples to the Ditransi-

tive Phrasing Generalization must be built using a non-extending prosodic structure-building operation. This claim is supported by two kinds of evidence. One, the distribution of resolution strategies for conflicts between emphasized material and GIVEN material. And two, the interaction between epithets and emphasized material.

There are several ways in which prosodic structure building is different from syntactic structure building. Most notably, there is a lot of evidence that prosodic structure building is sensitive to the output of syntactic structure building (Chomsky et al., 1956; Chomsky and Halle, 1968; Kiparsky, 1982; Selkirk, 1984; Truckenbrodt, 1999; Pak, 2008; Kalivoda, 2018), while the opposite doesn't seem to be true. However, as I hope I've shown in this chapter, we may be able to argue backwards from prosodic patterns to properties of the corresponding syntactic structures.

Debates over the structure of ditransitive verb phrases have been ongoing for several decades, and there is no clear consensus on the matter. In this chapter, I have argued that both syntactic and prosodic properties of (Erik Zyman's) English can be explained if the ascending verb phrase analysis of ditransitives is the correct analysis (i.e. either alone or in parallel with a descending structure). The ascending verb phrase analysis is supported by patterns of syntactic movement and ellipsis, but it's also supported by patterns of intonational/intermediate phrasing. Simple cases of prosodic phrasing have simple explanations when we take the ascending verb phrase to be the relevant verb phrase for the mapping between syntactic and prosodic structure. And more, complex cases of prosodic phrasing (e.g. cases of ineffability and cases of apparent non-isomorphism) can be given a fairly simple derivational analysis if the ascending structure is the input to the mapping from syntax to prosody.

CHAPTER 5

CONCLUSION

In the introduction I said that I hoped to convince you that derivations will not have any bearing on the predictions of your theory if your theory complies totally with the Extension Condition. And I think that the arguments and proof in Chapter 2 should have been enough to compel you. The formalization I develop in §2.2 is my best effort at capturing what I take to be the spirit of the Extension Condition, and by that I mean the strictest possible version of the Extension Condition that I know of. The proof is not a proof of the primary claim of the chapter—that is, that every strictly extending theory has a strongly equivalent declarative alternative. Instead, I prove that Collins and Stabler's (2016) strictly extending theory in particular has a strongly equivalent declarative alternative. But it should be clear that the formalization presented in Collins and Stabler 2016 is representative of minimalist theories generally; in fact, it is the stated goal of the article to formalize what they consider to be fundamental notions in minimalist syntax. For the purposes of the thesis of Chapter 2, the relevant notion is Merge (i.e. the only structure-building operation). Their definition of Merge, like most recent definitions, is definitely a member of the Extension Condition Family.

I also made a call to action in the introduction. I said that theories that make use of non-extending operations derive intermediate constituents, and I hope syntacticians that posit non-extending operations attempt to find evidence of those intermediate representations. As I showed in Chapter 3 (especially §3.2 and §3.3), it can be very challenging to find any evidence of intermediate constituents. I failed to do so. However, if we do discover evidence of intermediate constituents (and therefore of derivations), it can reveal something really deep and interesting about the nature of natural language.

In Chapter 4, I argue that there is some positive evidence for intermediate representations in a derivational theory of prosodic structure building. The argument was quite complex and based on some complicated data. In retrospect it isn't surprising to me, but when I first set out to find evidence of intermediate constituents, I didn't think I would have to look at the interaction between the syntax of ditransitives, intonational contours, information structure, and epithets to find it.

You may wonder why I was able to find evidence for prosodic intermediate constituents when I couldn't find any evidence for syntactic intermediate constituents. Of course, it could be that I failed to find something in the syntax that was there to be found. Another possibility is that there is something fundamentally different about the two components of grammar. I wish I had something more interesting to say about it.

REFERENCES

Abels, K. (2012). *Phases: An Essay on Cyclicity in Syntax*, Volume 543. Walter de Gruyter.

Adger, D. (2003). *Core syntax: A minimalist approach*. Oxford University Press.

Adger, D. (2010). A Minimalist theory of feature structure. In *Features: Perspectives on a Key Notion in Linguistics*. Oxford University Press.

Ahn, B. T. (2015). *Giving Reflexivity a Voice: Twin Reflexives in English*. Ph. D. thesis, University of California, Los Angeles.

Aissen, J. and D. M. Perlmutter (1976). Clause Reduction in Spanish. In *Proceeding of the 2nd Annual Meeting of the Berkeley Linguistics Society*, Berkeley, CA, pp. 1–30.

Arregi, K. (2016). Focus Projection Theories. In *Oxford Handbook of Information Structure*, Volume 1, pp. 185–202. Oxford University Press.

Baltin, M. R. (1982). A Landing Site Theory of Movement Rules. *Linguistic Inquiry 13*(1), 1–38.

Bar-Hillel, Y. (1953). A Quasi-Arithmetical Notation for Syntactic Description. *Language 29*(1), 47.

Barnes, J., N. Veilleux, A. Brugos, and S. Shattuck-Hufnagel (2010). Turning points, tonal targets, and the English L-phrase accent. *Language and Cognitive Processes 25*(7-9), 982–1023.

Barss, A. and H. Lasnik (1986). A note on anaphora and double objects. *Linguistic Inquiry 17*(2), 347–354.

Beckman, M. E. and J. B. Pierrehumbert (1986). Intonational structure in Japanese and English. *Phonology 3*, 255–309.

Bolinger, D. (1977). *Meaning and Form*. Longman.

Bošković, Ž. (2004). Be careful where you float your quantifiers. *Natural Language and Linguistic Theory 22*(4), 681–742.

Bowers, J. S. (2018). *Deriving Syntactic Relations*. Number 151 in Cambridge Studies in Linguistics. Cambridge, United Kingdom: Cambridge University Press.

Bresnan, J. (1982). Control and complementation. *Linguistic Inquiry 13*(3), 343–434.

Brody, M. (1995). *Lexico-Logical Form: A Radically Minimalist Theory*, Volume 27. MIT press.

Bruening, B. (2001). QR obeys superiority: Frozen scope and ACD. *Linguistic Inquiry 32*(2), 233–273.

Bruening, B. (2014). Precede-and-command revisited. *Language 90*(2), 342–388.

Bruening, B. (2018). CPs move rightward, not leftward. *Syntax 21*(4), 362–401.

Büring, D. (2003). On D-Trees, Beans, And B-Accents. *Linguistics and Philosophy 26*, 511–545.

Büring, D. (2016, 07). *Intonation and Meaning*. Oxford University Press.

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

Cheng, L. L.-S. and L. J. Downing (2016). Phasal syntax = cyclic phonology? *Syntax 19*(2), 156–191.

Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on information theory 2*(3), 113–124.

Chomsky, N. (1957). *Syntactic Structures*. Mouton, The Hague.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. MIT press.

Chomsky, N. (1970). Remarks on Nominalization. In R. Jacobs and P. S. Rosenbaum (Eds.), *Readings in English Transformational Grammar*, pp. 184–221. Ginn and Company.

Chomsky, N. (1973). Conditions on transformations. In S. R. Anderson and P. Kiparsky (Eds.), *A Festschrift for Morris Halle*. Holt, Rinehart and Winston.

Chomsky, N. (1977). *Essays on form and interpretation*. Elsevier North-Holland, Inc.

Chomsky, N. (1981). *Lectures on government and binding*. Foris Publications, Dordecht, Holland.

Chomsky, N. (1993). A minimalist program. In K. Hale and S. J. Keyser (Eds.), *View from building 20: Essays in Linguistics in Honor of Sylvain Bromberger*, pp. 1–52. MIT Press.

Chomsky, N. (1995). *The Minimalist Program*. MIT Press, Cambridge, Mass.

Chomsky, N. (2000). Minimalist inquiries: The framework. In R. Martin, D. Michaels, J. Uriagereka, and S. J. Keyser (Eds.), *Step by Step: Essays on Minimalist Syntax in Honor of Howard Lasnik*, pp. 89–155. MIT Press.

Chomsky, N. (2001). Derivation by phase. In M. Kenstowicz (Ed.), *Ken Hale: a life in language*, pp. 1–52. MIT Press.

Chomsky, N. (2005). Three Factors in Language Design. *Linguistic Inquiry 36*(1), 1–22.

Chomsky, N. (2007). Approaching UG from Below. In U. Sauerland and H.-M. Gärtner (Eds.), *Interfaces + Recursion = Language?*, pp. 1–30. Berlin, München, Boston: DE GRUYTER.

Chomsky, N. (2008). On phases. *Current Studies in Linguistics Series: Essays in honor of Jean-Roger Vergnaud 45*, 133.

Chomsky, N. and M. Halle (1968). *The sound pattern of English*. Harper and Row.

Chomsky, N., M. Halle, and F. Lukoff (1956). On accent and juncture in English. In M. Halle, H. Lunt, and C. v. Schooneveld (Eds.), *For Roman Jakobson: Essays on the occasion of his sixtieth birthday*. Mouton.

Citko, B. (2005). On the nature of Merge: External Merge, Internal Merge, and Parallel Merge. *Linguistic Inquiry 36*(4), 475–496.

Collins, C. (1997). *Local Economy*, Volume 29. MIT press.

Collins, C. (2017). Merge(X,Y) = {X,Y}. In L. Bauke and A. Blümel (Eds.), *Labels and Roots*, Number 128 in Studies in generative grammar, pp. 47–68. De Gruyter Mouton.

Collins, C. and E. Groat (2018). Distinguishing copies and repetitions.

Collins, C. and E. Stabler (2016). A formalization of minimalist syntax. *Syntax 19*(1), 43–78.

Dowty, D. R., R. E. Wall, and S. Peters (1981). *Introduction to Montague Semantics*, Volume 11 of *Studies in Linguistics and Philosophy*. Kluwer Academic Publishers.

Elfner, E. J. (2012). *Syntax-prosody interactions in Irish*. University of Massachusetts Amherst.

Epstein, S. D., E. M. Groat, R. Kawashima, and H. Kitahara (1998). *A Derivational Approach to Syntactic Relations*. Oxford University Press.

Epstein, S. D., H. Kitahara, and T. D. Seely (2012). Structure building that can't be. In *Ways of structure building*. Oxford University Press.

Ermolaeva, M. (2021). *Learning Syntax via Decomposition*. PhD, University of Chicago, Chicago.

Ermolaeva, M. and G. M. Kobele (2023). Agreeing Minimalist Grammars. ms.

Fitzpatrick, J. M. (2002). On Minimalist Approaches to the Locality of Movement. *Linguistic Inquiry 33*(3), 443–463.

Freidin, R. (1978). Cyclicity and the theory of grammar. *Linguistic Inquiry 9*(4), 519–549.

Freidin, R. (1999). Cyclicity and minimalism. In S. D. Epstein and N. Hornstein (Eds.), *Working Minimalism*, pp. 95–126. MIT Press.

Guimarães, M. (2000). In Defense of Vacuous Projections in Bare Phrase Structure. *University of Maryland Working Papers in Linguistics 9*, 90–115.

Hankamer, J. (1977). Multiple analyses. In *Mechanisms of Syntactic Change*, pp. 583–608. New York, USA: University of Texas Press.

Heck, F. (2016). *Non-Monotonic Derivations*. Habilitation, Leipzig University.

Heck, F. and G. Müller (2007). Extremely local optimization. In *Proceedings of WECOL*, Volume 26, pp. 170–183.

Hoeksema, J. (2000). Negative Polarity Items: Triggering, Scope, C-Command. In L. R. Horn and K. Yasuhiko (Eds.), *Negation and Polarity: Semantic and Syntactic Perspectives*, pp. 123–154. Oxford University Press.

Hopcroft, J. E. and J. D. Ullman (1979). *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley.

Izvorski, R. (1995). Wh-movement and Focus-movement in Bulgarian. In *Proceedings of CONSOLE 2*, pp. 54–67.

Jackendoff, R. (1990). On Larson's Treatment of the Double Object Construction. *Linguistic Inquiry 21*(3), 427–256.

Janke, V. and A. Neeleman (2012). Ascending and descending VPs in English. *Linguistic Inquiry 43*(1), 151–190.

Kalivoda, N. (2018). *Syntax-prosody mismatches in Optimality Theory*. University of California, Santa Cruz.

Kanerva, J. M. (1989). *Focus and phrasing in Chichewa*. PhD Thesis, Stanford University, Palo Alto, California.

Kaplan, R. M. (1994). The Formal Architecture of Lexical-Functional Grammar. In M. Dalrymple, R. M. Kaplan, J. T. Maxwell III, and A. Zaenen (Eds.), *Formal Issues in Lexical-Functional Grammar*, pp. 1–21. CSLI Publications.

Kaplan, R. M. and A. Zaenen (1989). Long-distance Dependencies, Constituent Structure, and Functional Uncertainty. In *Alternative conceptions of phrase structure*. University of Chicago Press.

Kayne, R. S. (1994). *The antisymmetry of syntax*. MIT press.

Kayne, R. S. (2008). Antisymmetry and the Lexicon.

Kiparsky, P. (1982). Lexical Morphology and Phonology. In SICOL and H. Ŏnŏ Hakhoe (Eds.), *Linguistics in the morning calm: Selected papers from SICOL-1981*, pp. 1–91. Hanshin Pub. Co.

Kitahara, H. (1995). Target α: Deducing strict cyclicity from derivational economy. *Linguistic Inquiry 26*(1), 47–77.

Kitahara, H. (2011). Relations in minimalism. *English Linguistics 28*(1), 1–22.

Kratzer, A. and E. Selkirk (2020). Deconstructing information structure. *Glossa: a journal of general linguistics 5*(1), 1–53.

Ladd, D. R. (2008). *Intonational phonology*. Cambridge University Press.

Larson, R. K. (1988). On the double object construction. *Linguistic Inquiry 19*(3), 335–391.

Larson, R. K. (1990). Double objects revisited: Reply to jackendoff. *Linguistic Inquiry 21*(4), 589–632.

Lasnik, H. and J. J. Kupin (1977). A restrictive theory of transformational grammar. *Theoretical Linguistics 4*, 173–96.

Lasnik, H. and M. Saito (1993). *Move α: Conditions on its application and output*. Current studies in linguistics. Massachusetts Institute of Technology.

Lebeaux, D. (1991). Relative Clauses, Licensing, and the Nature of the Derivation. In S. Rothstein (Ed.), *Perspectives on Phrase Structure: Heads and Licensing*, pp. 209–239. Brill.

Lee, S. J. and E. Selkirk (2022). Xitsonga tone: The syntax–phonology interface. In H. Kubozono, A. Mester, and J. Ito (Eds.), *Prosody and Prosodic Interfaces*, pp. 337–373. Oxford University Press.

Liberman, M. and A. Prince (1977). On stress and linguistic rhythm. *Linguistic Inquiry 8*(2), 249–336.

McCarthy, J. J. and A. Prince (2004). The Emergence of the Unmarked. In J. J. McCarthy (Ed.), *Optimality Theory in Phonology*, pp. 483–494. Wiley.

McCawley, J. D. (1968). Concerning the Base Component of a Transformational Grammar. *Foundations of Language 4*(3), 243–269.

Merchant, J. (2019). Roots don't select, categorial heads do: Lexical-selection of PPs may vary by category. *The Linguistic Review 36*(3), 325–341.

Milway, D. (2021). A Formalization of Agree as a Derivational Operation.

Milway, D. (2023). A formalization of Agree as a derivational operation. *Biolinguistics 17*, 1–39.

Montague, R. (2002). *The Proper Treatment of Quantification in Ordinary English*, Chapter 1, pp. 17–34. John Wiley & Sons, Ltd.

Müller, G. (2010). On deriving CED effects from the PIC. *Linguistic Inquiry 41*(1), 35–82.

Müller, G. (2023). Challenges for cyclicity. *Linguistische Arbeits Berichte 95*, 3–72.

Müller, G. (2017). Structure removal: An argument for feature-driven Merge. *Glossa: a journal of general linguistics 2*(1), 28.

Müller, G. (2020). Rethinking restructuring. In *Syntactic architecture and its consequences II: Between syntax and morphology*, pp. 149–190. Berlin: Zenodo.

Nissenbaum, J. W. (2000). *Investigations of covert phrase movement*. PhD, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Nunes, J. (2001). Sideward movement. *Linguistic Inquiry 32*(2), 303–344.

Pak, M. (2008). *The postsyntactic derivation and its phonological reflexes (Unpublished doctoral dissertation)*. Ph. D. thesis, University of Pennsylvania.

Partee, B. H., A. t. Meulen, and R. E. Wall (1993). *Mathematical methods in linguistics* (Corrected second printing of the first ed.). Number volume 30 in Studies in linguistics and philosophy. Dordrecht Boston London: Kluwer Academic Publishers.

Pesetsky, D. (1989). Language specific processes and the earliness principle. ms.

Pesetsky, D. (1995). *Zero syntax: Experiencers and cascades*. MIT press.

Pesetsky, D. (2021). Exfoliation: Towards a Derivational Theory of Clause Size. ms.

Phillips, C. (1996). *Order and structure*. Ph. D. thesis, Massachusetts Institute of Technology, Cambridge, MA.

Phillips, C. (2003). Linear order and constituency. *Linguistic Inquiry 34*(1), 37–90.

Pierrehumbert, J. B. (1980). *The phonology and phonetics of English intonation*. Ph. D. thesis, Massachusetts Institute of Technology.

Pollard, C. and I. A. Sag (1994). *Head-Driven Phrase Structure Grammar*. University of Chicago Press.

Radford, A. (1988). *Transformational Grammar: A First Course*. Cambridge University Press.

Richards, N. (1997). *What Moves Where When in Which Language?* Ph. D. thesis, Massachusetts Institute of Technology.

Richards, N. (1999). Featural cyclicity and the ordering of multiple specifiers. *Current Studies in Linguistics Series 32*, 127–158.

Richards, N. (2001). *Movement in Language: Interactions and architectures*. Oxford University Press.

Richards, N. (2004). Against Bans on Lowering. *Linguistic Inquiry 35*(3), 453–463. MIT Press.

Rizzi, L. (1990). *Relativized Minimality*, Volume 16. The MIT Press.

Rooth, M. (1992). A theory of focus interpretation. *Natural Language Semantics 1*(1), 75–116.

Rudin, C. (1988). On multiple questions and multiple WH fronting. *Natural Language and Linguistic Theory 6*(4), 445–501.

Rudin, C. (1990). Topic and Focus in Bulgarian. *Acta Linguistica Hungarica 40*(3/4), 429–447.

Rudin, C. (2013). *Aspects of Bulgarian syntax: Complementizers and WH constructions* (2nd ed.). Slavica Publishers.

Sande, H., P. Jenks, and S. Inkelas (2020). Cophonologies by ph(r)ase. *Natural Language & Linguistic Theory 38*, 1211–1261.

Schwarzschild, R. (1999). GIVENness, AvoidF and other constraints on the placement of accent. *Natural language semantics 7*(2), 141–177.

Selkirk, E. (1981). On the nature of Phonological Representation. In T. Meyers, J. Laver, and J. Anderson (Eds.), *The Cognitive Representation of Speech*, pp. 379–388. North-Holland Publishing Company.

Selkirk, E. (1986). On derived domains in sentence phonology. *Phonology 3*, 371–405.

Selkirk, E. (1996). The Prosodic Structure of Function Words. In J. J. McCarthy (Ed.), *Optimality Theory in Phonology* (1 ed.)., pp. 464–482. Wiley.

Selkirk, E. (2000). The interaction of constraints on prosodic phrasing. In *Prosody: Theory and experiment*, pp. 231–261. Springer.

Selkirk, E. (2011). The syntax-phonology interface. *The handbook of phonological theory 2*, 435–483.

Selkirk, E. O. (1984). *Phonology and syntax: the relationship between sound and structure*. MIT press.

Shieber, S. M. (1986). *An Introduction to Unification-Based Approaches to Grammar*. Brookline, Massachusetts: Mictrotome.

Shieber, S. M. (1988). Separating linguistic analyses from linguistic theories. In *Natural language parsing and linguistic theories*, pp. 33–68. Springer.

Stabler, E. (1997). Derivational minimalism. In C. Retoré (Ed.), *Logical Aspects of Computational Linguistics*, pp. 68–95. Springer Berlin Heidelberg.

Stanton, J. (2016). Wholesale Late Merger in Ā-movement: Evidence from Preposition Stranding. *Linguistic Inquiry 47*(1), 89–126.

Steedman, M. (1991). Structure and intonation. *Language 67*(2), 260–296.

Steedman, M. (1996). *Surface Structure and Interpretation*. Linguistic Inquiry Monographs. MIT Press.

Takahashi, S. and S. Hulsey (2009, July). Wholesale Late Merger: Beyond the A/Ā Distinction. *Linguistic Inquiry 40*(3), 387–426.

Thoms, G. and G. Walkden (2019). vP-fronting with and without remnant movement. *Journal of Linguistics 55*(1), 161–214.

Truckenbrodt, H. (1995). *Phonological phrases: Their relation to syntax, focus, and prominence*. Ph. D. thesis, Massachusetts Institute of Technology.

Truckenbrodt, H. (1999). On the relation between syntactic phrases and phonological phrases. *Linguistic Inquiry 30*(2), 219–255.

Uriagereka, J. (1999). Multiple spell-out. *Current Studies in Linguistics Series 32*, 251–282.

Van Riemsdijk, H. and E. Williams (1971). NP-Structure. *The Linguistic Review 1*(2), 171–217.

Van Riemsdijk, H. and E. Williams (1988). *Introduction to the theory of grammar*. Number 12 in Current studies in linguistics. The Massachusetts Institute of Technology.

Wells, R. S. (1947). Immediate Constituents. *Language 23*(2), 81–117.

Zyman, E. (2022). Phase-Constrained Obligatory Late Adjunction. *Syntax 25*(1), 84–121.

Zyman, E. (2024). On the definition of Merge. *Syntax*, 1–37.